LOCATION MODEL FOR CCA-TREATED WOOD WASTE REMEDIATION UNITS

by

Helena Isabel Caseiro Rego Gomes

Dissertation submitted in partial fulfilment of the requirements for the Degree of
Mestre em Ciência e Sistemas de Informação Geográfica
[Master in Geographical Information Systems and Science]

Instituto Superior de Estatística e Gestão de Informação
da
Universidade Nova de Lisboa

LOCATION MODEL FOR CCA-TREATED WOOD WASTE REMEDIATION UNITS

Dissertation supervised by

Professora Doutora Alexandra Branco Ribeiro

Professor Doutor Víctor Lobo

November 2004

LOCATION MODEL FOR CCA-TREATED WOOD WASTE REMEDIATION UNITS

ABSTRACT

There is growing concern about the environmental impacts and increasing difficulty to dispose preservative treated wood products at the end of their service life. In the next decades, in Portugal, a significant increase is expected in the amounts of treated wood that annually needs to be properly disposed. The recycling of these wastes, containing chromium, copper and arsenic (in the case of CCA-treated wood), should only be made after its remediation, so planning and optimisation of the remediation units locations is of major importance.

The objective of this study is the development of a location model to optimise the location of remediation plants for the treatment of CCA-treated wood waste for further recycling, minimizing costs and respecting environmental criteria.

The location model was implemented with geographic information using Geographic Information Systems (ArcGIS 8.2 © ESRI). All the uses of treated wood products were considered, using soil occupation data and the results of a questionnaire sent to wood preservation industries.

Two different clustering methods (Self-Organizing Maps and K-means) were tested in different conditions to solve the multisource Weber problem using SOMToolbox for MATLAB.

The solutions obtained with our data and with both clustering methods make sense and could be used to decide on the location of these plants. SOM has provided more robust and reproducible results than k-means, with the disadvantage of longer computing times. The main advantage of k-means, compared to SOM, is the reduced computing time allied to the fact that it allows us to obtain the best solutions in the majority of the cases, in spite of bigger variances and more geographical dispersion.

MODELO DE LOCALIZAÇÃO DE UNIDADES DE REMEDIAÇÃO DE RESÍDUOS DE MADEIRA PRESERVADA COM CCA

RESUMO

Os resíduos de madeira preservada têm originado uma preocupação ambiental crescente pela dificuldade em encaminhá-los para um destino final ambientalmente adequado. Prevêm-se aumentos significativos neste fluxo de resíduos nas próximas décadas, sendo que dezenas de milhares de m^3 gerados anualmente em Portugal terão de ter um destino adequado. A reciclagem destes resíduos apenas deverá ser realizada após a sua remediação, pelo que o planeamento e optimização da localização de unidades de tratamento assumem uma grande importância.

O principal objectivo deste trabalho consiste em optimizar a localização de unidades de remediação de resíduos de madeira preservada, de forma a assegurar-lhes uma gestão adequada e a minimizar os custos.

O modelo de localização foi implementado com base em informação georreferenciada, utilizando Sistemas de Informação Geográfica (ArcGIS 8.2 © ESRI). Foram tidas em conta as utilizações dos diferentes produtos de madeira preservada, utilizando informação georreferenciada sobre o uso do solo e os resultados do inquérito enviado às indústrias de preservação de madeira.

Foram testados dois métodos diferentes de análise de clusters (*Self-Organizing Maps* e *k-means*), em diferentes condições, para resolver este problema de localização. O software utilizado foi o SOMToolbox para MATLAB.

As soluções obtidas com os dois métodos fazem sentido e podem ser utilizadas para decidir a localização das unidades de remediação. Enquanto que o SOM obteve resultados mais robustos e reprodutíveis, em tempos computacionais mais longos, o *k-means* obteve melhores soluções, apesar da maior variância e dispersão geográfica.

KEYWORDS

CCA-treated Wood Waste

Integrated Waste Management

Location Models

Self-Organizing Maps (SOM)

K-means

Optimisation

PALAVRAS-CHAVE

Resíduos de Madeira Preservada com CCA

Gestão Integrada de Resíduos

Modelos de Localização

Self-Organizing Maps (SOM)

K-means

Optimização

LIST OF ACRONYMS

ACA – Ammoniacal Copper Arsenate

ACC – Acid Copper Chromate

ACQ – Ammoniacal Copper Quaternary

ACZA – Ammoniacal Copper Zinc Arsenate

BMW – Biodegradable Municipal Waste

C&D – Construction and Demolition

CBA – Copper Boron Azole

CC – Copper Citrate

CCA – Chromated Copper Arsenate

CCB – Chromated Copper Borate

CDDC – Copper Dimethyldithiocarbamate

CGA – Constructive Genetic Algorithm

CPLP – Capacitated Plant Location Problem

CPLPSS – Capacitated Plant Location Problem with Single Source Constraints

CSCLP – Capacitated Set Covering Location Problem

CZC – Chromated Zinc Chloride

DDAC – Didecylmethyl Ammonium Chloride

EU – European Union

GIS – Geographic Information Systems

GRASP – Greedy Randomised Adaptive Search Procedure

MCLP – Maximal Covering Location Problem

MFLP – Multiple Facility Location Problem

MSW – Municipal Solid Waste

OFLR – Obnoxious Facility Location and Routing Model

SOM – Self-Organizing Maps

USEPA – United States Environmental Protection Agency

VNS – Variable Neighbourhood Search

VNTS – Variable Neighbourhood Tabu Search

XRF – X-ray Fluorescence

ACKNOWLEDGMENTS

I would like to express my friendship and gratitude to my supervisors for their continuous support. Their encouragement and direction have been greatly appreciated and I could not have finished this work without their precious help.

A special word of thanks goes to Eng. Dario Reimão who was a key element in the wood preservation study in Portugal and provided me important information and encouragement for the development of this dissertation.

Thanks are due to Eng. Rui Sanches (Xiloquímica) and to Eng. Joana Nunes (AIMMP) that helped me with the distribution of the wood preservation industries questionnaire. Thanks to all wood preservation units that answered the questionnaire.

Tanks to Eng. Paula Veríssimo, Eng. Américo Cruz and Eng. Jaime Martins, from PT Comunicações, who provided me essential information, as well as Eng. Tiago Rogado, from IEP.

I would also like to thank Prof. Fernando Bação for the excellent discussions about SOM and other data mining relating matters.

Above all, I am eternally grateful to my parents who have led me to this path and walked with me, and who are teaching me more important things than these. A big hug to my little sister that always was by my side and I know that I can count with.

And, last but not least, I would like to thank to my dear colleagues Madalena, Paula, Elisabete and Roberto for their friendship and support during the last two years. Good luck for your thesis!

Project POCTI/32927/AGR/2000 approved by FCT and POCTI, with FEDER funds, partly supported this thesis.

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 Brief Problem Statement

There is a growing concern about the environmental impacts and an increasing difficulty to dispose treated wood products at the end of their service life. In the next decades a significant increase is expected in the amounts of treated wood removed from service. For example, in the U.S.A. the total volume of treated wood removed from service was estimated to be near 8 million m^3 in 1990, 9 million m^3 in 2000, 15 million m^3 in 2010 and about 18 million m^3 in 2020 (McQueen *et al.*, 1998). In Denmark, it was estimated that, until 2010, about 200 000 m^3 of impregnated wood will end up as waste and must be handled (Ottosen & Christensen, 2004).

In Portugal, tens of thousands of cubic meters of treated waste wood will need to be properly disposed annually. This raises issues such as integrated waste management, environmentally sound solution, responsibility, and finding the appropriate locations to handle these hazardous wastes, minimizing costs.

Location decisions are frequently made at all levels of human organization from individuals to governments. Such decisions are often strategic in nature, involving large sums of capital resources and with long term economic effects. Location models provide a systematic means by which the decision-maker can explore the various alternatives in order to identify an optimal management strategy.

There are several types of location problems, with different objectives and types of constraints. All of them are usually very complex and are sub-optimally solved using heuristics. Besides that, location of waste management facilities represents one of the most complicated location decisions faced by environmental policies (Llurdés *et al.*, 2003).

1.2 Objectives

According to the brief introduction to the studied problem, the main objectives of this dissertation are:

- Estimation of the magnitude of CCA-treated wood waste to be disposed in Portugal;
- Identification of the available management options for the preserved wood at the end of its service life;
- Identification of the key environmental and economical factors to be included in the optimisation model;
- Review of location problems and models, as well as of the different methods and heuristics used to solve these problems;
- Formulation of the location model for CCA-treated wood waste remediation units;
- Definition of the main constraints and identification of the model limitations;
- Simulation of different conditions and scenarios with discussion of the results;
- Testing and comparison of different methods to solve the location problem;
- Production of a final map with the visualization of the “optimal” locations for the remediation infrastructures;
- Identification of further developments and ways to go, as well as new applications for the model implemented.

1.3 Methodology

The methodology used in the development of this dissertation consists, essentially, in the following steps:

1. Extensive bibliographic review of location problems and treated wood waste management;
2. Collection and analysis of data in order to organize a database and identification of the key factors to be considered in the optimisation model;
3. Model implementation, discussion of results and identification of potential directions for further extensions and improvements.

Figure 1.1 presents, in a graphic manner, the methodology used.

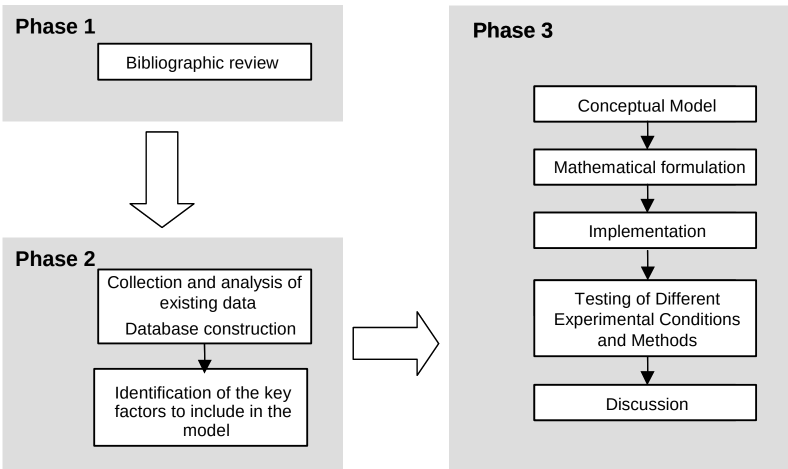

Figure 1.1 – Methodology used for the elaboration of this study.

1.4 Thesis Organization

This thesis is organized as follows: Section 2 reviews the treated wood waste management options, after a brief introduction about the production and uses of preserved wood. The legal framework and the management options of treated wood waste in Portugal are also discussed.

Section 3 presents a review of the location problems and its classification, as well as different search methods that can be used in their resolution. It is also referred the integration / interaction of location models and Geographic Information Systems (GIS).

The inventory of CCA-treated wood waste in Portugal is presented in Section 4. This inventory was based in a questionnaire sent to the preservation industries as well as in the collection and analysis of existing data. The GIS location model of CCA-treated wood is also explained in this section. The formulation of the location model is presented as well as the decision variables, assumptions, constraints and the experimental conditions tested.

Finally, Section 5 presents the results obtained and its discussion, followed by the Conclusions and Further Developments in Section 6.

2. INTEGRATED MANAGEMENT OF TREATED WOOD WASTE

2.1 Introduction

Treated wood removed from service that has no other useful application as a product is considered solid waste and generally did not represent a problem. However, recently in the European Union (EU) this waste was classified as hazardous, so there is a specific need to find an appropriate solution for its management.

Integrated solid waste management can be defined as the selection and application of suitable techniques, technologies, and management programs to achieve specific waste management objectives and goals (Tchobanoglous *et al.*, 1993). A hierarchy in waste management can be used to rank actions to implement specific programs. In the case of the EU, the hierarchy promotes, in first place the prevention or reduction of waste generation and its harmfulness, followed by the recovery of waste through recycling, reuse or reclamation or any other process, aiming to recover secondary raw materials, and finally the use of waste as a source of energy.

Typical waste disposal options for treated wood waste such as incineration and landfill disposal are becoming more costly because of increasingly strict regulatory requirements. Also the development of appropriate alternative management strategies and technologies is constrained by the wide dispersal of treated wood, the poor quality of the material once it is removed from service and the absence of appropriate means to identify treated wood and separate it from untreated wood (Cooper, 2003). So the lack of developed collection, sorting, cleaning and marketing systems continues to be a major barrier to keep this waste stream out of landfills and to increase recycling.

In spite of these barriers, recycling of treated wood wastes in wood composites has been progressively increasing. We also assist to the development of remediation methods to extract the preservatives from the wood.

Next, we discuss the production and use of treated wood, as well as the integrated management of its wastes at the end of service life.

2.2 Treated Wood Production and Use

2.2.1 Production

According to the definition of the European Committee for Standardisation (CEN, 35th Meeting of CEN/TC 38), "*wood preservatives are active ingredient(s) or preparations containing active ingredient(s) which are applied to wood or wood based products themselves, or which are applied to non-wood substrates (e.g. masonry and building foundations) solely for the purpose of protecting adjacent wood or wood-based products from attack by wood-destroying organisms (e.g. dry rot and termites)".*

Three main types of preservatives are used: water based, organic solvent based and tar oils. The preservatives are applied to the surface of wood by pressure impregnation, by dipping or immersion and by thermal processing (immersion in a hot bath of preservative) (WBG, 1998). Application of vacuum helps to improve the effectiveness of the process and to recover some of the chemicals used.

The most common waterborne chemicals are metal oxides, including chromated copper arsenate (CCA), acid copper chromate (ACC), ammoniacal copper arsenate (ACA), chromated zinc chloride (CZC) and ammoniacal copper zinc arsenate (ACZA) (Solo-Gabriele *et al.*, 1998). CCA is actually the most used wood preservative worldwide (Villumsen, 2003).

As its name suggests, CCA consists of three principal compounds – arsenic in the form of arsenic pentoxide ($As_2O_5.2H_2O$), chromium, in the forms of both hexavalent and trivalent chromium ($K_2Cr_2O_7$ or $Na_2Cr_2O_7.2H_2O$), and copper in the form of copper sulphate ($CuSO_4.5H_2O$) or copper oxide (CuO). Sonti Kamesam invented this wood preservative in 1930, in India (Cooper, 1999). The copper (23-25%) and arsenic (30-37%) in CCA act as fungicides and insecticides, while the chromium (38-45%) fixes the chemicals into the wood.

CCA treated timber is green in colour (due to oxidised copper compounds) and becomes grey with age, looking similar to untreated weathered wood. It does not have a distinctive odour, like creosote treated wood.

CCA pressure-treating like any other preservation method can increase the wood's serviceable life substantially, from perhaps 3-7 years for untreated wood, to as long as 60 years (Crawford *et al.*, 2000). Using treated wood reduces consumption of wood resources and reduces costs because of lower maintenance and less frequent replacement. The actual serviceability of pressure-treated wood is dependent on factors including climate, site-specific conditions and type of use.

In Australia, North America (Canada and U.S.A.) and some European Countries (such as Denmark and Portugal), the majority of the wood is treated preventively at industrial scale with vacuum-pressure processes. The picture is different in other European countries, such as Germany, where preventive surface treatments (e.g. immersion, brushing and injection) at the small professional enterprise scale are the predominant methods used (OECD, 2003).

A generic vacuum-pressure process diagram of wood preservation in Portugal is presented in Figure 2.1.

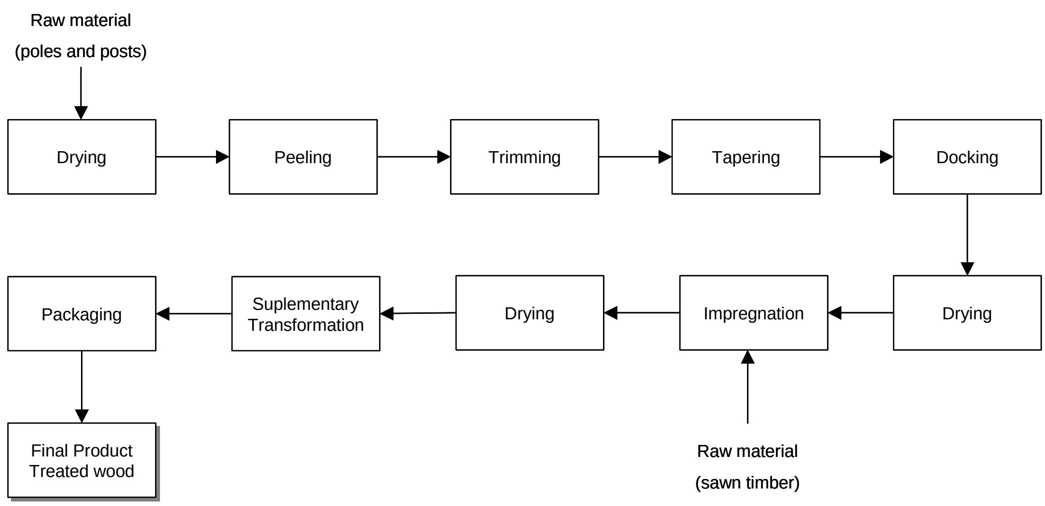

Figure 2.1 – Wood preservation process flow diagram (adapted from INETI, 2001).

Pressure processes are used to apply wood preservative by overcoming the resistance of the wood to deep penetration using pressure. The treatment is carried out in cylindrical airtight steel pressure/vacuum vessels. Two general categories of processes can be distinguished: the pressure-vacuum process and the double vacuum/low pressure process.

Generally, the pressure processes include the following steps (OECD, 2003):

1. Untreated wood is loaded onto small rails or tram cars (bogies) that are pushed into the cylinder using forklifts or other mechanical means;

2. The cylinder door is sealed via a pressure tight door, either manually with bolts or hydraulically, and a vacuum applied to remove most of the air from the cylinder and the wood cells;

3. Treating solutions (either heated or at ambient temperature, depending on the system) are then pumped into the cylinder and the pressure is raised. The total treating times and cycles will vary, depending on the species of wood, the commodity being treated and the desired product retention, but in all instances the treating process remains a closed system.

4. The pressure is released and the excess solution is removed from the cylinder, typically by pumping and goes back to the storage tank for reuse.

5. A final vacuum may be applied to remove the excess preservative that would otherwise drip from the wood. This final vacuum may recover 20-60% of the gross amount of the preservative injected.

6. Final steps in the process are the unloading of the retort and storage of the treated wood.

Wood preservation in Portugal started in the beginning of the 20th century and until the early 50’s was quite incipient (Reimão & Cockcroft, 1985). Since the 50’s, a great interest in the vacuum/pressure type of treatment began and several plants were installed or adapted for this purpose. In 1985, 14 enterprises had plants at 17 places, with a total of 26 cylinders installed (Reimão & Cockcroft, 1985). In 1998 industrial wood preservation by vacuum/pressure included the same 14 enterprises with plants at 22 places and a total of 24 cylinders installed (Ribeiro, 1998). A description of the methods most used in the wood preservation industry in Portugal can be found in Ribeiro (1998).

Some semi-industrial plants are also used for treating wood by immersion or spraying tunnel using organic solvent preservatives. There are also a number of deluging machines and immersion tanks being used by enterprises that produce timber components for building construction (Reimão & Cockcroft, 1985).

The vacuum/pressure impregnation industry in Portugal is mostly located in the western part of the country, north of Lisbon, which is more densely populated and where the demand for timber is larger (see Figure 2.2). It is also located in the area of the maritime pine (*Pinus pinaster*) forests where the raw material is available (Reimão & Cockcroft, 1985).

Figure 2.2 – Location of wood preservation plants (marked with •) in Portugal [data from the National Statistics Institute (Instituto Nacional de Estatística), 2004a].

2.2.2 Use

Examples of treated wood products are telephone and utility poles, railway ties, fence posts and dimension lumbers for outdoor construction. Applications of treated wood outdoors are, for example, house fronts (claddings), roof tiles, window frames, playing tools, garden houses, fences, landings, wharves, bridges, bank revetment, and acoustic barriers, among others. Domestic uses include decks, garden furniture, picnic tables, fences, playground equipment and agricultural or horticultural posts.

In the U.S.A and Canada, homeowners use treated wood extensively and so the distribution of treated wood waste is closely related to the population distribution, i.e., in urban and suburban locations (Cooper, 2003).

The greatest consumer of treated timber in Portugal is the agricultural sector (see Table 2.1), mainly for fence posts, fruit tree supports and vineyard stakes (Reimão & Cockcroft, 1985). Another great consumer is the construction of new highways that are fenced using treated wood posts. In 1984, there were already nearly 200 km of highways fenced using wooden posts (Reimão & Cockcroft, 1985). In 2001, Portugal had build 2601 km of highways (INE, 2002) and the majority were fenced with CCA-treated wood posts. Treated wood is also used on boardwalks in beaches, wetlands, gardens and golf courses.

Table 2.1 – Amounts of preserved wood products made in Portugal in 1984 (Reimão & Cockcroft, 1985).

Products	Amounts produced (m^3)
Railway sleepers	15000
Telephone poles	15000-20000
Agricultural posts	25000-35000
Sawn timber	10000

Besides this data, the only treated wood production data available refers to the period of 1992 to 2002 and was collected in the National Inventory of Industrial Production of the National Statistics Institute (INE). This data is presented latter in Section 4.1.1 (Characterization of Treated Wood Production).

2.2.3 Environmental Problems

There is an increasing public concern about environmental contamination caused by chemicals leaching from treated wood, both in service or disposed in landfills. Special attention has been dedicated to children exposure to CCA in playgrounds due to dislodgeable copper, chromium and arsenic from CCA-treated wood surfaces (EWG & HBN, 2001; Beck, 2002; Read, 2003; Stilwell *et al.*, 2003).

CCA-treated wood leach over time as noted by elevated soil concentrations below CCA-treated decks (FPL, 2000; Townsend *et al.*, 2001). High concentrations of Cu, Cr and As were detected in samples from rainwater, soil and sediments, river and estuary waters adjacent to the CCA-treated wood, contributing to potential direct and indirect human exposure routes (Breslin & Adler-Ivanbrook, 1998; FPL, 2000; Solo-Gabriele *et al.*, 2000; Maas *et al.*, 2002; Weis & Weis, 2002; Chirenjea *et al.*, 2003; Townsend *et al.*, 2003a). A study of soil contamination of wood preservation industrial sites in Portugal can also be found in Ribeiro (1998).

There are significant health risks associated with the exposure to the chemicals used in the wood treatment. Among the chemicals contained in CCA-treated wood, arsenic is characterized by the highest human toxicity (Townsend *et al.*, 2003a). The toxicological effects of the metals present in CCA, the impacts on nontarget organisms and on water quality are reviewed by Cox (1991).

Arsenic and chromium (VI) are both well recognized as strong carcinogens, and As has other detrimental health effects including cardiovascular disease, diabetes, anemia, skin lesions, vascular damage, and reproductive, developmental, immunological, and neurological effects (AWWA, 2004). Chromium (VI) is a known teratogen and can also be a powerful and corrosive irritant to the skin and mucous membranes. Both Cr (VI) and As can be ingested and inhaled, and As can be absorbed through the skin (Maas *et al.*, 2002). Copper is not highly toxic to humans, but it is a metal of concern in aquatic environments since it is highly toxic to aquatic life (MTBI, 2002).

This growing body of scientific evidence that CCA-treated timber poses a danger to both humans and the environment caused authorities around the world to impose tighter restrictions on its manufacture, use and disposal.

In Greece, with the publication of the Decision 1930/18.12.85 of the Ministry of Agriculture, timber impregnated with CCA and CCB (Chromated Copper Borate) can not be used for structures which come in contact with human, food, animal fodder, drinking water and for interior structures (Adamopoulos & Voulgaridis, 1998).

In Sweden, the National Chemicals Inspectorate introduced restrictions on the use of chromium and arsenic containing preservatives in 1994, so that CCA may no longer be used above ground level with a few exceptions (Jermer *et al.*, 2004). In Norway, the wood preservation industry has entered a voluntary agreement with the environmental authorities and the use of arsenic and chromium in wood preservation was phased out by October 2002 (Evans, 2004).

In Japan, since 1997, the acceptable limit of arsenic in wastewater became 0.1 mg L^{-1}. For this reason, Japanese wood preservation industries started to use other wood preservatives like Didecylmethyl Ammonium Chloride (DDAC), Ammoniacal Copper Quaternary (ACQ), Tanalith CuAz, copper-naphthate and zinc-naphthate. In the period of January to June 1997, the share of CCA preservatives was less than 30% in comparison to over 90% in the same period in 1996 (Suzuki, 1998).

In February 2002, the United States Environmental Protection Agency (USEPA) announced that manufacturers had agreed to voluntarily phase out the production of CCA-treated timber for residential uses over the following 2 years (USEPA, 2002) and, in January 2004, the USEPA has officially banned the manufacture of CCA-treated timber for residential use. A similar voluntary phase out for non-residential uses has been instituted in Canada (PMRA, 2002).

In the European Union (EU), the Commission Directive 2003/2/EC, of January 6, stated that arsenic compounds could not be used for the preservation of wood and CCA-treated wood could not be marketed. However, an exception was made for industrial purposes "*provided that the structural integrity of the wood is required for human or livestock safety and skin contact by the general public during the service is unlikely*". Such allowable uses included structural timber in non-residential buildings and industrial premises, bridges and jetties (but not in marine waters), noise barriers, avalanche control, highway fencing, earth retaining structures, power poles and underground railway sleepers. Member states should have adopted and published the Directive by June 2003 and implement it by 30 June 2004 (in Portugal, until now, none of these measures has been implemented). This Directive does not apply to CCA-treated wood already in place.

In spite of this environmental concern, the generalized ban on CCA treated wood will not resolve the issue of the existing treated wood in service that will end in the waste stream over the next years and that needs appropriate management, because a significant fraction of CCA remains within the treated wood product upon disposal. It would thus be imperative to develop disposal management strategies that will recapture discarded CCA-treated wood to properly dispose of the arsenic contained within it (Solo-Gabriele *et al.*, 2003).

2.3 Management Options for Treated Wood Waste

2.3.1 Introduction

In this section, different management options for treated wood waste will be reviewed, considering the hierarchy for waste management, namely minimization, reuse, recycling, incineration and landfill disposal.

Table 2.2 presents the results of the OECD Workshop on Environmental Exposure Assessment to Wood Preservatives (OECD, 2000), where the main routes of disposal of preserved wood were identified in different countries.

2.3.2 Minimization

For treated wood waste, minimization involves the use of alternative chemicals preservatives or/and the use of substitute structural materials. These alternative chemicals are thought to include products based on: copper and boron; chromium trioxide, copper and phosphoric acid; copper, didecylpolyethoxy-ammoniumborate; copper, didecyldimethylammoniumchloride; as well as heat-treated wood (reducing the rotting property of wood by prolonged exposure to about 200°C), according to the European Commission (Gendebien *et al.*, 2002). It should be noted that wood preservation treatment plants could easily switch to the arsenic-free treatment, like ACQ, using their existing equipment (Solo-Gabriele *et al.*, 2000; Gendebien *et al.*, 2002).

Table 2.2 – Main routes of disposal of preserved wood waste (OECD, 2000).

Country	Disposal of preserved wood waste
Australia	- Landfill - Recovery and reuse
Canada	- Landfill - Incineration - Reuse
France	- Incineration with energy recovery - Landfill
Germany	- Regulated reuse of untreated and treated wood - Incineration with energy recovery - Removal of treated wood from waste stream and special incineration - Landfill
Japan	- Regulated incineration
The Netherlands	- Depends on waste stream - Landfill - Incineration - Reuse: energy recover (also export)
Portugal	- Landfill - Incineration - Reuse
Switzerland	- Regulated incineration
Sweden	- Energy recovery - Incineration - Landfill
USA	- Landfill - Hazardous levels incineration

Solo-Gabriele *et al.* (1999) evaluated seven chemical alternatives to CCA. Among these seven, four were identified as the most promising substitutes for existing uses of CCA-treated wood. These four included ACQ, copper boron azole (CBA), copper citrate (CC), and copper dimethyldithiocarbamate (CDDC). The efficacy of the four alternatives was comparable to that of CCA in laboratory and/or field tests. ACQ and CDDC chemical concentrates appear to be more corrosive to treatment plant equipment, in particular to that made of brass and bronze. The cost of ACQ- and CDDC-treated wood was 10 to 30% higher than CCA-treated wood (Solo-Gabriele *et al*., 1999).

Another study developed in Portugal showed that preservation made with the condensed material from the baking steam of expanded cork board production could provide the wood with anti-swelling and anti-absorption characteristics, improve its dimensional stability and increase its resistance to fungi infestation (Gil & Duarte, 1997).

Alternative materials to preserved wood include plastic lumber, a material composed of recycled plastics such as polyethylene, polypropylene, polystyrene, and polyvinyl chloride. Its utilization allows the recycling of plastic and the reduction of the use of CCA-treated wood (Solo-Gabriele *et al.*, 1998; Cooper, 2003). Other alternatives are the use of wood with natural resistance (like western red cedar, mahogany, and teak) and the use of other structural materials like concrete, steel, aluminium, brick, fibreglass, and stone (Solo-Gabriele *et al*., 1998). However more complete and quantitative full life cycle assessments are needed to be sure that the environmental benefits of alternative materials are significant.

It is also possible to reduce the treated wood wastes through design to minimize waste during construction. Development of standard kits for outdoor furniture, modular fence panels and decks would ensure a minimum of re-processing during construction, satisfying simultaneously the consumer requirements (Cooper, 2003).

2.3.3 Reuse

The main difference between reuse and recycling is the processing needed to allow new uses for the wood waste. Reuse requires minimal processing (i.e. the original piece of discarded wood suffers minimal cutting) whereas recycling requires a significant change in the character of the wood waste prior to using it for another purpose (e.g. chipping and biding with adhesives for wood based composite materials) (Solo-Gabriele *et al.*, 1998).

There is a high potential for reuse of commercial products such as utility poles as the treated wood condition is often still very good. Products like treated poles are usually well preserved and in good condition for reuse. Many can be reused for the initial intended purpose or for posts, land pilings and retaining walls. In Canada, 75% of treated poles removed from service are reused (Cooper, 2003). In Greece, the reuse of poles is also high, reaching more than 60% (Adamopoulos & Voulgaridis, 1998). In Finland 50% to 70% of CCA-treated poles are given to landowners at the end of its lifecycle for reuse in other field applications (Hohenthal, 2001). In Ireland replaced poles are sold for agricultural use (McDarby, 2001).

McQueen *et al*. (1998) reported a percentage of 18% of reuse of other treated wood products in plant boxes, mail boxes, flooring and framing for shed, garden stakes, stair steps, fencing around compost bins, dog houses, crab traps and walkways to piers and waterfront and other construction uses. Other possible solutions for reuse CCA treated wood are the framing for compost bins or the reuse as shims and braces in construction (McQueen *et al.*, 1998).

In Portugal, when telephone lines in urban areas are replaced for buried cables, the poles are reused in other lines in rural areas, according to the Portuguese Telecom (PT Comunicações) (Martins, 2004).

Potential problems with these reuse options are related with the generation of contaminated sawdust and consequent human exposure (Solo-Gabriele *et al.*, 1998). Additionally, there are limits to reuse because people frequently get rid of CCA-treated wood since it does not look good anymore. If the wood is not aesthetically pleasing, other people may not want to reuse it.

In Denmark, with very few exceptions (e.g. poles), it is not allowed to reuse CCA treated wood as such and it must be handled as waste when first removed from service at the original place (Ottosen & Christensen, 2004). Also in Norway, since 2002, reuse of CCA-treated wood is not allowed (Evans, 2004).

2.3.4 Recycling

Increased use of waste wood by the composite industry using fiber and small particles appears to be inevitable (McKeever *et al.*, 1995). Approximately 19 x 10^6 m^3 $year^{-1}$ of wood treated with CCA, pentachlorophenol and creosote will be available for recycling by 2020 (Felton & Groot, 1996). In Finland, it is estimated that the amount of treated wood waste for recycling before 2015 will reach 130000 m^3 $year^{-1}$ (Hohenthal, 2001). Incorporation of CCA waste wood in composite materials such as wood cement, structural composites and particleboard has been identified as the recycling option with the highest potential (Solo-Gabriele *et al.*, 1998; Cooper, 2003).

Conventional wood-based composites fall into three main categories based on the physical configuration of the wood: fibreboard, flakeboard and particleboard (Felton & Groot, 1996). Wood-based composites rely on adhesives such as urea-formaldehyde and phenol-formaldehyde for binding purposes. Treated wood can be recycled in composite materials as it is or after removing contaminants through a remediation process (Cooper, 2003). However, it is difficult to bind treated wood due to the brittle nature of CCA-treated wood fibres and the CCA chemical deposits themselves which prevent the adhesive contact with the wood (Solo-Gabriele *et al.*, 1998).

In France, studies have been carried out for the preparation of concrete-wood composites to be used as filling material for road works (Deroubaix, 2004).

CCA-treated wood may also be shredded and used as mulch in right-of-ways, potting soil, and animal bedding. Such practices should be viewed with caution due to the potential for metals leaching. Given that the surface area available for leaching is much greater once the wood is mulched, there is a greater heavy metal leaching (Townsend *et al.*, 2003b). If the production of mulch is considered, technologies for removing CCA from the wood should be implemented first (Solo-Gabriele *et al.*, 1998).

There is also the possibility of using recycled plastic (high-density polyethylene) and CCA-treated wood particles to make composites. Composites containing particles from recycled CCA-treated pine exhibited flexural bending properties higher than those made with either particles from virgin pine or recycled urea formaldehyde bonded particleboard (Cooper, 2003; Kamdem *et al.*, 2004). The biological durability and the photo-protection properties were improved for samples containing recycled CCA-treated wood (Kamdem *et al.*, 2004).

One of the major problems involving recycling is the sorting of treated wood waste. Treated products such as poles are easily recovered but residential CCA-treated wood presents a challenge to collection and transportation because of the increasing quantities and its widespread distribution (Cooper, 2003).

The method for sorting should be effective, economically acceptable and fast enough. Use of electron paramagnetic resonance, colour spot reagents and other colorimetrical methods seems to be suitable (Humar & Pohleven, 2001). Laser-induced atomic emission spectroscopy and X-ray fluorescent analysis also have been used with good results (Peek, 1998). The sorting technologies, chemical stains and X-ray fluorescence (XRF), evaluated by Solo-Gabriele *et al*. (1999), showed considerable promise for sorting CCA-treated wood from other wood types. A pilot study using a XRF detector system reached 95% efficiency if recycled wood is to be used for fuel and at least 99.8% efficiency if the wood is to be recycled as mulch (Solo-Gabriele *et al.*, 2001).

After sorting, the next step is chipping of treated wood (see Figure 2.3). This process enables the easier handling of the wood in comparison with its original size and increases the surface area available for remediation. It may also be a preliminary step of the recycling process if the treated wood would be used as it is, without previous decontamination. Chipping of treated wood could be achieved either mechanically, either by steam explosion (Shiau *et al.*, 2000; Humar & Pohleven, 2001). Two recycling plants in Germany, with an annual capacity of around 50000 Mg[1], apply a disintegration method developed in the Fraunhofer Institute for Wood Research (WKI) (Speckels & Springer, 2001).

[1] We use the International System (SI) of Units and Mg corresponds to 10^6 g.

Different methods for wood preservative removal have been tested. The electrodialytic process[1] at laboratorial scale obtained removal efficiencies of 93% of Cu; 95% of Cr and 99% of As from CCA treated sawdust (Ribeiro *et al.*, 2000). At pilot scale, the best remediation efficiency was achieved in an experiment with an electrode distance of 60 cm (100 kg wood chips) and duration of 21 days. In this experiment 88% of Cu, 82% of Cr and at least 96% of As were removed from the wood (Villumsen, 2003).

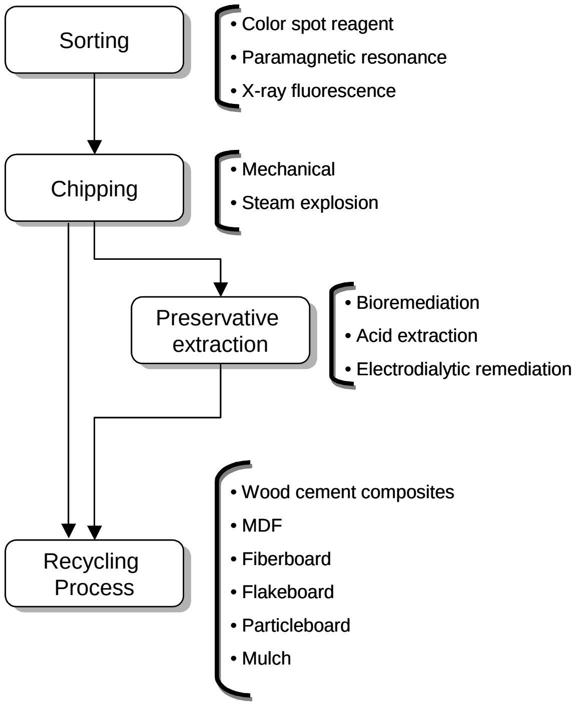

Figure 2.3 – Processes in the recycling of treated wood waste.

Other decontamination processes include bioremediation. Results suggest that a dual remediation process involving acid extraction (0.8% oxalic acid extraction for 18 h) and exposure to metal-tolerant bacteria (7 days bacterial culture with *Bacillus licheniformis* CC01) removes significant quantities of metals from CCA-treated wood (78% of Cu, 97% of Cr and 93% of As) (Clausen & Smith, 1998; Clausen, 2000). However the evaluation of the properties of particleboard

[1] The electrodialytic process uses a low-level direct current as the "cleaning agent", combining the electrokinetic movement of ions in the matrix with the principle of electrodialysis (Ribeiro *et al.*, 2000).

made from recycled CCA-treated wood showed that the oxalic acid treatment may have caused a slight embrittling of the fibre before pressing or interfered with urea-formaldehyde resin adhesion, causing an increase in the modulus of elasticity and decreases in the modulus of rupture and in the internal bound (Clausen & Muehl, 2000; Clausen *et al.*, 2001). The particleboard containing oxalic acid extracted and bioremediated particles showed generally high leaching losses of remaining elements (Kartal & Clausen, 2001).

Additionally, successful remediation of the chips was also accomplished through inoculation with fungi, achieving 98% removal of chromium. Among the wood destroying basidiomycetes that are tolerant to Cu, Cr, or As based preservatives, there are some species of the genus *Antrodia* (*Poria*) (Leithoff *et al.*, 1995; Stephan *et al.*, 1996; Humar & Pohleven, 2001). Good results have also been obtained with *Aspergillus niger* fermentation, with 97% removal of arsenic (Kartal *et al.*, 2004). In another study, the isolates of *Meruliporia incrassata* and *Antrodia radiculosa* gave the highest percent degradation of CCA-treated wood (Illman *et al.*, 2000).

The chemical extraction with hot sulphuric or nitric acids, aqueous ammonia solutions, acetic and formic acids, chelating organic acids such as citric and tartaric acids or other ones such as EDTA has also shown good results (Shiau *et al.*, 2000; Cooper, 2003; Kartal, 2003; Kartal & Kose, 2003). Likewise other experiments with hydrogen peroxide showed good results in extracting CCA (Kazi & Cooper, 1999).

However, a significant question remains: what concentration of residual preservative in remediated wood would dictate its classification as treated material? The new German Ordinance in Waste Wood[1] (*Altholzverordnung*), put in force in March 2003, specifies the limit values for wood chips used in the manufacture of derived timber products. There are also the European Panel Federation (EPF) criteria, but the thresholds defined are considerably higher than the German limits (see Table 2.3).

[1] The German Ordinance on the Management of Waste Wood is available in the URL http://www.bmu.de/files/wastewood_ordinance.pdf.

Table 2.3 – Limit values for wood chips used in the manufacture of derived timber products.

Parameter	Concentration* (mg kg^{-1} dry mass)	Threshold** (mg kg^{-1})
Arsenic	**2**	**25**
Lead	30	90
Cadmium	2	50
Chromium	**30**	**25**
Copper	**20**	**40**
Mercury	0,4	25
Chlorine	600	1000
Fluorine	100	100
PCP	3	5
PCB	5	-
Benzo(a)pyrene	-	0.5

* German Ordinance on Waste Wood

** European Panel Federation (Deroubaix & Cornillier, 2004)

Recycling into composite products is technically feasible and shows considerable promise, but is constrained by the belief that composites do not need preservative protection, concerns that landfill disposal is only deferred and that the identity of the treatment is lost, as well as a general reluctance of composite manufacturers to include treated wood (Cooper, 2003). Additionally, the segregation of wood waste is a complex and highly expensive option, that relies on understanding the flows of wood waste and the provision of consistent feedstocks to manufacturers of recycled products (Suttie, 2004).

There are also health concerns with processing treated wood, namely worker exposure to air emissions associated with recycling processes such as comminution sawing (airborne particles and fumes) and composite manufacturing (Felton & Groot, 1996).

According to Cooper (2003), less effort has been devoted to find ways to reuse the extracted CCA components. Recently, the possibility of using the process liquids from electrodialytic remediation pilot experiment as make-up water instead of tap water has been investigated and the results indicate that, due to problems with precipitation and/or incomplete fixation, the liquids cannot be used directly, but they are most probable usable in a diluted form (Villumsen, 2003).

2.3.5 Incineration with Energy Recovery

Treated wood can be used for energy recovery purposes [in Waste-to-Energy (WTE) units] or as a fuel in certain industrial operations such as cement kilns. Treated wood waste is also used in co-generation plants (Cooper, 1999; Cooper, 2003). In Greece, co-combustion of treated wood with coal has been tested in two power plants (Spanos *et al.*, 2001).

In Sweden, the main option for waste wood management, both domestic and foreign (e.g. Germany, The Netherlands, Norway), is incineration with energy recovery (Jermer *et al.*, 2004).

Wood utilized for energy recovery purposes has the advantage that it lessens the dependence on fossil fuels and eases the landfill burden through volume reduction (Solo-Gabriele *et al.*, 1998).

In the European Union, the incineration of treated wood waste, namely wood wastes containing organochlorinated compounds or heavy metals, is covered by the Directive 2000/76/EC relative to waste incineration. In Denmark, one of the few countries with legislation on treated wood waste, the incineration of these wastes is banned (Ottosen & Christensen, 2004).

The incineration of CCA treated wood in conventional wood combustion equipment could lead to arsenic emissions and produce high concentration Cu, Cr and As containing solid residues (Deroubaix, 1998; Cooper, 2003; Ottosen & Christensen, 2004). High concentrations of Cu, Cr and As were detected in Florida, in the ashes of an industrial furnace that burned wood waste (Solo-Gabriele *et al*., 2002). This makes impractical the application of the ashes to agricultural uses. Preferably, CCA treated wood should be incinerated in hazardous waste incineration plants and the bottom and flying ashes should also be disposed as hazardous waste, due to metal contents or these ashes should be submitted to a remediation process before being disposed (Ottosen *et al.*, 2003).

In fact, the authors of the pilot study of electrodialytic remediation concluded that after remediation of CCA treated wood waste it is possible to incinerate the wood in a municipal solid waste incinerator with energy recovery. There will be no risk of arsenic release and the copper and chromium rich ashes may be reused as a secondary raw material in the metal refining industry (Villumsen, 2003).

In Norway, during 2003, promising incineration tests were made, using a 10% mixture of CCA-treated wood in a common waste combustion site. Burning the treated wood with domestic waste, instead of sort it out, would enable a significant cost reduction for the treatment of these wastes (Evans, 2004).

Another important problem associated with the combustion of CCA-treated wood is the increase of several orders of magnitude in the levels of polychlorinated dibenzo-pdioxins and dibenzofurans (PCDD/Fs), compared to that of untreated timber (Tame *et al.*, 2003). CCA-treated wood contains copper (II), which is known to catalyse the so-called *de novo* formation of PCDD/Fs. Comparable levels of PCDD/Fs would be expected in residual ash from burning CCA-treated wood in backyard fires, stoves and wood heaters, as a consequence of similar combustion conditions (Tame *et al*., 2003).

Pyrolysing the CCA treated wood waste may be a good solution since low temperatures and no oxidising agents are used, which result in lower loss of metals compared to combustion (Deroubaix, 2004). Pyrolysis results in volatilisation of the volatile compounds whereas the mineral compounds (containing the metals) remain entrapped in a coal-type residue that is very rich in carbon. The condensable compounds in the pyrolysis gas condense while leaving the reaction zone due to the inverse temperature gradient. The pyrolysis gas leaving the reactor is used as fuel for the hot gas generator. The charcoal which is extracted at the bottom of the reactor, is cooled, compressed, removed and stored, ready to feed the subsequent stage where the metal particles are removed (Helsen *et al.*, 1998; Deroubaix, 2004).

2.3.6 Disposal

Disposal is the last option in the waste management hierarchy. Landfill disposal does not recover any value from the used product and potentially presents some problems of soil and groundwater contamination if the landfill is not properly conceived or explored. However it is one of the waste management options that absorb a great portion of the treated wood waste. It is estimated that 80000 Mg of preservative treated wood waste (of which as much as 50% may be CCA treated) are committed to landfill every year in the United Kingdom (Suttie, 2004).

The European Union (EU) has taken an aggressive stance against landfill disposal of all organic material, including wood, with the Council Directive 1999/31/EC of 26 April. By 2005, biodegradable municipal waste going to landfills must be reduced to 75% of the total amount (by weight) of biodegradable municipal waste produced in 1995. So, in the European Union countries, the amount of wood disposed in landfills must be substantially reduced or even totally avoided.

CCA treated wood is the most relevant source for the arsenic in municipal solid waste (MSW) that is subject to easy leaching in landfills (Gendebien *et al.*, 2002). The report of the European Commission about Hazardous Household Waste (HHW) refers a detailed study for the German Environmental Agency that further concluded that arsenic is the most important contributor to carcinogenic properties of landfill leachate, given the relatively high concentration of 1.6 mg L^{-1} in MSW landfills (Gendebien *et al.*, 2002). The German study estimates that over a period of an hundred years, a total of 1.2 g arsenic will be emitted per tonne of waste, which is equivalent to 24% of the arsenic content of MSW.

Another environmental problem associated with treated wood waste disposal is the deposition of these wastes in Construction and Demolition (C&D) landfills, which are generally unlined (Solo-Gabriele *et al.*, 2000; Solo-Gabriele *et al.*, 2001), enabling the leaching of Cr, Cu and As to the subjacent soils and groundwater.

In Australia, a review of the treated wood waste landfill disposal risks recommended improvements in existing landfills and the development of new lined landfills that could accept large quantities of treated timber (Pruszinski, 1999). This study also focuses on the monitoring of groundwater, assessing copper, chromium (III and VI) and arsenic.

2.4 Extended Producer Responsibility

There is a growing demand by regulatory agents and consumers for Extended Producer Responsibility of all manufactured goods, which is basically the application of the Polluter Pays Principle. The manufacturer must take responsibility for wastes generated by his product and for its disposal at the end of its life cycle.

This policy instrument is going to play an increasing role in achieving waste reduction, requiring producers to take responsibility for their products at the end of their lives, and to recover and recycle an increasing proportion of the materials embodied in them. The added cost to the manufacturer will eventually be passed on to the consumer, and the most successful manufacturer will be the one that minimizes the cost.

Application of this principle to treated wood would make the wood preservation industry responsible for taking back and dealing with spent consumer wood. What would this mean to the preservation industry? The option would be to treat with a system that offers no particular difficulties in disposal or to develop a viable recognition and sorting system and recycling technologies for the treated wood (Cooper, 2003).

In 1998, the Finnish Environment Institute and Finnish Wood Preserving Industry signed a branch agreement where industry was obligated to take care of treated wood at the end of service life (Tãhkãlã, 2002). In 2000, the Finnish Wood Preserving Industry established a company – DemoliteOy – to manage treated wood waste. The main shareholder is FWPA (Finnish Wood Preserving Association). This organization collects separately treated wood that is crushed and burned, recovering energy and using the ashes in metallurgical processes (Tãhkãlã, 2002).

2.5 Characterization of the Portuguese Situation

2.5.1 Legal Background

The Portuguese legal framework for waste management is defined in the executive order law 239/97 (“Decreto-Lei nº 239/97”), dated from September 9. This law establishes the general principles and applicable rules relating to waste management and final destination responsibilities. Waste management comprises the collection, transportation, storage, treatment, valorisation and elimination of waste, including monitoring of waste final disposal installations after its closure, as well as the planning of such operations.

The Portuguese law establishes the general principle that the producer is responsible for its produced waste. Also, in the case the producer cannot be known or determined, the waste holder is responsible for waste disposal and for its management costs. Responsibility for waste collection and treatment lies traditionally with the municipalities, currently associated in two different types of municipal waste management systems.

The Environmental Ministry (“*Ministério do Ordenamento do Território e do Ambiente*”) is the government body responsible for the Portuguese policy on waste. It acts at a central level, through the Waste Institute (“*Instituto dos Resíduos*”) and regionally, via the five Coordinating Commissions for Regional Development (“*Comissões de Coordenação do Desenvolvimento Regional*”). Other agencies with responsibilities in this field are the General Inspectorate for the Environment (“*Inspecção Geral do Ambiente*”) and the Water and Waste Regulatory Institute (“*Instituto Regulador das Águas e Resíduos*”).

In the above-mentioned legal document (“*Decreto-Lei nº 239/97*”), waste is divided into five legally defined waste types, according to their origin, nature and composition: hazardous waste, industrial waste, municipal waste, medical waste and other wastes. Different management mechanisms and responsibilities for are foreseen according to each of waste types.

The term ‘hazardous waste’ is defined in Council Directive on Hazardous Waste 91/689/EEC. This Directive was followed by a Council Decision 1994/904/EC of 22 December 1994, establishing the list of Hazardous Wastes. This Decision was replaced by Commission Decision 2000/532/EC, which established a new list of hazardous and non-hazardous wastes. This Decision has been amended by the Commissions Decisions 2001/118/EC, 2001/119/EC and 2001/573/EC.

The ordinance 209/2004 (“*Portaria Nº 209/2004*”), dated from 3rd March, transposed this European legal framework to the Portuguese Law. Treated wood waste can be classified with the codes showed in Table 2.4.

Table 2.4 – Classification of treated wood waste according to the European and Portuguese regulations.

Code	Description
03 01 04*	Sawdust, shavings, cuttings, wood, particle board and veneer containing dangerous substances
15 01 10*	Packaging containing residues of or contaminated by dangerous substances
17 02 04*	Wood containing or contaminated with dangerous substances in construction and demolition wastes
20 01 37*	Wood containing or contaminated with dangerous substances in municipal waste including separately collected fractions

* stands for hazardous waste

The management of hazardous wastes should be carried out with special care and separately from the other waste types in compliance with all the national and European legal requirements.

At the beginning of 2004, the executive order law 3/2004 (“*Decreto-Lei Nº 3/2004*”), dated from January 3, established the legal framework for obtaining the permits for the installation and operation of the Integrated Centres of Recovering, Valorisation and Elimination of Hazardous Wastes (“CIRVER”). In Portugal, maximum of two “CIRVER” will be authorized for the management of hazardous waste produced at a national level.

The Council Directive 1999/31/EC, of 26 April, on the landfill of waste, including the targets for reducing the biodegradable municipal waste (BMW), was transposed to the Portuguese law by the executive order law 152/2002 ("Decreto-Lei Nº 152/2002"), from 23rd May. The country has now to implement the necessary measures to avoid the biodegradable waste deposition, in order to accomplish to the European targets.

2.5.2 Treated Wood Waste Management

Portugal has lacked proper urban and industrial waste management for many years. The collection and treatment of solid waste was recently initiated and only in 2002 was announced that the more that 300 municipal uncontrolled dumps that existed in Portugal were sealed and recovered.

Regardless the efforts made in municipal solid waste management, Portugal has been delaying the implementation of hazardous waste management systems, mostly for political reasons. This results in illegal dumping of hazardous wastes and, consequently, in severe contamination problems that at this time cannot be correctly evaluated.

The latest development in this area was the inventory of the industrial waste (referred to the year 2001) developed by six Portuguese Universities in collaboration with the National Statistics Institute (INE), on behalf of a protocol established with the Environmental Ministry (the former "*Ministério das Cidades, Ordenamento do Território e Ambiente*"). This study obtained a total annual production of approximately 254000 Mg of industrial hazardous waste that result directly from the national industrial activity (Santana *et al.*, 2003).

In the previous study, the inventory dedicated to the wood preservation industry only quantifies the wastes of the industrial process and, therefore, the treated wood that reaches the end of its service life is not determined (Borrego *et al.*, 2003). From the wastes referred in Table 2.4 only sawdust, shavings, cuttings, wood, particleboard and veneer containing dangerous substances (code 03 02 04*) were quantified (a production of about $1{,}14 \times 10^{-1}$ kg m^{-3} of treated wood was determined) (Borrego *et al.*, 2003). Considering the 2002 production of treated wood products (117904 m^3), approximately 13,4 Mg of treated wood waste are produced annually in Portugal, in the form of sawdust, shavings, cuttings, wood, particleboard and veneer.

Another important document is the National Plan for the Prevention of Industrial Wastes (PNAPRI), where the wood preservation section is also considered (referred to the year 1998). According to this plan, the annual amount of wood waste produced in the preservation industries in Portugal is 1754 m^3 (INETI, 2001), what corresponds to approximately 1123 Mg of wood waste. Nevertheless this study does not discriminate between treated and non-treated wood.

The only information available about management of treated wood waste in Portugal is the one presented in Table 2.2, from the OECD Workshop on Environmental Exposure Assessment to Wood Preservatives (OECD, 2000). According to OECD, in Portugal the treated wood waste is disposed in landfills, incinerated or reused, but there is no information about the importance of any of these management options. Anyway, Portugal lacks an installation with appropriate conditions to incinerate hazardous wastes (there are only three municipal waste incinerators) and the Portuguese Government has also rejected the hazardous waste incineration in cement kilns.

According to the Portuguese Telecom (“PT Comunicações”), all telephone poles that reach the end of their service life are removed and sold to construction builders for reuse.

Portugal has been implementing extended producer responsibility systems for some specific waste streams like packaging, batteries, tyres, medicines, vehicles at end of life and, in a near future, waste oils and discarded electrical and electronic equipment. All these systems have a management society that represents the producers and is responsible for the proper final destination of the wastes. The producers have to pay a tax to assure the collection, sorting and, preferably, valorisation of the wastes generated by the products they place in the market. A similar system can be implemented for treated wood waste (see section 2.4).

Another gap in the Portuguese law is the management of Construction and Demolition (C&D) waste stream that can be a significant source of treated wood waste. It is important to sort the treated wood from these activities and to assure an appropriate final destination. The project of law currently in discussion establishes the obligation of a hazardous waste inventory previous to the beginning of the works and a declaration to the competent authorities of the quantities and management operations / operators of all C&D waste.

3. LOCATION PROBLEMS

3.1 Introduction

The purpose of location problems is to determine where to place a set of facilities (resources) in order to optimise certain objectives, such as minimizing transportation costs, providing equitable service to customers, maximizing market share captured and/or minimizing the time taken to deliver services, subject to specific constraints. Facility location models can differ in their objective function, the metric distance applied, the number and size of the facilities to locate and several other decision constraints (Hale & Moberg, 2003).

Facility location models are used in a wide variety of applications, in the public and private sectors: airports, bus stops, restaurants, fire stations, hospitals, waste treatment plants, warehouses are only some examples [see Current *et al.* (2002) for an extensive review of location problems applications].

Location models are also application specific, that is, their structural form (objectives, constraints and variables) is determined by the particular location problem under analysis. Consequently, a general location model that is appropriate for all potential or existing applications does not exist.

The universality of location decision-making has led to a strong interest in these problems and models within the scientific community, namely in the operations research and the management areas (Current *et al.*, 2002). The location problems research community also includes mathematicians, computer scientists, industrial engineers, regional planners, marketing researchers, among others, each one emphasizing a particular aspect of the problem (Bender *et al.*, 2002). However, due to the cross-disciplinary nature of location science many of the references are disjoint and some overlap (Hale & Moberg, 2003). Accordingly, location problems are solved with methods from different disciplines, not using a common language to describe the problems, and frequently researchers refer to the same location problem using different names.

Another problem is the difficulty to establish a continuous knowledge transfer between the researchers since, for example, operational researchers have a completely different way of describing their solution approaches than regional planners. As a consequence, improved and new solutions are not used because those that need the practical solution to a location problem do not understand the language in which the solution has been stated (Bender *et al.*, 2002).

To deal with these problems, some classification schemes have been proposed, in order to enrich scientific discussions and to eliminate misunderstandings in verbal model descriptions (Brandeau & Chiu, 1989; Eiselt *et al.*, 1993; Hamacher & Nickel, 1998).

In this section the classification and definition of the most important location problems are presented, as well as the main search methods cited in literature. The integration and interaction of location models and GIS is also discussed.

3.2 Classification of Location Problems

Different criteria are used, together or separately, to identify and classify location problems. The common elements in these problems are the demand points, located in space and with a demand specified and the facilities to be located, in order to optimise the objectives considered and respecting the specified constraints.

3.2.1 Decision Space

Location problems are generally solved on one of the three basic spaces: continuous, discrete and network spaces. The first deals with location problems on a continuous space (in one, two, or three dimensions) where the facilities can be located anywhere in the feasible region. The second looks at problems where locations must be chosen from a pre-defined set, while the third looks at location problems where the set of potential facility sites is located along some underlying network. Furthermore, the space available to facilities may be restricted by the introduction of forbidden zones and/or barriers, that allow modelling of important constraints. There are also multi-level network optimisation problems, that arise in many contexts such as telecommunication, transportation and electric power systems (Cruz *et al.*, 2003).

A further distinction in network problems is the location of the demand points and the location of the facilities. Demand points may be located at the nodes and also anywhere along the links between the nodes. In the latter case the demand points can also be treated as nodes, which are sometimes called pseudo-nodes because they only connect two links. Usually the demand points are assumed to be at the nodes and the facilities may either be located at the nodes and along the links or only at the nodes. In the former case it is called absolute location and, in the latter case, it is called vertex location problem (Rooij, 2000).

Continuous location models are often used as first approximations to real world problems, as a method to find out which regions are the most promising ones to be studied with more precision, since discrete or network models are more difficult to implement due to the necessity of huge amounts of data and to the complexity of the problems (Plastia, 2002).

There are also location problems in which facilities are extensive, i.e. those that can not be represented by isolated points, but as some dimensional structures, such as straight lines, line segments, polygonal curves or circles. A review on continuous location problems of dimensional structures can be found in Díaz-Báñez *et al.* (2004).

3.2.2 Number of Facilities (New and Existing Facilities)

The number of facilities to be located is also considered in the classification of location problems. In the simplest case, only one facility is to be located, but more often there are $p \in$ N facilities to be located (multiple facilities). Some authors also distinguish if the number of facilities is fixed or variable and the number of existing facilities (Eiselt & Laport, 1995). The last is used specifically when the decision maker wants to locate the facilities in an empty space without any similar or competing facilities (competitive location models) (Drezner, 1995b; Drezner & Eiselt, 2002; Plastria & Carrizosa, 2003).

Competition models or competitive location models involve the case where a competitor has the capability to readjust to any location decisions other competitors make over predefined time frames. If you locate a new branch that exploits a poorly served area of your competitor, he may, over a period of time, relocate a branch or locate a new one to recover some of that lost market. Thus, in making a location decision, it becomes necessary to analyse what response your competitor may have to keep from a lost of business. Competition models are often addressed by game theory and simulation (Eiselt & Laport, 1995; Ahn *et al.*, 2004).

In practice, many location problems occur in three different settings (Rooij, 2000): i) there are no restrictions on the location of facilities and there are no existing facilities (general problem); ii) locate new facilities in addition to existing facilities (additional or incremental facility location problem); iii) locate additional facilities and possibly close down some existing facilities (reorganization problem).

3.2.3 Objective Function

The dichotomy between public versus private sector problems is a common way of classification. Profit maximization and capture of larger market shares are the main criteria in private applications, while social cost minimization, universality of service, efficiency and equity are the objectives in the public sector (Marianov & Serra, 2002).

The following objective functions are the most used: minsum or minisum (minimizing the sum of weighted distances) and minmax or minimax (minimizing the maximal distance to the facility). We should also consider maxisum (maximizing the sum of weighted distances) and maximin (maximizing the minimal distance to the facility). In the existing literature, mainly three types of objective functions are used (Nickel & Puerto, 1999): the median objective (minsum), the center objective (minmax) and the cent-dian objective (a convex combination of minsum and minmax, also called k-centrum location problem). A review of objectives in location problems can be found in Eiselt & Laport (1995).

There are also multi-objective models that consider more than one objective. The simultaneous consideration of more than one objective improves the realism of the mathematical model and provides the decision maker with an efficient set of solutions (Current *et al.*, 1990; Giannikos, 1998; Lahdelma *et al.*, 2002; Melachrinoudis & Xanthopulos, 2003; Rakas *et al.*, 2004). A review of multicriteria planar location problems is given by Hamacher & Nickel (1996). Other multi-objective models are multicriteria semi-obnoxious network location problems where some of the sum-objectives are maximized (push effect) and some of the sum-objectives are minimized (pull effect) (Hamacher *et al.*, 2002; Krarup *et al.*, 2002).

3.2.4 Static and Dynamics Models

We can also distinguish between static and dynamic models. Static models try to optimise system performance for one representative period (all of the facilities are to be opened at one time and remain open over the planning horizon). Dynamic models recognize that problem parameters (e. g. demand) may vary over time and attempt to account for these changes over an extended time period. Explicitly dynamic models are those designated for problems where the facilities will be opened (and possibly closed) over time according to changes in the problem parameters (Drezner, 1995a; Owen & Daskin, 1998; Burkard & Dollani, 2001; Current *et al.*, 2002). A good review of dynamic facility location models is given by Owen & Daskin (1998). There is also the possibility of considering time-varying networks (Serra & Marianov, 1998; Hakimi *et al.*, 1999).

3.2.5 Deterministic and Stochastic Models

In practice, there is considerable uncertainty in most facility location problems: demand, travel time, facility costs, and even distance may change and these changes are often random. As a consequence, we have either deterministic models if input is (assumed to be) known with certainty or probabilistic models if input is subject to uncertainty. Stochastic formulations attempt to capture the uncertainty in problem input parameters such as forecast demand or distance values. The stochastic literature is divided into two classes: one that explicitly considers the probability distribution of uncertain parameters, and another that captures uncertainty through scenario planning (Owen & Daskin, 1998; Piersma, 1999). The consideration of time and uncertainty in location problems has helped to move us towards solving more realistic problem instances.

3.2.6 Allocation and Routing Models

There are also location-allocation models, where the demand points have fixed total demands, which are to be optimally allocated to the facilities (Cooper, 1963). These problems may be generically stated as: *"Given the location of a set of customers or demand centres and their associated demands, find the number and location of supply centres and the corresponding allocation of the demand to them so as to satisfy a certain optimisation criteria"* (Lozano *et al.*, 1998).

Other type of location problems are location-routing problems, where the overall effectiveness of the facility location depends, not only upon the distances from the individual demands, but also of the efficiency of the vehicle routes needed to serve multiple demands. In these problems, the objectives are, simultaneously, minimizing the weighted sum of distances and minimizing the maximum distance (Berman *et al.*, 2002). So, location-routing problems involve three inter-related, fundamental decisions: where to locate the facilities, how to allocate customers to facilities and how to route the vehicles to serve customers (Current *et al.*, 2002).

3.2.7 Classification Schemes

Proposals of classification schemes for location models have existed since 1979, when Handler and Mirchandani suggested a 4-position scheme which is applicable to network location models with center type objective functions (Hamacher & Nickel, 1998). In Position 1, information is given about the new facilities and position 2 contains information about existing facilities. In Position 3 the number of new facilities is given and in Position 4 the network type is described. This scheme is only usable for network models with a center-type objective and there is no option for describing special assumptions on the demand (Hamacher & Nickel, 1998).

Brandeau & Chiu (1989) present a taxonomy to distinguish location problems with respect to three criteria (objective, decision variables, system parameters) in table format, but did not provide a formal classification scheme. They survey more than 50 representative problems in location research based on the taxonomy presented in Table 3.1.

The review on multi-objective location problems made by Current *et al.* (1990) presents an analysis based in objectives (cost minimization, demand oriented, profit maximization and environmental concern) and considers five structural characteristics of these problems too. First, the number of facilities being sited is considered. Second, it indicates if the facilities are capacitated. Third, whether the decision space is continuous or discrete. Fourth, the parameters of the problem (e.g. demand, profit) are specified as deterministic or stochastic. Finally, it is indicated if the model is static or dynamic (Current *et al.*, 1990). However, these authors have not proposed a classification scheme, the criteria have only been used to analyse and evaluate the published papers exclusively on multi-objective location problems.

Eiselt *et al*. (1993) used a 5-position scheme which classifies competitive location models (those based on a noncooperative game-theory approach). In Position 1 information about the decision space is given (linear segment, line, circle, bounded subset of m-dimensional real space). Position 2 indicates the number of players (specified number, any arbitrary fixed number, markets with free entrance). In Position 3 a description of the pricing policy (mill pricing, uniform delivered pricing, perfect spatial discriminatory pricing) is given and in Position 4 the rules of the game under consideration are defined (Cournot-Nash equilibrium, subgame perfect Nash equilibrium). Position 5 describes the behaviour of the customers (minimization of distance, maximization of a deterministic utility, maximization of random utility). This classification scheme is too specialized to be a prototype for a general classification scheme for location models (Hamacher & Nickel, 1998).

Table 3.1 – Taxonomy developed by Brandeau & Chiu (1989).

I – Objective	Optimizing: Minimize Average Travel Time/Average Cost Maximize Net Income Minimize Average Response Time Minimize Maximum Travel Time/Cost Maximize Average Travel Time/Cost Minimize Server Cost Subject to a Minimum Service Constraint Optimize a Distance-Dependent Utility Function Other Non-Optimizing Type of Location Dependence of Objective Function: Server-Demand Point Distances Weighted vs. Unweighted Some vs. All Demand Points Routed vs. Closest Inter-Server Distances Absolute Server Location Server-Distribution Facility Distances Distribution Facility-Demand Point Distances Other
II – Decision Variables	Server/Facility Location Service Area/Dispatch Priorities Number of Servers and/or Service Facilities Server Volume/Capacity Type of Goods Produced by Each Server (in a Multi-Commodity Situation) Routing/Flows of Server or Goods to Demand Points Queue Capacity Other
III – System Parameter	Topological Structure Link vs. Tree vs. Network vs. Plane vs. *n*-Dimensional Space Directed vs. Undirected Travel Metric Network Constrained vs. Rectilinear vs. Euclidean vs. Block Norm vs. Round Norm vs L_p vs. Other Travel Time/Cost Deterministic vs. Probabilistic Constrained vs. Unconstrained Volume-Dependent vs. Nonvolume-Dependent Demand Continuous vs. Discrete Deterministic vs. Probabilistic Cost-Independent vs. Cost-Dependent Time-Invariant vs. Time-Varying Number of Servers Number of Service Facilities Number of Commodities Server Location Constrained vs. Unconstrained Finite vs. Infinite Number of Potential Locations Fixed vs. Dependent on System Status Zero vs. Nonzero Relocation Cost Deterministic vs. Probabilistic Location Zero vs Nonzero Fixed Cost Server Capacity Capacitated vs. Uncapacitated Reliable vs. Unreliable Service Area and Dispatch Priorities Cooperating vs. Noncooperating Servers Closest Distance vs Nonclosest-Distance Service Area Service Discipline FCFS vs. Priority Classes vs. Nonclosest-Distance Service Area Queue Capacity

Carrizosa *et al*. (1995) present a 6-position scheme for classifying planar models where both demand rates and service times are given by a probability distribution. Position 1 gives information about the distribution of existing facilities while Position 2 contains information about the distribution of new facilities. In Position 3 the number of new locations is given and in Position 4 the shape of the new location is described. Position 5 describes the shape of the existing facilities and, finally, Position 6 defines the metric used in the model to measure distances. This scheme is again very specialized and the six positions make it quite unmanageable (Hamacher & Nickel, 1998).

More recently, Hamacker & Nickel (1998) published a 5-position classification for location problems that has been used since 1992, when it was first implemented in a course on planar location theory. This classification is the basis of the software library LoLA (Library of Location Algorithms) (Bender *et al.*, 2002). This classification scheme has five positions written as: Pos1/Pos2/Pos3/Pos4/Pos5, where Pos1 provides information about the number and type of new facilities. Pos2 identifies the type of location model with respect to the decision space, distinguishing between continuous, network and discrete models. Pos3 is a description of particulars of the specific location model, such as information about the feasible solutions and the capacity restrictions, among others. Pos4 expresses the relation between new and existing facilities through a distance function or by assigned costs. Finally, Pos5 is a description of the objective function. According to the authors, this classification can be used to precisely describe all location models.

3.3 Location Problems

3.3.1 Weber problem

The Weber problem is a classic problem of location theory and constitutes the basis of many developments in this area. An extensive review and historical perspective of this problem, initially formulated by Pierre de Fermat (1601-1665) or by Battista Cavallieri (1598-1647) and solved by Evangelist Torricelli (1608-1647), is provided by Drezner *et al.* (2002). In the early 20th century, Weber returned to this problem trying to find the optimal location for a manufacturing plant with the objective to minimize the transportation cost of raw material.

Fermat initially formulated the problem as "*given three points in the plane, find a fourth point such that the sum of its distances to the tree given points is a minimum*". It can be stated as:

$$\min_{x,y}\left\{W(x,y)=\sum_{i=1}^{n} w_i d_i(x,y)\right\} \qquad (3.1)$$

where $d_i(x,y)=\sqrt{(x-a_i)^2+(y-b_i)^2}$ is the Euclidean distance between (x, y) and n fixed points with coordinates (a_i, b_i) and w_i are the weights associated with the fixed points (Drezner *et al.*, 2002).

Some of the several names that have been used to designate this problem are: the Fermat problem, the generalized Fermat problem, the Fermat-Torricelli problem, the Steiner problem, the generalized Steiner problem, the Steiner-Weber problem; the Fermat-Weber problem, one median problem, the median center problem, the minisum problem, the minimum aggregate travel point, the bivariate median problem and the spatial median problem (Drezner *et al.*, 2002).

3.3.2 Multifacility Weber Problem

The Multifacility Weber problem is the location problem in which several facilities each producing a different product (or rendering a different service) are to be located in order to minimize the sum of weighted distances between all facilities and all users as well as between the facilities.

If we have m new facilities and the weight between facility j and demand point i is w_{ij} and between facility j e s is v_{js}, then we have:

$$\min_{(x_j,y_j)j=1,\dots,m}\left\{\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}\sqrt{(x_j-a_i)^2+(y_j-b_i)^2}+\sum_{j=1}^{m-1}\sum_{s=j+1}^{m} v_{js}\sqrt{(x_j-x_s)^2+(y_j-y_s)^2}\right\} \qquad (3.2)$$

The Multifacility Weber Problem is a convex optimisation problem with a nondifferentiable objective function since two objects (demand points or facilities) may coincide, i.e., have the same location (Drezner *et al.*, 2002).

3.3.3 Location-allocation Problems

In location-allocation problems, the demand points have fixed total demands which have to be optimally allocated to the facilities (Cooper, 1963). An example of a simple location-allocation problem is:

$$\min_{\substack{(x_j,y_j)j=1,...m \\ w_{ij};i=1,...,n;j=1,...,m}} \left\{ \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}\sqrt{(x_j-a_i)^2+(y_j-b_i)^2} \right\} \tag{3.3}$$

subject to

$$\sum_{j=1}^{m} w_{ij} = w_i \quad i=1,...,n$$

$$w_{ij} \geq 0, \quad \forall\ i=1,...,n; \quad j=1,...,m$$

A specific location allocation problem is the Multisource Weber Problem in which several facilities producing the same product are to be located in order to minimize the sum of weighted distances from all users to their closest facility (Hansen *et al.*, 1998). The Multisource Weber Problem can be formulated as follows:

$$\min_{x_i,y_i,z_{ijj}} \sum_{i=1}^{p}\sum_{j=1}^{n} z_{ij}\cdot w_j \cdot d_j(x_i,y_i) \tag{3.4}$$

subject to

$$\sum_{i=1}^{p} z_{ij} = 1\ , \quad j=1,\ 2,\ ...,n$$

$z_{ij} \in [0,1]$

i = 1, 2, …, p

j = 1, 2, …n

where *p* facilities must be located to satisfy the demand of *n* users, x_i y_i denote the coordinates of the i^th^ facility, d_j the Euclidean distance from (x_i, y_i) to the j^th^ user, w_j the demand (or weight) of the j^th^ user and z_{ij} the fraction of this demand which is satisfied from the i^th^ facility. There is always an optimal solution with all $z_{ij} \in \{0, 1\}$, i.e., each user satisfied from a single facility and for that reason the problem is also called location-allocation problem.

3.3.4 *p*-Median

The *p*-median problem has been widely addressed in literature. It is a classic optimisation problem that involves the location of facilities in such a manner that the total weighted distance of all users to their closest facility is minimized. The *p*-median problem has received widespread attention because it is appropriate for many facility location decisions and forms the basis for more complex problems.

The p-median location problem was originally defined by Hakimi (1964, 1965) on a network of nodes and arcs. The 1-median problem on a plane (continuous feasible space) is the Weber problem (Hale & Moberg, 2003) mentioned in section 3.3.1.

Hakimi (1964, 1965) proved that for network instances of the problem, an optimal solution exists for which all of the facility locations are at nodes on the network. This discovery enabled him to formulate the network version of the *p*-median problem as a binary integer program (Hakimi, 1965).

The recognition (decision) form of the *p*-median problem is NP-complete[1] but the optimisation form seems to be NP-hard. The proof is supplied by Kariv & Hakimi (1979) through a polynomial time reduction from the dominating set problem which is known to be NP-complete. Incidentally, Kariv and Hakimi give the proof for NP-hardness of the optimisation form then concluding the NP-completeness of the recognition form in a remark that follows the proof.

Limiting potential facility locations to network nodes, however, reduces the number of possible location configurations to:

$$\binom{N}{P} = \frac{N!}{P!(N-P)!} \tag{3.5}$$

[1] A problem NP-complete belongs to the complexity class of decision problems for which answers can be checked for correctness by an algorithm whose run time is polynomial in the size of the input and no other NP problem is more than a polynomial factor harder. Informally, a problem is NP-complete if answers can be verified quickly and a quick algorithm to solve this problem can be used to solve all other NP problems quickly.

where N represents the number of nodes in the network and P the number of facilities (Owen & Daskin, 1998).

In 1970, ReVelle and Swain formulated the *p*-median problem as an integer programming model with the following objective function:

$$\mathrm{Min}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i \cdot d_{ij} \cdot x_{ij} \tag{3.6}$$

The model formulation is based in the following notation:

p – number of facilities to be located

n – total number of demand nodes

a_i – the amount of demand at node i

d_{ij} – distance or time between demand area i and site j

i, I – index and set of demand areas, usually nodes of a network

j, J – index and set of facility sites, usually nodes of a network

$$x_{ij} = \begin{cases} 1 \textit{ if demand i assigns to facility at j} \\ 0 \textit{ if not} \end{cases}$$

$$y_j = \begin{cases} 1 \textit{ if site j is selected for a facility} \\ 0 \textit{ if not} \end{cases}$$

Subject to the following constraints:

$$\sum_{j=1}^{n} x_{ij} = 1,\ i=1, 2, ..., n \tag{3.7}$$

$$x_{ij} \leq y_j,\ \ i, j = 1, 2, ..., n \tag{3.8}$$

$$\sum_{j=1}^{n} y_j = p \tag{3.9}$$

$$x_{ij}, y_j \in \{0,1\},\ i,j = 1, 2, ..., n \tag{3.10}$$

The first constraint ensures that each demand point must assign to at least one facility site. Since weighted distance will only increase with multiple assignments, this constraint will hold as en equality in any optimal solution. Constraint (3.8) maintains that any such assignment must be made to only those sites that have been selected for a facility. Constraint (3.9) restricts the placement/allocation of facilities to exactly *p*. The final constraint lists the integer requirements for the y_j variables. Since at least one optimal solution to a network problem is comprised of a set of nodes, the above formulation can be used to identify the global optimal solution for a network-based application (Church & Sorensen, 1994).

A reformulation of the classic *p*-median problem named COBRA was presented by Church (2003), built on the notion that many assignment variables in the classic model are redundant. A new variable substitution procedure was introduced based in a property associated with geographical proximity. This property allows making variable substitutions and reducing the size of the problem in terms of the number of variables and constraints needed in the classic *p*-median location model formulation. The reductions can be at least 10% and sometimes as much as 80% of the original variables (Church, 2003).

3.3.5 *p*-Center

While the *p*-median problem consists in locating *p* facilities so as to minimize the sum of weighted distances from the demand point to its closest facility, the *p*-center problem consists in locating *p* facilities so as to minimize the largest distance from a demand point to its nearest facility (Bespamyatnikh *et al.*, 2002).

The *p*-center problem is one of the best-known NP-hard discrete location problems and is considered harder to solve than the *p*-median problem (in the discrete case). The *p*-median instances can be easily solved exactly with up to 500 demand nodes, while for the *p*-center the largest instance considered has 75 demand nodes and was solved heuristically (Mladenovic´ *et al.*, 2003).

This problem can be formulated as follows:

$$\text{Min } W \tag{3.11}$$

subject to

$$\sum_{j \in J} x_j = p \tag{3.12}$$

$$\sum_{j \in J} y_{ij} = 1, \ \forall\, i \in I \tag{3.13}$$

$$y_{ij} - x_j \leq 0, \ \forall\, i \in I, j \in J \tag{3.14}$$

$$W - \sum_{j \in J} h_i d_{ij} y_{ij} \geq 0, \quad \forall\, i \in I \tag{3.15}$$

$$x_j \in \{0,1\}\ ,\ \forall\ j \in J \tag{3.16}$$

$$y_{ij} \in \{0,1\}\ ,\ \forall\ i \in I,\ j \in J \tag{3.17}$$

where W is the maximum distance between a demand node and the facility to which it is assigned, I is the set of demand nodes indexed by i; J is the set of candidate facility locations indexed by j, h_i is the demand at node I and p is the number of facilities to locate.

The objective function (3.11) minimises the maximum demand-weighted distance between each demand node and its closest open facility. Constraint (3.12) stipulates that p facilities are to be located. Constraint set (3.13) requires that each demand node is assigned to exactly one facility. Constraint set (3.14) restricts demand node assignments only to open facilities. Constraint set (3.15) defines the lower bound on the maximum demand-weighted distance, which is being minimised. Constraint set (3.16) established the decision variable as binary. Constraint set (3.17) requires the demand at a node to be assigned to only one facility.

In a network, the “vertex” p-center problem restricts the set of candidate facility sites to the nodes of the network while the “absolute” p-center problem allows the facilities to be anywhere along the arcs. Both versions can be either weighted (the distances between demand nodes and facilities are multiplied by a weight associated with the demand node formulation) or unweighted (all demand nodes are treated equally) (Current *et al.*, 2002).

3.3.6 *p*-Dispersion Problem

The p-dispersion problem differs from the above problems in two ways, since it is concerned only with the distance between new facilities and the objective is to maximise the minimum distance between any pair of facilities (Current *et al.*, 2002).

Considering M a large constant (e.g. max $\{d_{ij}\}$) and D the minimum separation distance between any pair of facilities, this problem can be formulated as follows:

$$\text{Max } D \tag{3.18}$$

subject to

$$\sum_{j \in J} x_j = p \qquad (3.19)$$

$$D + (M - d_{ij})\, x_i + (M - d_{ij})\, x_j \leq 2M - d_{ij} \qquad \forall\, i, j \in J,\ i < j \qquad (3.20)$$

$$x_j \in \{0,1\} \qquad \forall\, j \in J \qquad (3.21)$$

where *I* is the set of demand nodes indexed by *i*; *J* is the set of candidate facility locations indexed by *j, p* is the number of facilities to locate and d_{ij} is the distance between demand node *i* and candidate site *j*.

The objective function (3.18) maximises the distance between the two closest facilities. Constraint (3.19) requires that *p* facilities are located. Constraint (3.21) is a standard integrality constraint. Constraint (3.20) defines the minimum separation between any pair of open facilities.

3.3.7 Capacitated Plant Location Problem or Fixed Charge Location Problem

The *p*-median problem makes three important assumptions that may not be appropriate for certain location scenarios. It assumes that each potential site has the same fixed costs for locating a facility. It also assumes that the facilities being sited do not have capacities on the demand that they can serve (uncapacitated problem). Finally, it also assumes that one knows, *a priori*, how many facilities should be open (*p*) (Current *et al.*, 2002).

The Fixed Charge Location Problem or Capacitated Plant Location Problem (CPLP) relaxes all of these constraints. So, there is a set of potential locations for plants with fixed costs and capacities, and a set of customers, with demand for goods supplied from these plants. The transportation cost per unit for goods supplied from the plants to all the customers is given. The problem is to find the subset of plants that will minimize the total fixed and transportation costs such that demand of all customers can be satisfied without violating the capacity constraints of the plants (Ghiani *et al.*, 2002).

Two stages in the decision process can be identified: in the first stage a choice is made on the subset of the plants to be opened and in the second stage the assignment of the customers to these plants is made (Sridharan, 1995).

The CPLP can be formulated as follows:

$$\min \sum_{j \in J} f_j x_j + \alpha \sum_{i \in I} \sum_{j \in J} h_i d_{ij} y_{ij} \tag{3.22}$$

subject to

$$\sum_{j \in J} y_{ij} = 1 \qquad \forall\ i \in I \tag{3.23}$$

$$y_{ij} - x_j \leq 0, \ \ \forall\ i \in I, j \in J \tag{3.24}$$

$$\sum_{j \in J} h_i y_{ij} - C_j x_j \leq 0, \qquad \forall\ i \in I \tag{3.25}$$

$$x_j \in \{0,1\}\ ,\ \forall\ j \in J \tag{3.26}$$

$$y_{ij} \in \{0,1\}\ ,\ \forall\ i \in I, j \in J \tag{3.27}$$

where I is the set of demand nodes indexed by i; J is the set of candidate facility locations indexed by j, f_j is the fixed cost of locating a facility at candidate site j, C_j is the capacity of a facility at candidate site j, α is the cost per unit demand per unit distance, h_i is the demand at node I and d_{ij} is the distance between demand node i and candidate site j.

The objective function (3.22) minimises the sum of the fixed facility location costs and the total travel costs for demand to be served. The second set of terms in the objective function is often referred to as demand-weighted distance. Constraint (3.25) prohibits the total demand assigned to a facility from exceeding the capacity of the facility, C_j. Constraint sets (3.23), (3.24), (3.26) and (3.27) function as similar constraint sets in the previous problems. Relaxing constraint (3.27) allows demand at a node to be assigned (partially) to multiple facilities.

When there is an additional restriction on CPLP that each customer is served only from a single plant, the Capacitated Plant Location Problem with Single Source constraints (CPLPSS) is obtained.

There is also the capacitated plant location problem with multiple facilities in the same site (CPLPM), a special case of the CPLP, in which several identical facilities can be opened in the same site and has been studied for the location of polling stations (Cappanera *et al.*, 2004).

3.3.8 Uncapacitated Facility Location Problem

The uncapacitated facility location problem (UFLP) can be stated as follows: "*Given n possible sites and demands at m locations, determine the optimal location of facilities to fulfil all demands such that the total cost of establishing the facilities and fulfilling the demands (distribution cost) is minimized*" (Al-Sultan & Al-Fawzan, 1999).

Adding a fixed cost to the *p*-median objective function and removing the constraint that dictates the number of facilities to be located enables us to formulate the uncapacitated fixed charge problem. The result is a problem that determines endogenously the number of facilities to locate and places them so as to minimize total (fixed plus transportation) costs. The close relationship between the two problem formulations results in a large degree of similarity between the algorithms used to solve them (Hamacher & Nickel, 1996).

The UFLP can be formulated as follows:

$$\text{Min} \sum_{i\in I}\sum_{j\in J} c_{ij} y_{ij} + \sum_{j\in J} f_j x_j \tag{3.28}$$

subject to

$$\sum_{j\in J} y_{ij} = 1 \qquad \forall\ i \in I \tag{3.29}$$

$$y_{ij} \le x_j\ ,\ \ \forall\ i \in I,\ j \in J \tag{3.30}$$

$$x_j \in \{0,1\}\ ,\ \forall\ j \in J \tag{3.31}$$

$$y_{ij} \in \{0,1\}\ ,\ \forall\ i \in I,\ j \in J \tag{3.32}$$

where I is the set of demand nodes indexed by i; J is the set of candidate facility locations indexed by j, f_j is the fixed cost of locating a facility at candidate site j, c_{ij} is the total variable cost of serving all user i's demand from facility j and x_j and y_{ij} are the decision variables.

3.3.9 Covering Problems

Covering problems involve locating facilities in order to cover all or most demand within some desired service distance (often called the maximum service distance). A demand is said to be covered if it can be served within a specified time (Owen & Daskin, 1998). The covering problems are divided into two major segments: the ones in which coverage is required and where it is optimised. Two examples are the location set covering problem and the maximal covering problem (both NP-complete for general networks) (Plastia, 2002).

In the **set covering problem**, the objective is to minimize the cost of facility location such that a specified level of coverage is obtained. This formulation makes no distinction between nodes based on demand size. Each node, whether it contains a single customer or a large portion of the total demand, must be covered within the specified distance, regardless of cost. If the coverage distance is small, relative to the spacing of demand nodes, the coverage restriction can lead to a large number of facilities being located (Owen & Daskin, 1998; Rooij, 2000).

The set covering problem allows us to examine how many facilities are needed to guarantee a certain level of coverage to all customers and can be formulated as follows:

$$\min \sum_{j \in J} x_j \tag{3.33}$$

subject to

$$\sum_{j \in N_i} x_j \geq 1 \qquad \forall\ i \in I \tag{3.34}$$

$$x_j \in \{0,1\} \qquad \forall\ j \in J \tag{3.35}$$

where *I* is the set of demand nodes indexed by *i*; *J* is the set of candidate facility locations indexed by *j*, N_i is the set of all candidate locations that can cover demand point *i* and x_j is the decision variable.

The objective function (3.33) minimises the number of facilities located while constraint set (3.34) ensures that each demand node is covered by at least one facility. Constraint set (3.35) enforces the yes or no nature of the decision (Current *et al.*, 2002).

The objective function of the set covering location model can be generalized by including site-specified costs as coefficients of the decision variables. In this case, the objective would be to minimise the total fixed cost of the facilities configuration rather that the number of facilities sited (Current *et al.*, 2002).

The **maximal covering location problem (MCLP)** seeks to maximize the amount of demand covered within the acceptable service distance by locating a fixed number of facilities. In a classical sense, a demand point is assumed to be covered completely if located within the critical distance of the facility and not covered at all outside of the critical distance.

This problem can be formulated as follows:

$$\max \sum_{i \in I} h_i z_i \tag{3.36}$$

subject to

$$\sum_{j \in N_i} x_j - z_i \geq 0 \quad \forall\ i \in I \tag{3.37}$$

$$\sum_{j \in J} x_j = p \tag{3.38}$$

$$x_j \in \{0,1\}\ ,\ \forall\ j \in J \tag{3.39}$$

$$z_i \in \{0,1\}\ ,\ \forall\ i \in I \tag{3.40}$$

where I is the set of demand nodes indexed by i; J is the set of candidate facility locations indexed by j, N_i is the set of all candidate locations that can cover demand point i, h_i is the demand at node i, p is the number of facilities to locate and z_i is the decision variable:

$$z_i = \begin{cases} 1 \ \textit{if demand i is covered} \\ 0 \ \textit{if not} \end{cases}$$

The objective function (3.36) maximises the total demand covered. Constraint set (3.37) ensures that demand at node i is not counted as covered unless we locate a facility at one of the candidate sites that covers node i. Constraint (3.38) limits the number of facilities to be sited. Constraints sets (3.39) and (3.40) reflect the binary nature of the decision and demand node coverage, respectively.

Since the optimal solution to a MCLP is likely sensitive to the choice of the critical distance, determining a critical distance value when the coverage does not change in an abrupt way from "fully covered" to "not covered" at a specific distance may lead to erroneous results (Karasakal & Karasakal, 2004). The approach used by Karasakal & Karasakal (2004) assumes that the demand point can be fully covered within the minimum critical distance, partially covered up to a maximum critical distance, and not covered outside of the maximum critical distance.

Both the set covering and the maximal covering problem formulations assume a finite set of potential facility sites. Typically the set of potential sites consists of some (if not all) of the demand nodes of the underlying network. Research extensions to these models have shown that even if facilities are allowed anywhere on the network, the problem can be reduced to one with finite choices for facility location (Owen & Daskin, 1998).

In the set covering location model the capacity of each facility is assumed to be unlimited. When the capacity is limited, the problem turns over into the **capacitated set covering location problem (CSCLP)**, which may be formulated as follows: "*determine the number of facilities and their locations such that all demand points are covered with the restrictions that the coverage distance and the capacity of the centres are limited*" (Rooij, 2000).

3.3.10 Hub Location Problem

Many logistic systems such as airline networks and inter-modal carriers employ hub systems, designed to utilize larger capacity or faster vehicles over the long-haul portion of an origin to destination delivery. These systems reduce transportation cost per distance or total delivery time (Campbell *et al.*, 2002).

The basic hub location model can be formulated as:

$$\text{Min} \sum_{i \in N} \sum_{j \in N} h_{ij} \left(\sum_{k \in N} c_{ik} y_{ik} + \sum_{m \in N} c_{jm} y_{jm} + \alpha \sum_{k \in N} \sum_{k \in N} c_{km} y_{ik} y_{jm} \right) \tag{3.41}$$

subject to

$$\sum_{j \in N} x_j = p \tag{3.42}$$

$$\sum_{j \in J} y_{ij} = 1, \quad \forall\ i \in I \tag{3.43}$$

$$y_{ij} - x_j \leq 0, \quad \forall\ i \in I, j \in J \tag{3.44}$$

$$x_j \in \{0,1\}, \ \forall\ j \in J \tag{3.45}$$

$$y_{ij} \in \{0,1\}, \ \forall\ i \in I, j \in J \tag{3.46}$$

where h_{ij} is the number of unit of flow between nodes *i* and *j*, c_{ij} is the unit cost of transportation between nodes *i* and *j*, α is the discount factor for transport between hubs and x_j and y_{ij} are the decision variables.

The objective function (3.41) minimises the sum of the cost of moving items between a non-hub node and the hub to which the node is assigned, the cost of moving from the final hub to the destination of the flow and the inter-hub movement cost which is discounted by a factor of α. The constraints are similar to the other location problems, especially to the *p*-median problem.

3.3.11 Obnoxious Facility Location Problems

The previous location problems involve the location of desirable facilities; the ones that customers try to have it as close as possible. However, there are also obnoxious location problems in which customers no longer consider the facility desirable or friendly, but instead avoid the facility and stay away from it. An obnoxious facility could be e.g., a nuclear plant, a military depot or a garbage dump. Quite often the classification of obnoxious facilities is originated by the NYMBY (Not in My Back Yard) syndrome. During the last years location problems involving obnoxious facilities have received an increasing interest (Giannikos, 1998; Cappanera, 1999; Burkard *et al.*, 2000; Cappanera *et al.*, 2004).

One way to model such obnoxious facilities would be to allow negative weights on the vertices of the graph and attribute them to those vertices for which the facilities to be located are obnoxious. Clearly, the general versions of these problems are NP-hard because they contain the classical facility location problems (where all weights are nonnegative) as special cases (Burkard *et al.*, 2000).

A review of obnoxious facility location problems in the plane and particularly on networks is presented by Cappanera (1999). Some objective functions of these location problems are: i) maximin (maximise the distance between the facilities and the demand nodes) the so called *p*-dispersion problem, ii) maxisum (maximise the sum of distances between the facilities and the demand nodes) – the so called antimedian problem or maxian problem; iii) maximise the minimum weighted square distance; iv) maximise the sum of the logarithms of Euclidean distances.

The formulation of the maxisum problem, one of the most common, is the following:

$$\mathrm{Max}\sum_{i\in I}\sum_{j\in J} h_i d_{ij} y_{ij} \tag{3.47}$$

subject to

$$\sum_{j\in J} x_j = p \tag{3.48}$$

$$\sum_{j\in J} y_{ij} = 1 \quad \forall\ i \in I \tag{3.49}$$

$$y_{ij} - x_j \le 0,\ \forall\ i \in I,\ \forall\ j \in J \tag{3.50}$$

$$\sum_{k=1}^{m} y_i[k]_i - x[m]_i \ge 0, \qquad \forall\ i \in I,\ \ m = 1,\dots,N-1 \tag{3.51}$$

$$x_j \in \{0,1\}\ ,\ \forall\ j \in J \tag{3.52}$$

$$y_{ij} \in \{0,1\}\ ,\ \forall\ i \in I,\ \forall\ j \in J \tag{3.53}$$

where I is the set of demand nodes indexed by i; J is the set of candidate facility locations indexed by j, h_i is the demand at node i, p is the number of facilities to locate and d_{ij} is the distance between demand node i and candidate site j.

The objective function (3.47) aims to maximise the demand-weighted total distance. This objective forces demands to be assigned to the most remote facility, so the constraint (3.51) ensures that demands are assigned to the nearest facility. In this constraint, $[k]_i$ is the index of the k^{th} farthest candidate location from demand node *i*. Constraint (3.51) then states that if the m^{th} closest facility to demand node *i* is opened then demand node *i* must be assigned to that facility or to a closer facility.

3.3.12 Bicriteria Location or Push-pull Problems

Bicriteria location or push-pull problems simultaneously consider the minisum and minimax objective function (Ohsawa, 2000; Hamacher *et al.*, 2002; Krarup *et al.*, 2002; Plastia, 2002; Melachrinoudis & Xanthopulos, 2003). In these problems, the facility possesses both desirable and undesirable characteristics. There are attracting and repelling forces: customers or users may either try to attract (or pull) desirable facilities closer to them, or repel (or push) undesirable facilities from them (Eiselt & Laport, 1995; Current *et al.*, 2002; Krarup *et al.*, 2002).

An example is the location of an airport that is a desirable facility in the sense of needed accessibility, yet it is undesirable because of the associated noise. Fire stations, police stations, and hospitals can also be considered as such facilities, because they cause congestion and noise. Planners locating these facilities may wish to study the tradeoff between minimizing the distance to the farthest inhabitant and maximizing the distance to the nearest inhabitant – "*close but not too close*".

Bicriteria location or push-pull problems include, for example, the uncapacitated facility location problem with additive noxious effects, the uncapacitated facility location problem with minimal covering, the minsum-minsum problem, the minsum-maxmin problem, the quadratic assignment problem, the quadratic knapsack problem, the *p*-dispersion problem and the *p*-defense-sum problem (Krarup *et al.*, 2002).

Krarup *et al.* (2002) reviewed different approaches used in push-pull problems, distinguishing these strategies as the following ones:

1. Treat the bi-objective problem as really bi-objective and derive (approximations to) the Pareto-set;
2. Fix a bound for the desirable objective part and optimise the undesirable objective, what leads to a standard pull objectives with additional restrictions, either around a set of subjects or between facilities;
3. Fix a bound for the desirable objective part and optimise the undesirable objective;
4. Combine the push and pull objectives into a single objective.

3.3.13 Summary

Table 3.2 presents a summary of some of the most important location problems studied in the existing literature, highlighting the objective function used in those problems.

Table 3.2 – Objective function used in some location problems.

Problem	Objective function	Reference
Weber problem	Find the "minisum" point which minimizes the sum of weighted Euclidean distances from itself to *n* fixed points	Drezner *et al.*, (2002)
Location-allocation problems	Minimise the sum of distance, optimising the fixed total demands allocated to the facilities	Cooper (1963)
Location-routing models	Minimise the weighted sum of distances and minimise the maximum distance	Berman *et al*. (2002)
p-median	Minimise the demand-weighted total distance between demand nodes and the facilities to which they are assigned	Hakimi (1964;1965)
p-center problem	Minimise the maximum distance that demand is from its closest facility given that we are locating a pre-determined number of facilities.	Hakimi (1964;1965) Mladenovic´ *et al.* (2003)
p-dispersion problem	Maximise the minimum distance between any pair of facilities	Current *et al.*, (2002)
Fixed charge location problem or capacitated facility location problem	Minimise total facility and transportation costs. It determines the optimal number and locations of facilities, as well as the assignments of demand to a facility	Sridharan (1995); Murray & Gerrard (1997)
Uncapacitated facility location problem (UFLP)	Minimise the demand-weighted total distance between demand nodes and the facilities to which they are assigned. The number of facilities to be opened is not fixed beforehand and there are site dependent costs for each opened facility	Al-Sultan & Al-Fawzan (1999); Zhou (2000); Ghosh (2003)
Maximal covering location model (MCLP)	Maximise the demand that is covered. The model assumes that there may not be enough facilities to cover all of the demand nodes. If not all nodes can be covered, the model seeks the location scheme that covers the most demand	Church & ReVelle (1976)
Set covering location model	The objective is to locate the minimum number of facilities required to "cover" all of the demand nodes	Current *et al.* (2002)
Hub location problems	Minimise the sum of the cost of moving items between a non-hub node and the hub to which the node is assigned	Campbell *et al.* (2002)
Push-pull models	Simultaneously consider the minisum and minimax objective function	Krarup *et al.* (2002)

3.4 Search Methods

Location models are often extremely difficult to solve, at least optimally. Even the most basic models are computationally intractable for large problem instances (Current *et al.*, 2002). So the more usual approach is the use of heuristics to sub-optimally solve these problems. In this section the most frequent heuristics cited in the location problems literature are presented, together with some approaches to solve them optimally.

3.4.1 Local Search

Local search is perhaps the simplest among search methods. It starts with a given initial solution, as the current solution, and checks its neighbourhood for a better solution. If such solutions exist, then local search designates the best solution found in the neighbourhood as the current solution and repeats the process. In case the neighbourhood of the current solution does not contain any solution better than it, local search returns to the current solution and terminates. This method does not guarantee globally optimal solutions to most combinatorial problems (Ghosh, 2003), but generally returns relatively good quality solutions. The critical issue in local search is essentially how to choose neighbours.

In the literature, local search was used for solving the UFLP (Ghosh, 2003) and the *p*-median problem (Resende & Werneck, 2002b). Local search heuristics were also used to obtain the upper bounds for the Lagrangean relaxation for solve the capacitated *p*-median problem (Lorena & Sene, 2003).

3.4.2 Tabu Search

Tabu Search (TS) is a metaheuristic that guides local heuristic search procedures to explore the solution space beyond local optimality (Glover & Laguna, 1997), being one of the improvements on local search. It follows the basic principle of local search, moving at each iteration from the current solution to the best solution available in the neighbourhood.

In tabu search sites that were involved in recent moves to participate in a move at the current iteration are discouraged. This is done maintaining a tabu list. After each iteration, the sites involved in the move at the current iteration are put in the tabu list (they are said to achieve a tabu status) and stay there for a pre-defined number of iterations (called the tabu tenure). Sites with a tabu status cannot participate in moves during their tabu tenure, unless such moves satisfy an aspiration criterion. Once the tabu status of a site is removed, it can participate in future moves, and no permanent record of its past tabu status is maintained. The length of the tabu tenure can be set using one of three approaches: fixed, dynamic and random (Ghosh, 2003).

A study to define two key elements in tabu search methods to use on the *p*-median problem, based on the tabu list size and on an implementation on how to define the aspiration criterion, is presented by Salhi (2002).

Tabu search was used for solving the *p*-median problem (Carreras & Serra, 1999; Salhi, 2002), the UFLP (Al-Sultan & Al-Fawzan, 1999; Ghosh, 2003), the single source capacitated location problem (Cortinhal & Captivo, 2003) and the *p*-center problem (Mladenovic´ *et al.*, 2003).

The computational results show that tabu search performed better than local search for the SSCLP (Cortinhal & Captivo, 2003).

3.4.3 Greedy Randomised Adaptive Search Procedure

A greedy randomised adaptive search procedure (GRASP) possesses four basic components: a greedy function, an adaptive search strategy, a probabilistic selection procedure and a local search technique. These components are interlinked, forming an iterative method with a feasible solution constructed at each independent GRASP iteration. Each iteration consists of two phases, a construction phase and a local search phase and the best overall solution is kept as the result (Resende & Werneck, 2002a; Resende & Ribeiro, 2003).

A GRASP was used for the maximum covering problem obtaining facility placements that are nearly optimal (on the verifiable solutions the error never exceeded 2%) (Resende, 1998).

3.4.4 Interchange Heuristics

The interchange algorithm developed by Teitz & Bart (1968) starts with a set of *p* arbitrarily chosen centres. In the next step the effect of an interchange between the first of the (*m*-*p*) non-centres and each of the *p* centres is evaluated.

The algorithm underlying is as follows: an initial solution, a list of facility sites termed the current solution is compiled and the value of the objective function is calculated. The heuristic consists of two loops. The outer loop selects every candidate – a feasible facility location – on the network in sequence. As each candidate is selected, it is passed to the inner loop, which calculates the values of the objective function that would result from substituting the candidate for each of the facility sites in the current solution. On completion on the inner loop, if a substitution yields an objective function value lower than that of the current solution, the candidate site is swapped into the current solution. If more than one substitution results in lower objective function values, the candidate site yielding the lowest value is swapped into the current solution. At the end of the outer loop (the end of one iteration), if no swaps have been made the heuristics terminates and the current solution is the final-solution; if a swap has been made the heuristic starts another iteration (Teitz & Bart, 1968).

The Teitz and Bart heuristic frequently converges to the optimum solution, performing very well when compared with exact techniques and other heuristics, irrespective of problem size (Densham & Rushton, 1991).

Another interchange approach is the one implemented by Whitaker and modified by Resende & Werneck (2002b) called fast interchange heuristic. It was successfully applied to the *p*-median problem obtaining speedups of up to three orders of magnitude with respect to the best previously known implementation. This implementation is especially well suited to relatively large instances and, due to the preprocessing step, to situations in which the local search procedure is run several times for the same instance (such as within a metaheuristic) (Resende & Werneck, 2002b).

The global/regional interchange algorithm (GRIA) is another interchange method that combines node partitioning and drop-and-add (Densham & Rushton, 1991). It starts with an arbitrary selection of *p* centres. In the global phase it first drops a centre, then adds one of the remaining (*m*-*p*) candidates (greedy drop and add steps). In the regional phase it determines the median of each subset (subset median step). After, the two phases are repeated until no interchanges have to be made.

Both interchange algorithm and GRIA have been used to solve the *p*-median problem (Church & Sorensen, 1994).

3.4.5 Heuristic Concentration

Heuristic concentration is a two-stage metaheuristic that can be applied to a wide variety of combinatorial problems. It is particularly suited to location problems in which the number of facilities is given in advance. In such settings, the first stage of heuristic concentration repeatedly applies some random-start interchange (or other) heuristic to produce a number of alternative facility configurations. A subset of the best of these alternatives is collected and the union of the facility sites in them is called a concentration set. Among the component elements of the concentration set, those sites that are members of the optimal solution are likely to be included. In earlier studies, the second stage of heuristic concentration has consisted of an exact procedure to extract the best possible solution from the concentration set (Rosing & ReVelle, 1997; Rosing *et al.*, 1998; Rosing & Hodgson, 2002).

The heuristic concentration was designed to escape the traps of local optima which tend to be found by some base heuristic techniques (Rosing & ReVelle, 1997).

The heuristic concentration is a method which has been shown to arrive at better, often optimal, heuristic solutions to the *p*-median problem (Rosing & Hodgson, 2002). The computation time and solution quality obtained through the application of heuristic concentration for the *p*-median problem are only possible because of the concentrating effect of heuristic concentration on the reduction of the search space from hundreds, or perhaps thousands (the total number of demand nodes/potential facility sites) to a few tens of possible sites (Rosing & Hodgson, 2002).

The comparison between tabu search and heuristic concentration in the resolution of *p*-median problem showed best results (optimal or best known) for heuristic concentration over 80% of the cases. On the other hand, tabu search finds the optimal (or best known) in less than 10% of the cases (Rosing *et al.*, 1998).

3.4.6 Variable Neighbourhood Search

Variable Neighbourhood Search (VNS) is a metaheuristic for solving combinatorial and global optimisation problems. The basic idea subjacent is a systematic change of neighbourhood structures within a local search algorithm. The algorithm remains centred around the same solution until another solution better than the current is found and then jumps there (Hansen & Mladenovic´, 1997; Hansen & Mladenovic', 2001; Mladenovic´ *et al.*, 2003).

By exploiting the empirical property of closeness of local minima that holds for most combinatorial problems, the basic VNS heuristic has two important advantages: i) by staying in the neighbourhood of the present solution the search is done in an attractive area of the solution space which is not, moreover, perturbed by forbidden moves; and, ii) as some of the solution attributes are already in their optimal values, local search uses several times fewer iterations than if initialised with a random solution, so, it may visit several high-quality local optima in the same computational time one descent from a random solution takes to visit only one (Mladenovic´ *et al.*, 2003).

Variable neighbourhood search was used to solve the *p*-median problem (Hansen & Mladenovic´, 1997; Hansen & Mladenovic', 2003), the *p*-center problem (Mladenovic´ *et al.*, 2003) and the Multisource Weber problem (Brimberg *et al.*, 2000). Several parallelization strategies for the VNS to solve the *p*-median problem were tested by Crainic *et al.* (2001). Also García-López *et al.* (2002) considered the parallelization of the VNS and coded it in C to solve large instances of the *p*-median problem.

In a study that compared several heuristics, i.e. alternative location-allocation, projection, Tabu Search, genetic algorithms and several versions of Variable Neighbourhood Search (VNS), it was found that VNS gives consistently best results on average, in moderate computing time, when the number of facilities is large (Brimberg *et al.*, 2000). The reason why Variable Neighbourhood Search outperforms Tabu Search appears to be the effect of Tabu restrictions; they create many pseudo local minima and they make it more difficult to reach the global optimum (Hansen & Mladenovic´, 1997).

3.4.7 Scatter Search

Scatter Search is a population-based or evolutionary metaheuristic that uses a reference set to combine its solutions and construct others. It derives its foundations from strategies originally proposed for combining decision rules and constraints (in the context of integer programming) (Laguna, 2002). The method generates a reference set from a population of solutions and then a subset is selected from this reference set. The selected solutions are combined to get starting solutions to run an improvement procedure. The result of the improvement can motivate the updating of the reference set and even the updating of the population of solutions (García-López *et al.*, 2003).

Scatter search and its generalized form path relinking contrast with other evolutionary procedures, such as genetic algorithms, by providing unifying principles for joining solutions based on generalized path constructions (in both Euclidean and neighbourhood spaces) and by utilizing strategic designs where other approaches resort to randomization. Additional advantages are provided by intensification and diversification mechanisms that exploit adaptive memory (Glover, 1999).

The process of generating combinations of a set of reference solutions in Scatter Search consists of forming linear combinations that are structured to accommodate discrete requirements (such as those for integer-valued variables). The resulting combinations may be characterized as generating paths between and beyond these solutions, where solutions on such paths also serve as sources for generating additional paths (Glover *et al.*, 2004).

The initial population must be a wide set of disperse solutions. However, it must also include good solutions (Resende & Werneck, 2002a; Resende & Ribeiro, 2003). Several strategies can be applied to get a population with these properties, for instance, by using a random procedure to achieve a certain level of diversity. Then a simple improvement heuristic procedure must be applied to these solutions in order to improve them. The initial population can also be obtained by a procedure that provides simultaneously disperse and good solutions like GRASP procedures.

Scatter search was applied in parallel to solve the *p*-median problem (García-López *et al.*, 2003).

3.4.8 Simulated Annealing

Simulated Annealing is based on a strong analogy between combinatorial optimisation and the physical process of crystallisation. This process inspired Metropolis *et al.,* in 1953, to propose a numerical optimisation procedure known as the Metropolis algorithm, which works as follows. Starting from an initial situation with 'energy level' *f (0)*, a small perturbation in the state of the system is brought about. This brings the system into a new state with energy level *f (1)*. If *f (1)* is smaller than *f (0)*, then the state change is accepted. If *f (1)* is greater than *f (0)*, then the change is accepted with a certain probability. A movement to a state with a higher energy level is sometimes allowed to be able to escape from local minima.

The probability of acceptance is given by the Metropolis criterion:

$$P\,(\text{accept change}) = \exp\left(\frac{f(0) - f(1)}{s_0}\right) \qquad (3.54)$$

where s_0 is a control or freezing parameter. Next, the freezing parameter is slightly decreased and a new perturbation is made. The energy levels are again compared and it is decided whether the state change is accepted. This iterative procedure is repeated until a maximum number of iterations is reached or until change occurrences have become very rare.

The analogy with spatial optimisation assumes that physical states are replaced by *solutions* and *energy levels* by costs. Practice has shown that a sufficiently slow decrease of the freezing parameter yields in almost all cases the optimal solution (Aerts & Heuvelink, 2002).

Chiyoshi & Galvão (2000) used an algorithm that combines elements of the vertex substitution method of Teitz and Bart with the general methodology of simulated annealing for the *p*-median problem. The methodology used seemed to work well on all tested problems. Optimal solutions were obtained for 26 of the 40 problems, and although high optimum hitting rates (greater than 50%) were obtained for only 20 problems, the worst gap to the optimum was 1.62%, all runs of the 40 test problems considered.

There are other heuristics that, like simulated annealing, are based on an analogy drawn from particle physics. The key idea is to let particles represent facilities with mass proportional to the capacities and to let the particles "jostle around" and find a low energy configuration, in the hope that the configuration so found leads to a good solution to the original location-allocation problem. This is like finding an approximately stable solution for an n-particle (n-body) system with interacting forces. Hence, the name simulated n-body. Experimental results based on randomly generated graphs suggest that these heuristics outperforms simulated annealing both in cost minimization as well as execution time (Simha *et al.*, 2000).

Another approach is the double-annealing algorithm that consists of two annealing processes synchronized with each other; each process is executed on a different subset of variables and with a different annealing parameter ('temperature') and the synchronization depends on the saturation of the two variable subsets. In order to improve such synchronization, deannealing steps are also performed. The double annealing algorithm is quite robust and easy to tune (it is rather insensitive to the initial values of the annealing parameters and to the initialisation) and it is able to achieve good approximate solutions on the *p*-median problem (Righini, 1995).

3.4.9 Genetic Algorithms

Genetic algorithms constitute a problem-solving approach that uses the concept of natural adaptation and selective breeding of organisms. Holland and his associates, as well as Fogel, introduced the general idea in the 1960s, but it was not until recently that genetic algorithms became more popular among the operations researchers (Fogel, 1999; Bozkaya *et al.*, 2002).

With genetic algorithms, a chromosome, which is the encoded representation of the genetic material of an organism, corresponds to a solution in the feasible solution set of a problem. In other words, each feasible solution of a problem is encoded as a string of characters (usually binary string) and this string carries the information defining that particular solution. Each chromosome has a fitness value that corresponds to the objective function of the associated solution. The chromosome, the basic element of genetic algorithms, is the medium for carrying out the operations needed to perform a random search for a “good” solution over the solution space, imitating the natural selection process.

Initially, there is a pool (or population) of solutions (or chromosomes) that are randomly generated. Random generation of these solutions is necessary for an unbiased representation of solutions from different parts of the solution space. Next, two solutions (or parents) from this pool are selected for “mating” in order to produce two new solutions (or offspring). The selection of parents may be random or based in the fitness values with the expectation that good genetic material carried by the two parents will be transferred to the offspring.

Copying parts if the two parent chromosomes generates the chromosome of an offspring. In its most basic form, this is done by identifying a random crossover point in the parent chromosomes and exchanging the sections of the two parents chromosomes after the crossover point, which results in the generation of two offspring from two parents. This is a basic crossover operator and there are many different crossover operators that directly affect the algorithm performance (Bozkaya *et al.*, 2002).

An offspring may be subject to mutation with a given probability, for the objective of introducing genetic variety in the population and hence force the algorithm out of local optima. Following this step, the parents are replaced by the offspring in order to keep the population size constant. Alternatively, one can generate as many offspring as the population size and then replace the entire population with the offspring. These two replacement strategies (i.e. the partial and complete replacement of the pool) are called overlapping and non-overlapping replacement, respectively, and the set of new solutions that replace the parents is referred to as the new generation.

The above process can be repeated as many times as needed until a certain stopping criterion is satisfied. This criterion could be a fixed number of generations or could be related to the quality of the solutions discovered.

The first researchers to develop a genetic algorithm for the *p*-median problem were Hosage and Goodchild in 1986. They used a simple genetic algorithm with conventional genetic operators. A binary string represented each candidate solution, where each bit corresponds to a facility index. Each allele (1 or 0) indicates whether or not the corresponding facility is selected as a median. If the number of bits set to 1 is different from *p* the solution is deemed invalid and a penalty (proportional to the extent of restriction violation) is applied to the fitness of the individual. The classic binary individual representation was not very suitable for this problem, wasting memory and processing time and was discarded by the initial authors (Church & Sorensen, 1994; Correa *et al.*, 2001). Subsequent testing has shown that the process is very sensitive to the genetic encoding technique (Church & Sorensen, 1994; Bozkaya *et al.*, 2002).

Subsequently genetic algorithms have been successfully used to solve location problems. A genetic algorithm with a heuristic hypermutation operator was used to solve the *p*-median problem obtaining better results than Tabu Search (Correa *et al.*, 2001).

Three different coding/decoding techniques for genetic algorithms were tested by Chiou & Lan (2001) to solve the *p*-median problem: the simultaneously clustering method; the stepwise clustering method and the cluster seed point method. The cluster seed point method yielded good results for medium-to-large scale problems (50-200 objects) (Chiou & Lan, 2001).

A genetic algorithm with three crossover operators and the concept of “invasion” was developed by Bozkaya *et al.* (2002) for solving the *p*-median problem, demonstrating that genetic algorithms can generate good solutions to location problems.

A metaheuristic called Constructive Genetic Algorithm (CGA) was used to solve the capacitated *p*-median problem (Narciso & Lorena, 2002). The CGA shows some innovative features in relation to the traditional genetic algorithms, such as population formed by only structures and/or schemes, dynamic population, mutation in complete structures and the possibility of using heuristics in the representation of schemes and structures, which improve the computational results.

A new genetic algorithm for the *p*-median problem was proposed by Alp *et al.* (2003). This algorithm does not use some of the features common in other genetic algorithms (such as mutation), and the operator used to generate new solutions is a greedy selection heuristic as opposed to a crossover operator. It takes the union of two solutions and dropping facilities one-at-a-time to generate a feasible solution. Hence, the authors suggest it may be more accurate to call it a *hybrid evolutionary heuristic* and not genetic algorithm. In 85% of the test problems it generated solutions that were within 0.1% of the optimum and its worst solution was only 0.41% away from the optimum (Alp *et al.*, 2003)

3.4.10 Bionomic Algorithms

Bionomic algorithms, introduced by Christofides in 1994, are a class of metaheuristic techniques that share the core of their approach with genetic algorithms. They are in fact evolutionary metaheuristic algorithms that update a whole population of solutions (the solution set) at each iteration. Moreover, the updating process in all of them consists in defining a child solution from a set of parent solutions of the previous generation, where the exact definition of the child often goes through some randomisation step (Maniezzo *et al.*, 1998).

Bionomic algorithms share with the evolutionary scatter search approach of (Glover *et al.*, 2004), the possibility of having variable-sized solution sets and the use of multiple parents, whereas genetic algorithms limit the number of parents to two. On the other hand, bionomic algorithms formally require a local optimisation of the solutions (called *maturation*), an activity first introduced in the scatter search approach that was excluded from genetic algorithms until the late-1980s, though it has now become standard practice in genetic algorithms applied to combinatorial optimisation problems (Maniezzo *et al.*, 1998).

The bionomic algorithm was used to solve the capacitated *p*-median problem (Maniezzo *et al.*, 1998).

3.4.11 Lagrangean Heuristic

The basic idea behind a Lagrangean heuristic is the computation of a lower bound through a Lagrangean relaxation, one of the most widely used methods in location problems (Senne & Lorena, 2000; Ghiani *et al.*, 2002; Lorena & Sene, 2003; Cappanera *et al.*, 2004; Karasakal & Karasakal, 2004). Applying a decomposition technique can usually solve such problem, and the information obtained can be used to construct a feasible solution of the original problem.

The procedure is usually encapsulated in a subgradient optimisation algorithm, on the basis of which a sequence of Lagrangean multipliers is generated in order to achieve the highest lower bound as possible. The availability of both upper and lower bounds on the optimal objective function value is an attractive feature of

Lagrangean heuristic for two reasons. The first is that the relative gap between upper and lower bounds is typically used as a measure of the maximal error of the solution. The latter is that we can specify how close to optimal we would accept a solution. However, in most cases, these bounds are not easily obtained and usually Lagrangean relaxation is combined with other heuristics as we refer in section 3.4.16.

A Lagrangean relaxation was used to obtain lower bounds for single source capacitated location problem (Cortinhal & Captivo, 2003). In the same study Lagrangean heuristics followed by search methods and by a tabu search metaheuristic gives upper bounds for this problem.

Others authors formulated the MCLP in the presence of partial coverage, and developed a solution procedure based on Lagrangean relaxation (Pereira & Lorena, 2001; Karasakal & Karasakal, 2004).

A Lagrangean heuristic has used to solve the capacitated plant location problem with multiple facilities in the same site (Ghiani *et al.*, 2002).

Moreover, a Lagrangean relaxation was used to decompose an Obnoxious Facility Location and Routing model (OFLR) into a Location subproblem and a Routing subproblem, that were also relaxed with Lagrangean Heuristics (Cappanera *et al.*, 2004).

3.4.12 Branch and Bound

A Branch and Bound algorithm (or A*) finds a global optimum. However, explicit enumeration is normally impossible due to the exponentially increasing number of potential solutions. The use of bounds for the function to be optimized combined with the value of the current best solution enables the algorithm to search parts of the solution space only implicitly.

If we want to minimize a function f(x), where x is restricted to some feasible region (defined, e.g., by explicit mathematical constraints). To apply branch and bound, we must have a means of computing a lower bound on an instance of the optimisation problem and a means of dividing the feasible region of a problem to create smaller subproblems. There must also be a way to compute an upper bound (feasible solution) for at least some instances; for practical purposes, it should be possible to compute upper bounds for some set of nontrivial feasible regions.

At any point during the solution process, the status of the solution with respect to the search of the solution space is described by a pool of yet unexplored subset of this and the best solution found so far. Initially only one subset exists, namely the complete solution space, and the best solution found so far is ∞ (Clausen, 1999). The unexplored subspaces are represented as nodes in a dynamically generated search tree, which initially only contains the root, and each iteration of a classical Branch and Bound algorithm processes one such node. The iteration has three main components: selection of the node to process, bound calculation, and branching.

Branch-and-bound is a well-known technique largely applied to many location problems and was used to solve a multi-level network optimisation model (Cruz *et al.*, 2003).

3.4.13 Voronoi Diagrams

The Voronoi diagram is a very simple diagram. Given a set of two or more, but finite number of distinct points in the Euclidean plane, all locations in that space are associated with the closest member(s) of the point set with respect to the Euclidean distance. The result is a tessellation of the plane into a set of the regions associated with the members of the point set.

All of the Voronoi regions are convex polygons. The boundary between two adjacent regions is a line segment, and the line that contains it is the perpendicular bisector of the segment joining the two sites. Usually, Voronoi regions meet three at time at Voronoi points. If three sites determine Voronoi regions that meet at a Voronoi point, the circle through those three sites is centered at that point, and there are no other sites in the circle.

Location problems like the continuous *p*-median problem; the constrained *p*-median problem and the continuous *p*-center problem have been solved through Voronoi diagrams (Suzuki & Okabe, 1995).

A very complete explanation of the application of the ordinary Voronoi diagram, the farthest-point Voronoi diagram, the weighted Voronoi diagram, the network Voronoi diagram, the Voronoi diagram with a convex distance function, the line Voronoi diagram and the area Voronoi diagram to eight different types of continuous location problems can be found in Okabe & Suzuki (1997).

The nearest and farthest-point Voronoi diagrams were used in a single facility bicriteria location model associated with maximin and minimax criteria in the Euclidean plane (Ohsawa, 2000).

Competitive location problems have also been solved through the Voronoi game (Ahn *et al.*, 2004). Two players, White and Black, who place a specified number, n, of facilities in a region, play this game. They alternate placing their facilities one at a time, with White going first (as in Chess). After all 2n facilities have been placed, their decisions are evaluated by considering the Voronoi diagram of the 2n points. The player whose facilities control the larger area wins.

3.4.14 Clustering Algorithms

Clustering algorithms have been employed in many fields of human knowledge that require finding a “natural association” among some specific data (Procopiuc, 1997). Some location problems constitute clustering problems so there are some examples in the literature of the application of clustering algorithms to sub-optimally solve these problems.

A new local search heuristic called j-means can be applied to the Multisource Weber Problem. This heuristic was compared with k-means, h-means[1] and VNS,

1 H-means is a heuristic that first selects an initial partition, computes the centroids of its clusters, then reassigns entities to the closest centroid and iterates until stability is reached. It is different from k-means that is an interchange heuristic, where points are reassigned to a cluster other than their own, one at a time, until a local optimum is reached.

and the results on standard test problems from the literature were promising (Hansen & Mladenovic', 2001).

Estivill-Castro & Houle (2001) used a medoid-based clustering method that incorporates both proximity and density information using the Voronoi diagram of the data points and its dual, the Delaunay triangulation. Medoid-based clustering is similar to mean-based clustering, but it allows only data points to be chosen as representatives. This method was used to solve the discrete *p*-median problem and showed robustness with respect to random initialisation, additive noise and multiplicative noise.

3.4.15 Neural Networks

Neural networks are, in a simple way, networks where the nodes correspond to neurons and the arcs correspond to synaptic connections in the biological metaphor. Neural networks are powerful tools in applications where formal analysis would be difficult or impossible, such as pattern recognition and nonlinear system identification and control (Bishop, 1996; Frasconi *et al.*, 1997; Bicciato *et al.*, 2001).

Inspired by the biological nervous system, neural networks are being used to solve a wide variety of complex scientific, engineering, and business problems. Commercial applications include investment portfolio trading, data mining, process control, noise suppression, data compression, and speech recognition. Neural networks are suited for such problems because, like their biological counterparts, a neural network can learn from experience or data, and therefore can be trained to find solutions, recognize patterns, classify data and forecast events.

Unlike analytical approaches commonly used in fields such as statistics, neural networks require no explicit model and no limiting assumptions of normality or linearity. The behaviour of a neural network is defined by the way its individual computing elements are connected and by the strength of those connections, or weights. The weights are automatically adjusted by training the network according to a specified learning rule until it properly performs the desired task.

Supervised neural networks are trained to produce desired outputs in response to example inputs while unsupervised neural networks are trained by letting the network continually adjust itself to new inputs.

Self-Organizing Maps (SOM) or Kohonen maps (Kohonen, 2001) are structured artificial neural networks with the capability of mapping high dimensional input patterns into an ordered array of competing output units so as to capture the global structure of the input space through the use of local adaptation rules (Lozano *et al.*, 1998). It's an unsupervised neural network that tries to preserve topological relations between points in the input and output spaces.

SOM provide a visual representation of a vector quantification algorithm that places a number of vectors into a high-dimensional input data space in such a way that they approximate the original data patterns in an ordered fashion (Kohonen *et al.*, 1995). SOM were used to solve location-allocation problems (considering that the demand is spread over a given continuous area according to a given probability) (Lozano *et al.*, 1998; Hsieh & Tien, 2004) and uncapacitated location-allocation problems (with discrete demand) (Hsieh & Tien, 2004). These location problems can be treated as clustering problems.

SOM have shown better results than hierarchical clustering methods and are less time consuming (Mangiameli *et al.*, 1996). Comparing SOM with Simulated Annealing, Hsieh & Tien (2004) have obtained better experimental results with SOM for large location-allocation problems.

3.4.16 Hybrid Heuristics

A very usual approach to solve location problems is the combination of heuristics, also called hybrids, to improve the results (better approximations to optimal solutions) and decrease the computational time for problems with increasing number of demand points.

Pérez *et al.* (2003) created the variable neighbourhood tabu search (VNTS), consisting of a combination of the variable neighbourhood search (VNS) and tabu search, and used it for solving generic location–allocation problems, namely the median cycle problem[1].

Another combination using variable neighbourhood search and decomposition called Variable Neighbourhood Decomposition Search (VNDS) was used on the *p*-median problem (Hansen *et al.*, 2001). This method uses a sequence of subproblems (problems of smaller sizes than the initial one) that are generated from the different preselected set of neighbourhoods. If the solution of the subproblem does not lead to an improvement in the whole space, the neighbourhood is changed. Otherwise, the search continues from the solution in the first pre-selected neighbourhood. The process is iterated until some stopping condition is met.

Another approach used for UFLP consists in the application of the Greedy–Interchange heuristic using a small subset of candidate facilities, and application of the newly developed heuristic named Balloon Search[2] (Hidaka & Okano, 2003). The authors concluded that a Greedy heuristic improved the total cost by 9%–11%, that the Interchange heuristic improved the total cost by an additional 0.5%–1.5%, and that Balloon Search improved it by a further 0.5%–1.5%.

The *p*-median problem for instances up to 3795 nodes has been solved using a Branch-and-Cut-and-Price algorithm that combines the Lagrangean relaxation, preprocessing, a column-and-row generation approach to solve LP-relaxation and cutting planes (Avella *et al.*, 2003).

[1] The connection structure is a cycle visiting a fixed depot and the allocation structure is of star type. One version of this problem consists in finding the cycle that visits the depot and minimizes the sum of connection and allocation costs. Another version consists in finding the length of a tour passing through a depot while imposing an upper bound on the sum of the distances from all the users to the cycle (Pérez *et al*., 2003).

[2] The optimal location is found as the median of an expanding “balloon” that includes the subset of the assigned customers or demand nodes. The word “balloon” is used for a subset of customers who are assigned to a location and who are included in the first members in ascending order of transportation cost (Hidaka & Okano, 2003).

The capacitated *p*-median problem was also solved using the Lagrangean/Surrogate[1] Approach, combined with location-allocation heuristics (Lorena & Sene, 2003). This approach has the same quality of the Lagrangean dual bound, but is obtained with modest computational times.

The Lagrangean/Surrogate approach has also been used as an acceleration process to a column generation to solve capacitated *p*-median problems, identifying new productive columns and accelerating the computational process (Lorena & Senne, 2004).

A GRASP with path-relinking[2] was also used to solve the *p*-median problem, where the best solutions found in all iterations are kept in a pool (the so called elite solutions) and subsequently are combined with the solutions obtained after the local search through path-relinking (Resende & Werneck, 2002a; Resende & Werneck, 2004).

A merger of GRASP and Adaptive Memory Programming (AMP) into a new GRAMPS framework was developed to solve the capacitated *p*-median problem (Ahmadi & Osman, in press). GRAMPS is implemented with a local search descent based on a restricted k-interchange neighbourhood. The results show that GRAMPS has an efficient learning mechanism and is competitive with the existing methods in the literature (Ahmadi & Osman, in press).

A new method for the capacitated *p*-median problem based on a set partitioning formulation of the problem was developed by Baldacci *et al.* (2002). A valid lower bound to the optimal solution cost is obtained by combining two different heuristic methods for solving the dual of the LP-relaxation of the exact formulation. The

---

[1] The surrogate relaxation is used to solve the surrogate dual problem. For given multipliers (u,v) $(u \geq 0)$, the surrogate dual objective is the relaxed mathematical program: $\text{Max}\{f(x): x \text{ in } X, ug(x) \leq 0, vh(x) = 0\}$.

[2] Path-relinking is a generalized form of scatter search that was first proposed in the context of the Tabu Search metaheuristics, but it has been also applied with a variety of other methods (Glover *et al.*, 2004). This approach generates new solutions by exploring trajectories that connect high-quality solutions, by starting from one of these solutions, called an initiating solution, and generating a path in the neighbourhood space that leads toward the other solutions, called guiding solutions. This is accomplished by selecting moves that introduce attributes contained in the guiding solutions. Path-relinking makes reference to paths between and beyond selected solutions in neighborhood space, rather than in Euclidean space as in the case of Scatter Search (Glover *et al.*, 2004).

dual solution obtained is used for generating a reduced set partitioning problem that can be solved by an integer programming solver. The solution achieved might not be an optimal CPMP solution; however the new method allows estimating its maximum distance from optimality.

3.5 Location Problems and GIS

One of the foci of developing decision support capabilities of Geographic Information Systems (GIS) has been the integration of maps with multiple criteria decision models, evaluating location alternatives on the basis of suitability criteria (Jankowski *et al.*, 2001). However, there is still a specific need to develop spatial optimisation algorithms and models integrated with GIS to support location problems and, consequently, decision support (Kim & Openshaw, 1999b).

Researchers have demonstrated that exploratory data techniques can be successfully applied to support multicriteria spatial decision making (Jankowski *et al.*, 2001). However, the selection of a heuristic procedure for the solution of a location problem in a GIS system must be based on some criteria, that include (Church & Sorensen, 1994): i) robustness of heuristic process; ii) ease of understanding; iii) speed or efficiency of technique; iv) ease of development and integration.

A good review of the history of location modelling supported by GIS as well as the obstacles to the application of location models, issues associated with the integration of location models into GIS and future needs in GIS functionality to support location models are provided by Church (1999). In this section, only some of the major issues in the integration / interaction of GIS and location models are referred.

One of the major advantages of the use of GIS in location models is the enhancement of visual exploration and comparison of results from a location model, since many attributes can be presented simultaneously with model results. The presentation of a map with demand allocated to located facilities by directed lines is one of the most popular visualisation graphs in location problems. This plot provides an easy-to-understand view of where the facilities are, which demand is served by which facility, as well as the potential area differences in service regions.

GIS can also have an important role in the definition of demand areas and facility sites and can add significant value to a given application when data gathered for another purpose are available to characterize demand or the feasibility of specific sites or regions (Church, 1999).

However, the demand areas and the representation of potential facility sites are often characterised to a relatively fine level of detail by hundreds, if not thousands, of demand points. Since many approaches cannot easily handle thousands of demand points and sites, some type of data aggregation is necessary and this can be aided by the use of GIS.

Spatial aggregation of demand is particularly relevant since GIS allow integration of digital maps with extensive databases for analysis and display (Bowerman *et al.*, 1999). Many GIS are used to convert data such as street addresses on transportation network to spatial coordinates (geocodification). Given this, in the near future it is likely that location-allocation models will be used increasingly in conjunction with GIS in situations where the actual geographic locations of each individual in a demand area are available, what may lead to very large data sets.

Accordingly, the issue of effectively handling aggregation error for facility location models is likely to become increasingly important. There is an extensive literature about aggregation errors in location problems [see, for example, Current & Schilling (1987); Andersson *et al.* (1998); Erkut & Bozkaya (1999); Zhao & Batta (2000); Hodgson & Hewko (2003)]. In the long term, the need to build in good demand point aggregation algorithms into geographic information systems is clearly desirable (Andersson *et al.*, 1998).

One of the few examples of the use of GIS in location problems is the use of simple Avenue scripts in ArcView, externally linking to the location algorithms in Fortran 77 (simulated annealing, genetic algorithms, tabu search and Monte Carlo methods) (Kim & Openshaw, 1999a).

Lorena *et al*. (1999) have also used an Avenue script to implement a Lagrangean/surrogate approach to solve the capacitated *p*-median problem in ArcView.

Other authors have proposed the incorporation of the heuristic concentration code into GIS, obtaining good solutions to the *p*-median and a range of other location problems (Rosing *et al.*, 1999).

Another recent approach is the application of the CARE algorithm[1] in ArcInfo software for the capacitated set covering location to determine the minimum number of interviewers required to cover all addresses in the Netherlands. The CARE algorithm produced good results in a reasonable time, even for large-scale (real world) networks (Rooij, 2000).

There is also LoLA (Library of Location Algorithms) that is a collection of algorithms for location theory (available at http://www.mathematic.uni-kl.de/~lola) that can be linked to ArcView through an Avenue script (Bender *et al.*, 2002).

There are several important issues related specifically to location model and GIS integration. One of them is the compatibility of data structures. There are differences between the data structure that has been designed to best support a location model algorithm and the principal data model used in a GIS. Many times, the GIS data structure is not used directly by the solution process. Hence, the exporting and importing of data and results loosely couples the solution process and the GIS (Church, 1999).

Another issue is error propagation. It is important to understand how data error may propagate in any procedure that is used and to what extent it may alter results in location models. It is necessary to analyse such occurrences, as they can diminish reliability in the final results.

According to Current (1999), the biggest issue facing the GIS community in facility location planning is that most location models and their solutions

[1] The method is named CARE algorithm after the three basic elements: center adding, center repositioning, and center elimination algorithms (Rooij, 2000).

counterparts have been developed and tested for “green-field planning problems”. Then the models are typically defined and the algorithms designed to solve the case when all the facilities are new. Even though all current GIS implementations have the capability to fix existing sites into the solutions, they do not have the capability to move existing sites, i.e. sites can only be fixed into a solution or out of a solution. This capability only supports one side of what can be called the “brown-field planning” (i.e. adding to, taking away, or transforming an existing configuration) there must be the capability to solve for a new configuration which maintains much of what currently exists and which adds or moves specific facilities to better locations. Expanding GIS location routines to be capable of supporting this type of functionality should be a high priority.

4. MODEL IMPLEMENTATION

4.1 Inventory of CCA-treated Wood Waste in Portugal

4.1.1 Characterization of Treated Wood Production

Limited information was available concerning wood preservation industry and the use of preserved wood products, as well as on disposal practices for CCA-treated wood waste in Portugal. For that reason, a questionnaire for the preservation industry was developed in this study (see Appendix I). The questionnaire inquired about the treatment process used by each plant, the quantity of treated wood produced and the type of products manufactured.

The questionnaire was sent to the 20 Portuguese wood preservation industries and the response rate obtained was 45%. The questionnaire was send directly to the industries in the Belém database (database with the Portuguese industries classified by activity) of the National Statistics Institute ("*Instituto Nacional de Estatística*"). The initial mailings were followed by telephone calls and by a second mailing in some cases. It was also distributed to the associates of the Portuguese Association of Wood and Furniture Industries ("*AIMMP – Associação das Indústrias de Madeira e Mobiliário de Portugal*") through an internal mailing of the Association.

For confidentiality reasons, we do not present the answers of the wood preservation industries individually, but only the main conclusions of the questionnaire.

The industries that answered to the questionnaire have initiated their activity between 1978 and 1995, 44% of the respondents started working in the 80's and 33% of the plants are posterior to 1990.

In relation to the preservation product, a total of 1335 Mg[1] of preservation products are consumed in Portugal per average year in the respondent industries, with a clear predominance of the different CCA formulations (Figure

[1] We use the International System (SI) of Units and Mg corresponds to 10^6 g.

4.1). The Portuguese preservation industries use mainly CCA Type 2[1] (62%), followed by CCA Type 1[2] (29%). CCA Type C[3] and other formulations of CCA, as well as CCB, are also used, but assume a minor importance. None of the respondent industries used creosote or other organic preservation products.

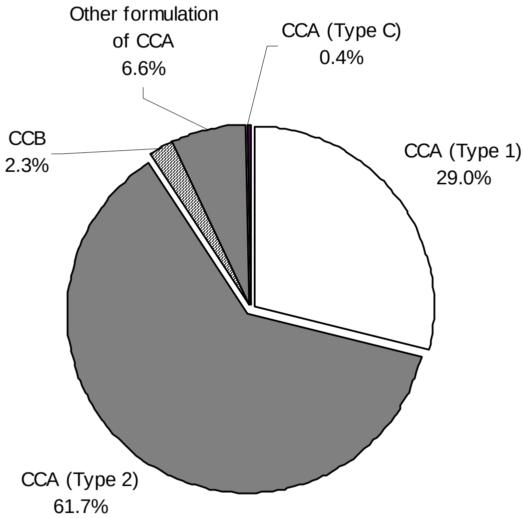

Figure 4.1 – Preservation products used in the preservation industries that answered to the questionnaire.

The majority of the preservation industries have only one pressure treatment equipment (67%), while 22% have two cylinders. The cylinder diameter varies between 1.0 m and 2.0 m and its length is between 7.0 m and 18.50 m.

In the respondent industries, the total treated wood production in 2003 was 75282 m^3, which we estimate to correspond to 64% of the total production, according to the 2002 data reported in the National Survey to Industrial Production (INE, 2004b). The annual average production of these industries is approximately 79000 m^3 and the exports represent 9% of the production. In 2003, there is a higher volume of exports, i.e., 9.5% of the production in these industries. The most common products produced by the treating plants are vineyard stakes (26% of the production) and fence posts (20%). Other products

[1] 35.0% m/m $CuSO_4.5H_2O$; 45.0% m/m $K_2Cr_2O_7$ or $Na_2Cr_2O_7.2H_2O$ and 20.0% m/m $As_2O_5.2H_2O$

[2] 32.6% m/m $CuSO_4.5H_2O$; 41.0% m/m $K_2Cr_2O_7$ or $Na_2Cr_2O_7.2H_2O$ and 26.4% m/m $As_2O_5.2H_2O$

[3] 47.5% m/m CrO_3; 18.5% m/m CuO and 34% m/m $As_2O_5.2H_2O$

include sawn timber, fruit tree supports, highway posts and telephone poles as represented in the Figure 4.2.

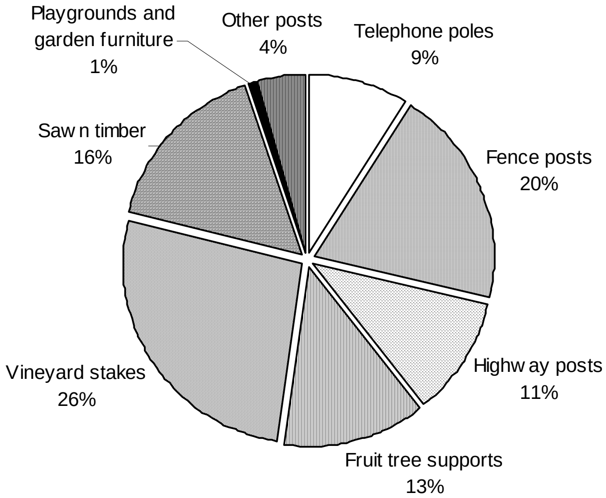

Figure 4.2 – Distribution of the treated wood products made in Portugal in the respondent industries.

According to the answers obtained, the production of treated wood waste in the respondent industries corresponds to an average value of 4% of the production. These wastes are incinerated in heat boilers, recycled in wood composites or reused internally.

4.1.2 Estimation of Treated Wood Waste Production

The inventory of CCA-treated wood in Portugal was based in the results of the survey made to the wood preservation plants and in the treated wood production, exports and imports data available in the National Statistics Institute (INE).

This quantification is based in a mass balance approach (see Figure 4.3), modified from Solo-Gabriele *et al.* (1998), considering estimates of the quantities of CCA-treated wood generated over time (between 1984 and 2002) and assumptions concerning the service-life for treated wood. The estimation of CCA-treated wood waste production was done for a time span of twenty years, until 2022.

The approach used can be represented by the following equation:

$$\frac{dU}{dt} = P - E + I - D \tag{4.1}$$

where **U** is the mass of CCA-treated wood in use at any time t; **dU/dt** is the rate of change of U with respect to time (mass/time); **P** is the mass of CCA-treated wood produced per unit time; **E** is the mass of CCA-treated wood exported per unit time; **I** is the mass of CCA-treated wood imported per unit time and **D** is the mass of CCA-treated wood disposed per unit time. To convert the volumes in mass the density of 640 kg m^{-3} (water content of 12%) was used, characteristic of the *Pinus pinaster* Portuguese wood (Carvalho, 1997).

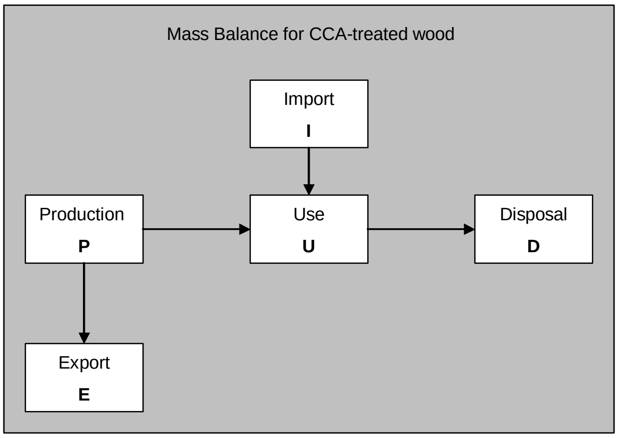

Figure 4.3 – Mass balance approach for the inventory of CCA-treated wood [modified from Solo-Gabriele *et al*. (1998)].

The production, export and import data between 1992 and 2002 are presented in Figure 4.4. The 1984 production is the one obtained by Reimão & Cockcroft (1985) (see Table 2.1) and the data between 1985 and 1991 was estimated.

It was assumed that 90% of treated wood was preserved with CCA, according to the results of the questionnaire to the preserving industries and the information available about the commercialisation of preservation products (Sanches, 2004).

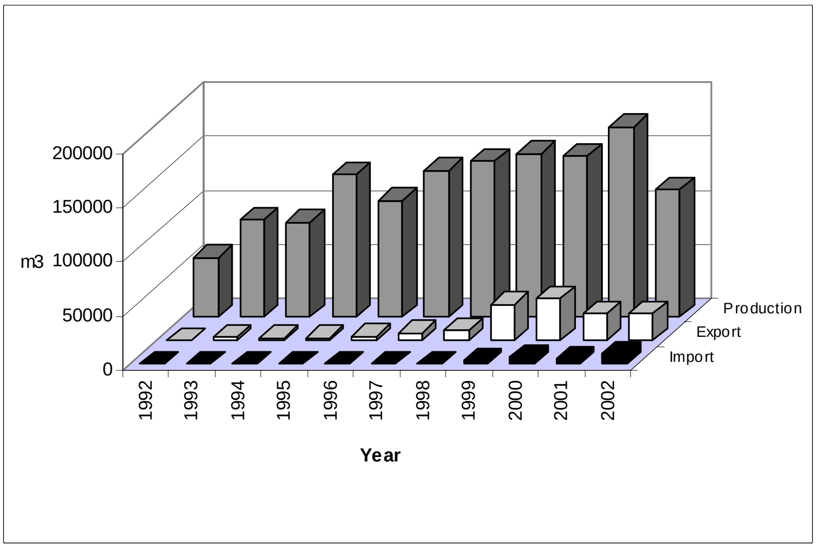

Figure 4.4 – Production, export and import of treated wood products in Portugal (m^3), from 1992 to 2002 (data from the National Statistics Institute).

The service life of treated wood products is different, according to their use. Lumbers, timbers and fence posts used in residential applications, for example, generally have short service lives (Solo-Gabriele *et al.*, 1998). Cooper (2003) refers that, in U.S.A., 67% of decks replaced were less than 10 years old and the average service life of decks was only 9 years. Other products, like utility poles and railway sleepers, generally have longer service lives (40 to 60 years).

In our case, the following average service lives were assumed: a) for poles 40 years; b), for sawn wood, fence posts and other products 20 years and c) for playgrounds and garden furniture 10 years. Table 4.1 presents the average service life of treated wood products considered for the mass balance approach.

The quantities of treated wood discriminated by products and, consequently, with different service lives, were obtained with the results to the questionnaire sent to the preservation industries (see Figure 4.2).

The appearance of the preserved wood and changes in style and tastes of the owners seems to be determinant of its service life (McQueen *et al.*, 1998). So it is clear that residential/consumer treated wood may not be kept in service for its full potential service life.

Table 4.1 – Service life of treated wood products.

Product	Percentage of what (%)	Duration of service life (years)	Reference
Poles	25	35 to 37	Solo-Gabriele *et al.* (1998); Humar & Pohleven (2001)
	50	38 to 42	
	25	43 to 45	
Sawn wood, fence posts	25	20 to 22	McQueen *et al.* (1998)
	50	23 to 27	
	25	28 to 30	
Playgrounds and garden furniture	25	5 to 7	Solo-Gabriele *et al.* (1998); Humar & Pohleven (2001); Cooper (2003)
	50	8 to 12	
	25	13 to 15	

The quantification of treated wood waste presented in Figure 4.5 does not take into account these changes in style and tastes, nor degradation and uncontrolled combustion of treated wood waste. In Portugal, a significant amount of the telephone poles in forest and rural areas are accidentally burned during forest fires, especially on Summer.

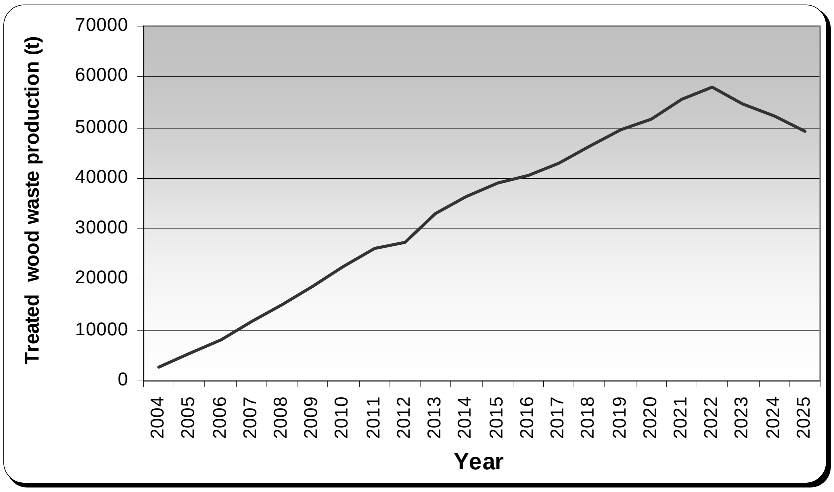

Figure 4.5 – Estimated production of CCA-treated wood waste based on expected service life and in the different uses of treated wood products.

The results presented in Figure 4.5 show a peak in the annual production of CCA-treated wood waste in 2022, where it will reach approximately 58000 Mg.

4.2 Location of CCA-treated Wood Waste

Another important step for the development of the optimisation model was the determination of CCA-treated wood waste location, i.e. the location of the demand nodes. This model takes into account the uses and the production of CCA-treated wood in Portugal. Simultaneously with the results of the questionnaire to the preservation industries and considering the soil occupation, a Geographic Information System (GIS) model was used to identify the places in the national territory where treated wood is in use and potentially will become a waste.

GIS enables us to see a model representation of the world (a simplified view of this complex world) as a collection of thematic layers, allowing a new insight of the problem or situation in analysis. GIS support the capture, management, manipulation, analysis, modelling and display of spatially referenced data for solving complex planning and management problems and also include the additional ability to perform spatial operations. The efficient analysis and visualisation of information during decision-making offers managers and professionals greater productivity and timesaving and, undoubtedly, more informed decisions.

Cartographic modelling, Map Algebra or Mapematics are generic forms to express and organize the methods to select and use the variables and spatial operations to develop a GIS model.

A logic extension of the use of concepts and algebraic methods as spatial analysis elements is the utilization of natural language (Burrough, 1986). So the procedures to the development of a cartographic model can be sum-up as: i) identifying of the necessary geographic information in the form of layers or themes; ii) making use of logic and natural language to express the processes between the available data and the solution obtained; iii) drawing a flowchart with the steps representing the two processes described; iv) adding to the flowchart the necessary commands to the implementation of the model.

Table 4.2 presents the first of these steps that is the metadata of the geographic information used in the treated wood waste location model, discriminating the data model and the source of the information.

Figure 4.6 represents the flowchart mentioned in steps iii) and iv), where operations are expressed in natural language and the commands used in the ArcGIS 8.2 © ESRI software to implement the GIS model are given. The results of the GIS model are presented in Figure 4.7.

Table 4.2 – Metadata of the geographic information used in the GIS model.

Data	Source	Data model
Population	Data from the 2001 Portuguese population census made by the National Statistics Institute ("*Instituto Nacional de Estatística*")	Tabular
Wood Preservation Plants	Data from Belém Database available in the National Statistics Institute ("*Instituto Nacional de Estatística*")	Tabular
Land use	Corine Land Cover layers provided by the Portuguese Geographic Institute ("*Instituto Geográfico Português*")	Vectorial
Highways	National Road Plan 2000 of the Portuguese Road Institute ("*Instituto das Estradas de Portugal*")	Vectorial
Administrative divisions	Portugal Administrative Map provided by the Portuguese Geographic Institute ("*Instituto Geográfico Português*")	Vectorial

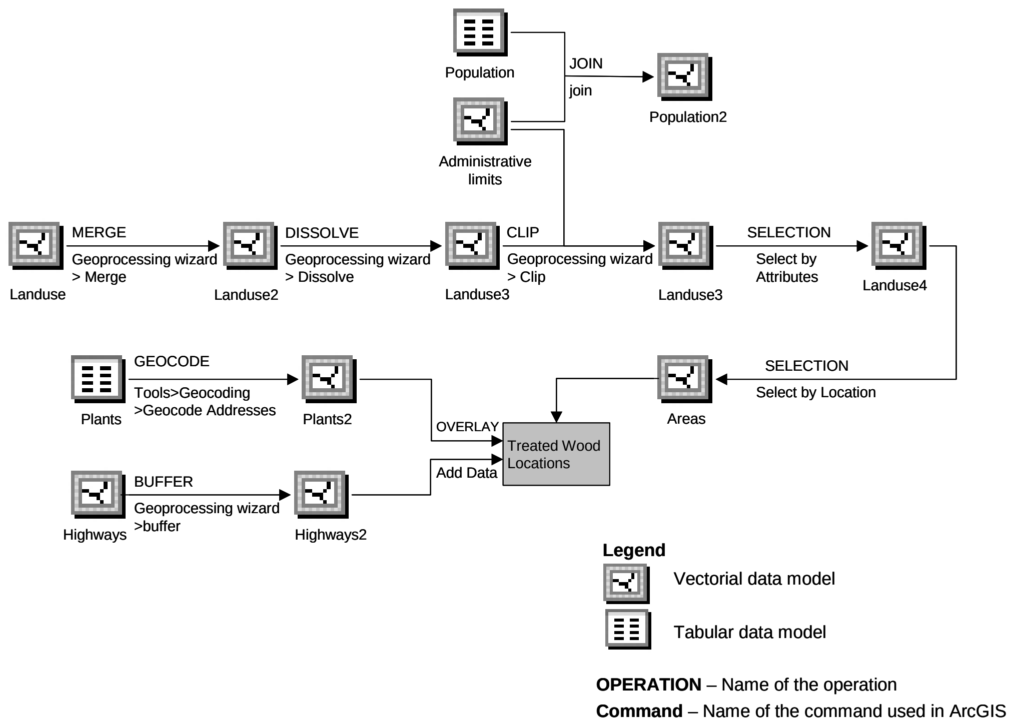

Figure 4.6 – Flowchart with the operations and commands used to implement the GIS location model with the software ArcGIS 8.2.

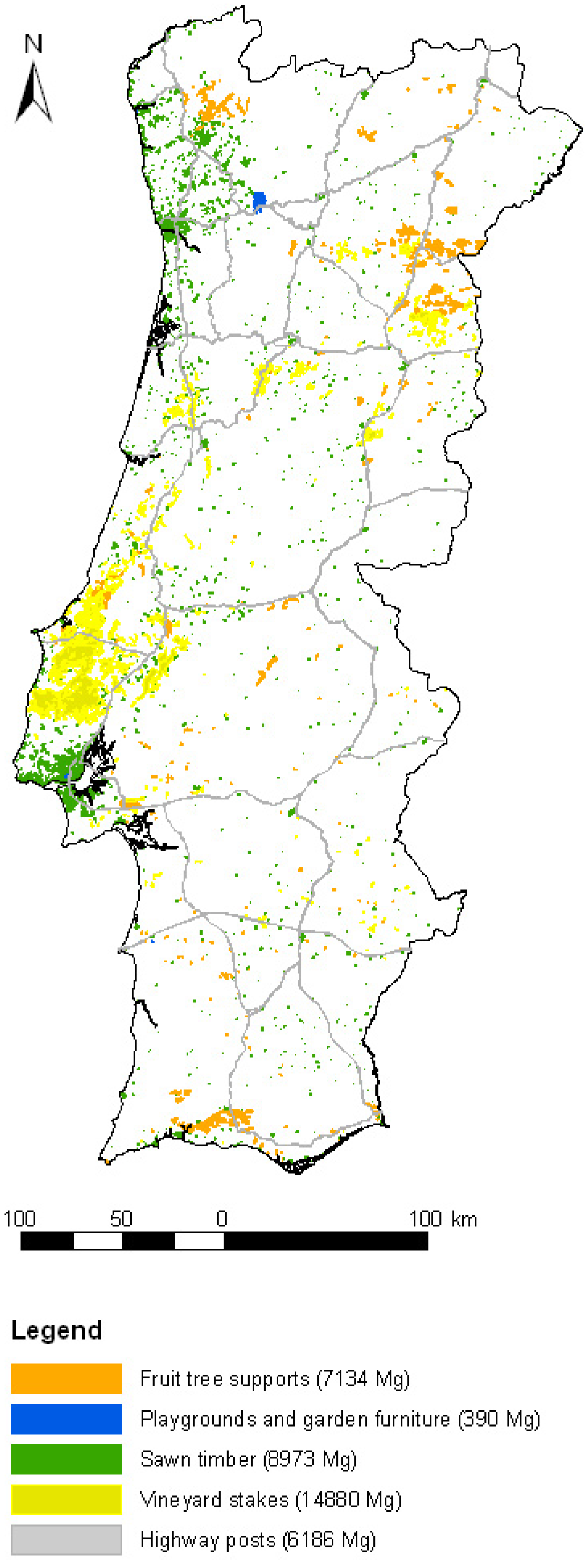

Figure 4.7 – Treated wood locations and quantities in 2022.

Some national particularities where considered, such as the fact that in specified regions for the production of quality wine in the north of Portugal ("Região Demarcada do Douro" and "Região Demarcada do Vinho Verde") are not used treated wood vineyard stakes, since concrete, schist or granite stakes are preferred (Reimão, 2003).

After obtaining the small regions where CCA-treated wood waste potentially exist, we determined the centroids of those polygons. These points will be our demand nodes in the location problem. After that, we assign an estimate of the production of CCA-treated wood waste to these centroids. That estimate is proportional to the areas and to the specific CCA-treated wood product use located in that region. For example, considering equal areas with vineyard stakes or playgrounds and garden furniture, the corresponding centroid of the vineyard region will have a bigger potential quantity of wastes, since vineyard stakes are more important or have a greater importance in the production of CCA-treated wood.

The highway poles were treated separately, since the geographic information available is the location of the highways, i.e. a line and not a polygon like in the other CCA-treated wood products. So, we have created a buffer area with a width of 10 m and split those polygons in smaller ones with a length of 10 km. Then, we determined the centroid of that area and assigned potential quantities of treated wood waste to all centroids exactly like we have done before. In this case, all the quantities are equal, given that the areas are similar.

Since it was not possible to obtain the location of treated wood telephone poles and, according to the Portuguese Telecom ("*PT Comunicações*"), all the removed poles are sold for reuse in construction, these treated wood wastes were not considered as demand nodes in the location model.

We also considered as potential sources of CCA-treated wood waste the preservation industries, since, according to the results of the questionnaire, about 4% (in average) of the production is rejected and becomes waste. The potential amount of CCA-treated wood waste was also assigned to the points that represent the preservation industries location.

Initially, we expected that the preservation industries would have an influence area around them where the treated wood consumption would be higher and, consequently, the production of waste was also more significant. However, that is not the case in Portugal, since the preservation industries location is more related with the proximity of raw material than with the proximity of consumers (Reimão, 2003).

We also implemented another approach to this problem, using the population data, instead of the potential location of the wastes. Consumer lumber makes up the majority of current and future expected production of CCA treated wood – about 56% of 1995 production compared to about 82% of expected waste wood generated in 2010 (Cooper, 2003). So we assume that the wood that needs to be sent to the remediation plants is proportional to the number of inhabitants in each area. Thus, in some experiments, we use population instead of amount of waste wood needing remediation.

The GIS was also used to find the population of each “freguesia” of Portugal (NUTS 5) (see Figure 4.6). The data is from the latest census (from the year 2001), provided by the Portuguese Statistics Institute (INE) and was linked to the administrative regions, in order to obtain the geographical location of the centroid of the “freguesia” and its population.

4.3 Formulation of the Location Model for the Remediation Units

The problem we are trying to solve consists in selecting an optimal configuration of CCA-treated wood remediation units, to minimise transportation costs, considering the location of demand nodes where these wastes are produced.

The model formulation starts with the identification of the key variables. In this case these variables are the location of the demand nodes and the waste quantities that have to be transported, as well as the location of remediation units. Besides that, the location of the remediation units and the allocation of demand nodes are considered simultaneously.

The problem is subject to the following constraints:

- Demand is concentrated in a set of discrete locations – demand nodes. These demand points have fixed total demands;
- Every demand node shall be allocated to the closest remediation unit;
- There is no interaction between remediation units;
- There are no spatial constraints (forbidden areas, barriers) to the location of remediation units. It can be located anywhere on the planar coordinate system.

The following assumptions were made for the location model of treated wood waste remediation units:

- Although the facilities in study are hazardous waste treatment units, they were not considered as obnoxious facilities, like others in literature [see for example Giannikos (1998); Nema & Gupta (1999); Lahdelma *et al.* (2002); Llurdés *et al.* (2003)]. Therefore we did not consider push objectives (see section 3.3.11) and the model formulation assumed that the facilities were "desirable".
- We assume that all remediation units are equal and employ the electrodialytic process that has proven to be efficient in the removal of Cr, Cu and As in a pilot study [see more details in Villumsen (2003)]. After the remediation, the wood can be recycled.
- Capital or fixed costs are not considered. We assume that these costs are equal to all remediation units.
- The transportation costs between the demand nodes and the remediation units are proportional to the Euclidean distance.

- We did not consider the capacity of the remediation units; it is an uncapacitated facility location problem.

The final step is the formulation of mathematical equations for the objectives and constraints defined. Considering the optimisation criterion and a set of demand nodes with known demands, the location-allocation problem or the multisource Weber problems consist in selecting the locations of a number of facilities (remediation units) and to decide the corresponding allocation of the demand nodes to the remediation units (see also section 3.3.3).

So the problem can be formulated as:

$$\min_{x_i, y_i, z_{ij}} \sum_{i=1}^{p} \sum_{j=1}^{n} z_{ij} \cdot w_j \cdot d_j(x_i, y_i) \qquad (4.2)$$

subject to

$\sum_{i=1}^{p} z_{ij} = 1 , \quad j = 1, 2, ..., n$

$z_{ij} \in [0,1]$

$i = 1, 2, \ldots, p$

$j = 1, 2, \ldots n$

where *p* facilities must be located to satisfy the demand of *n* demand nodes, x_i y_i denote the coordinates of the i^{th} facility, d_j the Euclidean distance from (x_i, y_i) to the j^{th} user, w_j the demand (or weight) of the j^{th} user and z_{ij} the fraction of this demand which is satisfied from the i^{th} facility. There is always an optimal solution with all $z_{ij} \in \{0, 1\}$, i.e., each demand node is satisfied from a single facility.

The objective function is neither convex nor concave and may have a large number of local minima (Drezner *et al.*, 2002). This problem can also be interpreted as an enumeration of the Voronoi partitions of the demand set and it was proven to be NP-hard in Meggido & Supowit (1984).

4.4 Classification of the Location Model

According to the classification scheme proposed by Hamacher and Nickel (1998) and presented in section 3.2.7, the location model can be classified as: ***N / P / alloc / l_2 / Σ***.

N in position one means that *n* points have to be located. P in position 2 represents the Euclidean plane, while *alloc* in position 3 means that we are dealing with an allocation problem where demand nodes are allocated to the facilities. In position 4, we have l_2 that specifies the distance function, in this case the Euclidean distance. Finally, Σ means that the objective function considered is the classical Weber objective function or minisum objective function.

4.5 Dataset and Methods Used

Location problems are usually formalized by considering a number of demand nodes, each with a given location and weight (amount of resources demanded), a number of facility locations that must be found, and a cost function that takes into account the relation between the demand nodes and the facility locations (for more details on location problems see 3.3).

In this case, as explained in 4.2, the demand nodes will be producers of waste wood, the facilities will be the remediation units, and the cost function will be the Euclidean distance between the demand nodes and the facilities, considering the amount of wood generated at each demand node. As producers of CCA-treated waste wood, we have considered the centroids of the regions where the soil occupation is related with treated wood products, i.e. vineyards, highways, gardens, orchards, among others.

We have also considered a simple approach where the demand nodes consist in the centroids of each “freguesia” of Portugal (NUTS 5) and the demand is represented by the number of inhabitants.

Location-allocation problems can be treated as clustering problems. The problem solving process can be viewed as properly clustering N demand nodes into P groups served by the associated P remediation units, to achieve the minisum objective.

The k-means algorithm (MacQueen, 1967) is one of the best known classic statistical clustering methods. It generates a specific number of disjoint, flat (non-hierarchical) clusters. The k-means method is numerical, unsupervised, non-deterministic and iterative, and is basically a stochastic hill-climbing optimisation technique.

The k-means algorithm starts with an initial partition of the cases into k clusters. Subsequent steps modify the partition to reduce the sum of the distances for each case from the mean of the cluster to which the case belongs. The modification consists of allocating each case to the nearest k means of the previous partition. This leads to a new partition for which the sum of distances is strictly smaller than before. The improvement step is repeated until the improvement is very small. This method is very fast.

There is a possibility that the improvement step leads to fewer than k partitions. In this situation one of the partitions (generally the one with the largest sum of distances from the mean) is divided into two or more parts to reach the required number of k partitions. The algorithm can be re-run with different randomly generated starting partitions to reduce the chances of the heuristic producing a poor solution.

On the other hand, the works of Lozano *et al.* (1998) and Hsieh & Tien (2004) show that Self-Organizing Maps (SOM) or Kohonen maps may be a potential heuristic method meeting the requirements of both quality of solution and speed of computation.

The SOM provides a visual representation of a vector quantification algorithm that places a number of vectors into a high-dimensional input data space in such a way that they approximate the original data patterns in an ordered fashion (Kohonen, 2001). It is an unsupervised neural network that tries to preserve topological relations between points in the input and output spaces. Generally, SOMs have many neurons and these are spread in the input space proportionally to the density of data points.

The SOM is related to clustering algorithms, in that it generates grouping of data points taken to be described by a single vector of typical values. The SOM is distinct from standard clustering methods since it does not operate with separate clusters: rather, it allocates data points to groups which are related (Curry & Morgan, 2004).

By this property, the locations of the N demand nodes, X_i for i=1, …, N are used input patterns to form a training set and P output units will be the remediation units locations, W_j for j=1,…, P. In addition, the demands of N nodes can be represented by the repeating occurrences in a training set. During the training process of a SOM, the location of remediation units, W_j, is updated to reinforce the proximity between the distribution of the input locations/demand of the N demand nodes and the facilities (remediation units) location.

There are two layers in the SOM: the input layer and the competitive (output) or Kohonen layer, which are fully interconnected. The competitive layer includes the output units which are typically organized in a planar (2-D) lattice.

The algorithm computes a similarity (distance) measure between the input vector and the weight vector of each unit. The unit with the closest (shortest Euclidean distance) weight vector is chosen as the winner, which is then allowed to move even closer to the input vector (Nour & Madey, 1996).

After training, the network converges so that each weight vector may coincide with the centroid of a particular cluster of vectors from the input data. Thus, the weight vectors become detectors of different classes or clusters.

K-means algorithm is essentially the same as Kohonen's learning law except that the learning rate is the reciprocal of the number of cases that have been assigned to the winning cluster; this decrease of the learning rate makes each codebook vector the mean of its cluster and guarantees convergence of the algorithm to an optimum value of the error function (the sum of squared

Euclidean distances between cases and codebook vectors) as the number of training cases goes to infinity (Sarle, 2004). Kohonen's learning law with a fixed learning rate does not converge. As it is well known from stochastic approximation theory, convergence requires the sum of the infinite sequence of learning rates to be infinite, while the sum of squared learning rates must be finite (Kohonen, 2001). These requirements are satisfied by MacQueen's k-means algorithm.

A comparison of the two methods showed that SOM is less prone to local optima than k-means and also that the search space is better explored by SOM (Bação *et al.*, 2004). This is due to the effect of the neighbourhood parameter which forces units to move according to each other in the early stages of the process. This characteristic can be seen as an “annealing schedule” which provides an early exploration of the search space. On the other hand, k-means gradient orientation forces a premature convergence which, depending on the initialization, may frequently yield local optimum solutions.

4.6 Experimental Conditions

In order to determine if the increasing amounts of CCA-treated wood waste estimated (see 4.1.2) have an effect on the location of the remediation units, we made experiments for three different years: **2010**, **2015** and **2020** for the demand nodes that represent waste wood quantities.

Regarding the population dataset, only the data referring to the 2001 census were used. We could have used both datasets together, CCA-treated wood waste and population, but we consider that the population data would be redundant since urban areas are also represented in the CCA-treated wood waste dataset.

We also considered three different possibilities for the number of remediation units. Therefore we try to solve the location problem considering **5**, **10** and **15** remediation units.

The implementation of the clustering techniques that we used did not allow weighting of different data patterns. To achieve this weighting, we generated data patterns for each data set (see Table 4.3). These data patterns were then put into random order so as not to introduce unnecessary bias.

Table 4.3 – Data patterns generated for each data set.

Data Set	Initial number of points	Number of points in the generated data patterns
Population	4034	99146
CCA-treated wood waste in 2010	3381	155044
CCA-treated wood waste in 2015	3381	267876
CCA-treated wood waste in 2020	3381	355110

Two different tests (**Test 1** and **Test 2**) were made, with different experimental conditions, that will now be described.

When using SOM, we have to provide an output space grid upon which the neurons are set, i.e. the neurons are organized into a dimensional and fully laterally connected topology. In Test 1 we used three different grids, a **5 x 1**, a **5 x 2** and a **5 x 3** grid. In Test 2 we used **5 x 1**, **10 x 1** and **15 x 1** grids. In all grids we used a **rectangular topology**.

In all experiments, we used **random initialization** and **two learning phases**. The neurons adapt (learn) in such a way that the topological relationships in the input pattern space are preserved.

In the first phase we used an **initial radius of 3** (linearly decreasing to 0), a **learning rate** α of **0.3** (also decreasing linearly to 0), a **sequential training algorithm** and **bubble neighborhood**. The second learning phase differed from the first in the initial radius (we used an initial radius of 2) and in the learning rate (0.05). The value of the learning rate α was reduced linearly to zero with the number of times the data was applied to the network. This was to ensure some convergence of the process.

We tested different numbers of training epochs. In Test 1, in the first learning phase, we used **3**; **30** and **300 epochs**, while in the second we used **6**; **60** and

600 training epochs. In Test 2, we used always **5 epochs** in the first learning phase and **10 epochs** in the second learning phase, in face of the results of the first test, as explained in the next chapter (5. Results and Discussion). Table 4.4 summarizes the experimental conditions in Test 1.

Tables 4.5 and 4.6 summarize experimental conditions in Test 2, for the two clustering methods tested, SOM and k-means, respectively. For the k-means method we only tested the CCA-treated wood waste demand nodes.

Table 4.4 – Experimental conditions in Test 1.

Run	Demand Nodes	Grid	Training epochs	
			1st learning phase	2nd learning phase
P1	Generated data patterns considering the centroids of the "freguesia" and its population	5 x 1	3	6
P2			30	60
P3			300	600
P4		5 x 2	3	6
P5			30	60
P6			300	600
P7		5 x 3	3	6
P8			30	60
P9			300	600
CA1	Generated data patterns considering the centroids of the potential areas with CCA-treated wood wastes and its production in 2010	5 x 1	3	6
CA2			30	60
CA3			300	600
CA4		5 x 2	3	6
CA5			30	60
CA6			300	600
CA7		5 x 3	3	6
CA8			30	60
CA9			300	600
CB1	Generated data patterns considering the centroids of the potential areas with CCA-treated wood wastes and its production in 2015	5 x 1	3	6
CB2			30	60
CB3			300	600
CB4		5 x 2	3	6
CB5			30	60
CB6			300	600
CB7		5 x 3	3	6
CB8			30	60
CB9			300	600
CC1	Generated data patterns considering the centroids of the potential areas with CCA-treated wood wastes and its production in 2020	5 x 1	3	6
CC2			30	60
CC3			300	600
CC4		5 x 2	3	6
CC5			30	60
CC6			300	600
CC7		5 x 3	3	6
CC8			30	60
CC9			300	600

In all experiments we used the SOM Toolbox for MATLAB implementation developed at Helsinki University of Technology (Vesanto *et al.*, 2000). All runs of the location model were executed in a 2.00 GHz CPU Pentium 4 PC running Windows XP, using MATLAB R12. The MATLAB code is presented in Appendix II.

Table 4.5 – Experimental conditions in Test 2 using SOM.

Experiment	Demand Nodes	Grid	Training epochs	
			1st learning phase	2nd learning phase
p1	Generated data patterns considering the centroids of the "freguesia" and its population	5 x 1	5	10
p2		10 x 1		
p3		15 x 1		
a1	Generated data patterns considering the centroids of the potential areas with CCA-treated wood wastes and its production in 2010	5 x 1	5	10
a2		10 x 1		
a3		15 x 1		
b1	Generated data patterns considering the centroids of the potential areas with CCA-treated wood wastes and its production in 2015	5 x 1	5	10
b2		10 x 1		
b3		15 x 1		
c1	Generated data patterns considering the centroids of the potential areas with CCA-treated wood wastes and its production in 2020	5 x 1	5	10
c2		10 x 1		
c3		15 x 1		

Table 4.6 – Experimental conditions in Test 2 using k-means.

Experiment	Demand Nodes	Number of remediation units	Algorithm
k1	Generated data patterns considering the centroids of the potential areas with CCA-treated wood wastes and its production in 2010	5	Batch
k4		10	
k7		15	
k2	Generated data patterns considering the centroids of the potential areas with CCA-treated wood wastes and its production in 2015	5	
k5		10	
k8		15	
k3	Generated data patterns considering the centroids of the potential areas with CCA-treated wood wastes and its production in 2020	5	
k6		10	
k9		15	

5. RESULTS AND DISCUSSION

5.1 Results from Test 1

The objective of our location problem is to minimize the average weighted distance of the demand nodes to the remediation unit. The average distance of a demand node to a remediation unit in each of the experiments in Test 1 is given in Table 5.1. This distance corresponds to the mean quantization error that is the difference between each data vector and its best matching unit (BMU) or, by other words, the neurons as idealized prototypes of the data and the data itself.

All the maps with the CCA-treated wood waste remediation unit locations are presented in Appendix III.

The results presented in Table 5.1 show that the increase in the number of epochs does not affect significantly the average distance to a remediation unit. In fact, increases of 100 times the number of training epochs do not influence greatly the mean quantization error or average distance to a remediation unit. On the other hand and simply due to statistical variation, there are cases with 300 training epochs in the first learning phase and 600 training epochs in the second that resulted in slightly higher average distances than the ones obtained in the experiments where we used 30 and 60 training epochs in the first and the second learning phase, respectively (that’s the case of P3 and CA6, for example). For these reasons, in Test 2 we used only 5 training epochs in the first learning phase and 10 training epochs in the second, what allowed us to make more runs, with different experimental conditions, since the computing time is significantly lower.

The location of the remediation units obtained is very similar in the three years considered (2010, 2015 and 2020) as showed in Figures 5.1, 5.2 and 5.3. We can also observe that the results obtained with the population data as demand nodes were similar to the ones achieved using the location and quantity of CCA-treated wood waste, particularly in the cases where we consider 10 and 15 remediation units (Figures 5.2 and 5.3).

Geographically, the solutions obtained in this test provide a good coverage of the country and of the demand nodes and, at a simple glance; the solutions seem viable.

Table 5.1 – Average distance (in km) to a remediation plant in Test 1.

Experiment	Average Distance (km)	Number of training epochs	Grid
P1	36.08	3; 6	
P2	35.98	30; 60	5 x 1
P3	36.04	300; 600	
P4	24.41	3; 6	
P5	24.05	30; 60	5 x 2
P6	24.07	300; 600	
P7	19.92	3; 6	
P8	20.06	30; 60	5 x 3
P9	20.06	300; 600	
CA1	43.89	3; 6	
CA2	43.91	30; 60	5 x 1
CA3	43.90	300; 600	
CA4	25.03	3; 6	
CA5	24.98	30; 60	5 x 2
CA6	25.11	300; 600	
CA7	18.84	3; 6	
CA8	18.47	30; 60	5 x 3
CA9	18.50	300; 600	
CB1	43.84	3; 6	
CB2	43.89	30; 60	5 x 1
CB3	43.84	300; 600	
CB4	24.90	3; 6	
CB5	24.55	30; 60	5 x 2
CB6	24.95	300; 600	
CB7	18.77	3; 6	
CB8	18.42	30; 60	5 x 3
CB9	18.41	300; 600	
CC1	44.08	3; 6	
CC2	44.10	30; 60	5 x 1
CC3	44.09	300; 600	
CC4	24.96	3; 6	
CC5	24.95	30; 60	5 x 2
CC6	24.90	300; 600	
CC7	18.98	3; 6	
CC8	18.42	30; 60	5 x 3
CC9	19.96	300; 600	

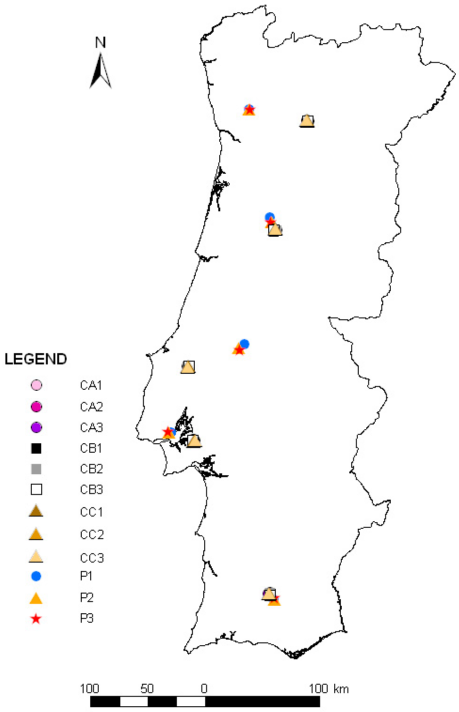

Figure 5.1 – Results for Test 1 for 5 remediation units.

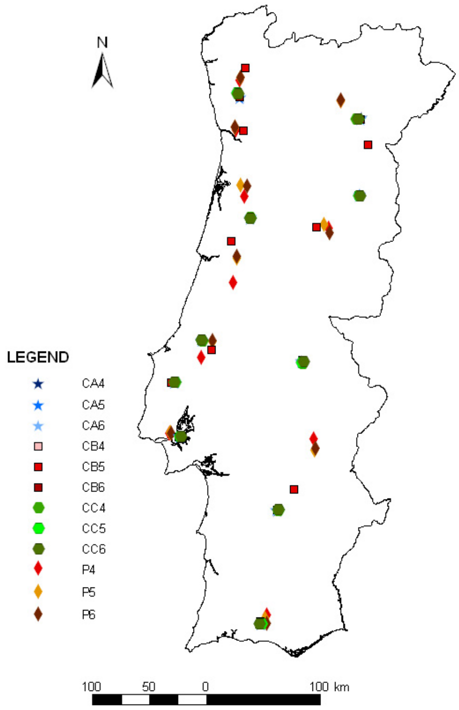

Figure 5.2 – Results for Test 1 for 10 remediation units.

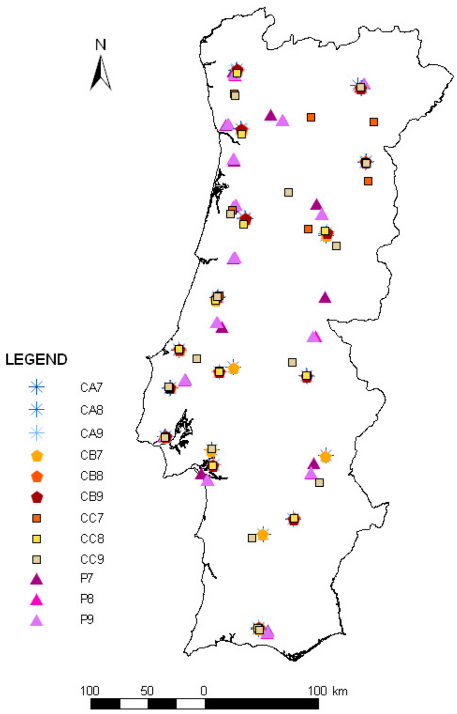

Figure 5.3 – Results for Test 1 for 15 remediation units.

5.2 Results from Test 2

The average distance to a remediation plant in each of the experiments in Test 2 is given in Table 5.2. The results are presented side by side according to the experimental conditions and to ease the comparison between the two clustering methods tested (SOM and k-means). The minimum average distance for the same experimental conditions (demand nodes and number of remediation units) is marked in bold. We can observe that k-means obtains average distances slightly lower than SOM in 5 experiments (a1, a3, b1, b2 and c1). The best solution is always obtained with k-means (also marked in bold).

Table 5.2 – Average distance (in km) to a remediation plant in Test 2 (over 30 runs) and best solution found in each experiment.

Year	SOM				K-means			
	Experiment	**Average distance (km)**	**Best solution (km)**	**σ**	**Experiment**	**Average distance (km)**	**Best solution (km)**	**σ**
2010	a1	43.907	43.784	0.021	k1	**41.490**	**38.521**	2.788
	a2	**25.075**	24.890	0.091	k4	26.348	**24.476**	1.333
	a3	19.300	18.725	0.164	k7	**19.107**	**18.632**	0.381
2015	b1	43.829	43.769	0.040	k2	**43.637**	**38.852**	3.535
	b2	24.943	24.908	0.022	k5	**24.837**	**24.257**	0.101
	b3	**19.184**	18.533	0.220	k8	19.647	**18.368**	0.879
2020	c1	44.095	43.990	0.041	k3	**40.332**	**38.783**	1.953
	c2	**24.924**	24.885	0.050	k6	26.042	**24.454**	1.025
	c3	**18.983**	18.737	0.347	k9	20.462	**18.350**	1.201

Appendix III presents all the maps produced with the CCA-treated wood waste remediation units location for the experimental conditions tested.

The results show that the remediation units locations are alike in the three years considered (2010, 2015 and 2020), as we have also observed in Test 1. Another important remark is the consistence and the reproducibility of the results of the SOM experiments, observed in all experiments in Test 2. K-means, however, is more instable. As it can be observed in Figures 5.4 to 5.9 while SOM results (remediation unit locations) are almost coincident in all runs of the same experiment, k-means results are more disperse. The variance in SOM solutions is significantly lower, as we can see comparing the standard deviation of SOM and k-means average distance in Table 5.2.

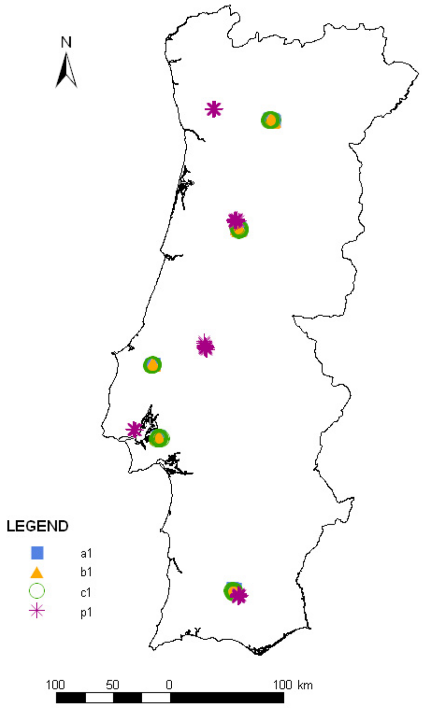

Figure 5.4 – Results for Test 2 for 5 remediation units using SOM.

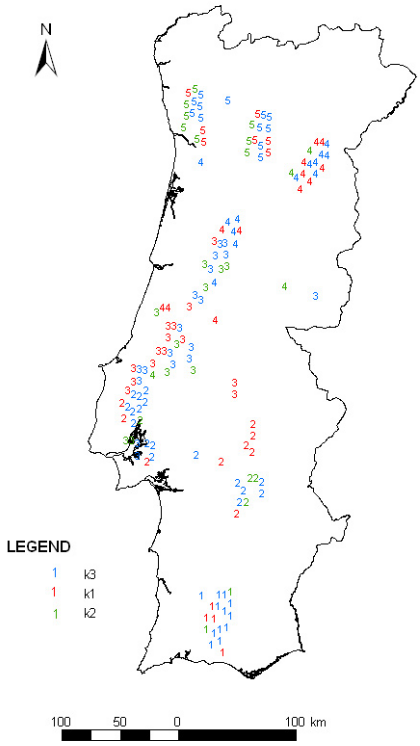

Figure 5.5 – Results for Test 2 for 5 remediation units using k-means.

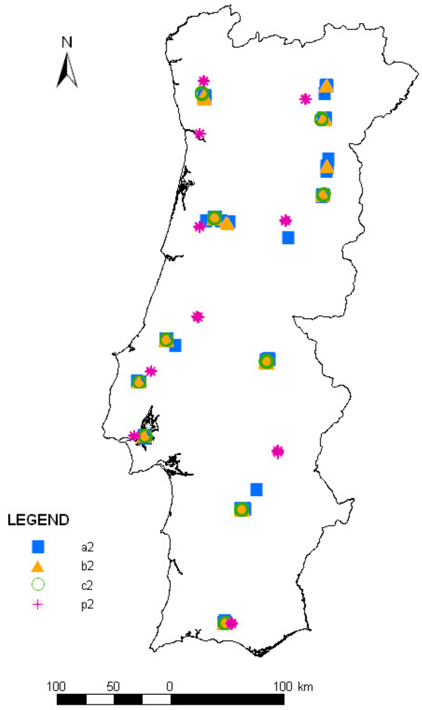

Figure 5.6 – Results for Test 2 for 10 remediation units using SOM.

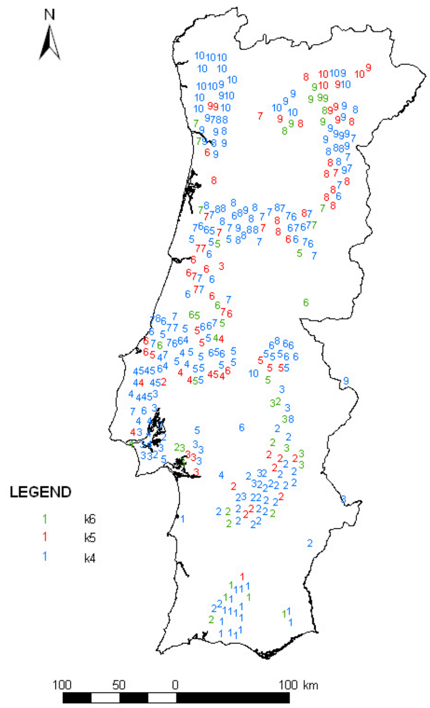

Figure 5.7 – Results for Test 2 for 10 remediation units using k-means.

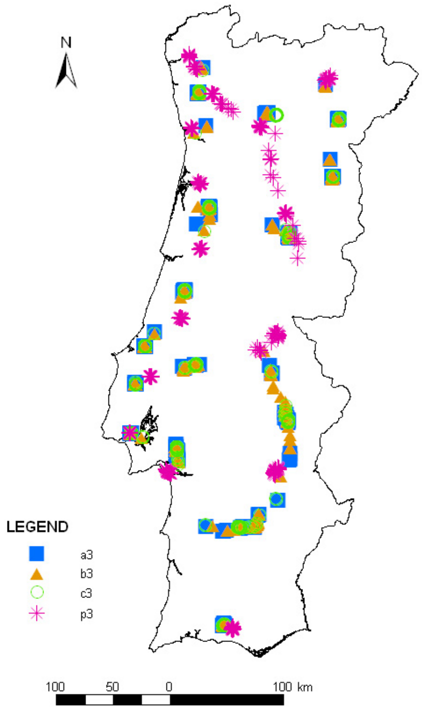

Figure 5.8 – Results for Test 2 for 15 remediation units using SOM.

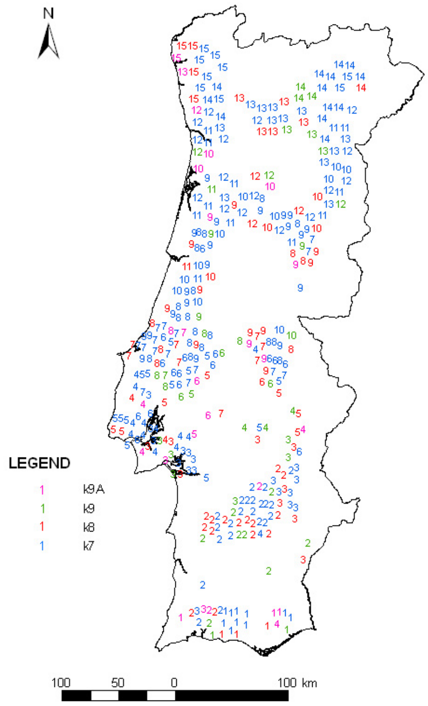

Figure 5.9 – Results for Test 2 for 15 remediation units using k-means.

As in Test 1, we can observe the proximity between the solutions obtained with CCA-treated wood waste demand nodes and with population demand nodes.

The computing time of the tested clustering methods is similar in each experiment, increasing proportionally to the number of demand nodes, as can be seen in the Figures 5.10, 5.11 and 5.12. We can also see that the k-means (k1 – k9) method is significantly faster than SOM (experiments p, a, b and c).

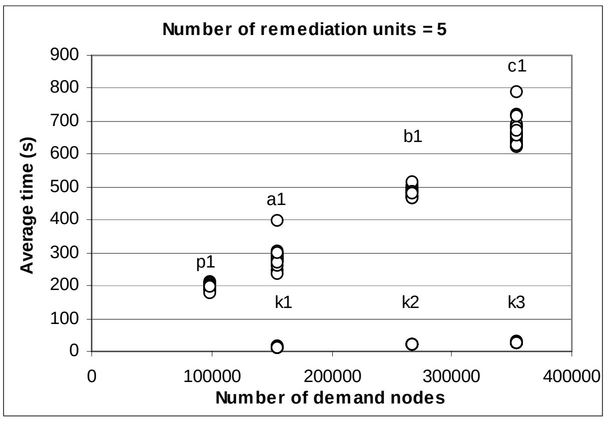

Figure 5.10 – Average run time for the 5 remediation units experiments.

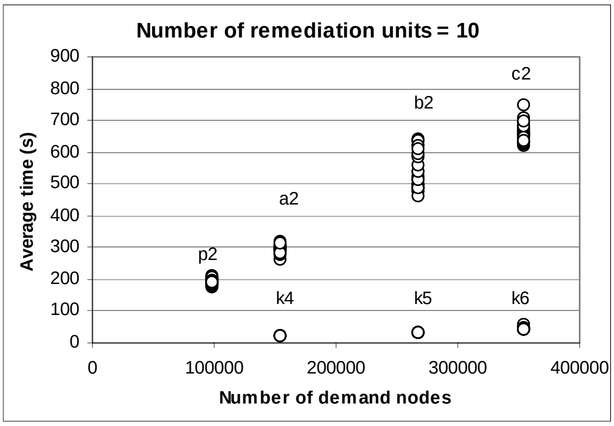

Figure 5.11 – Average run time for the 10 remediation units experiments.

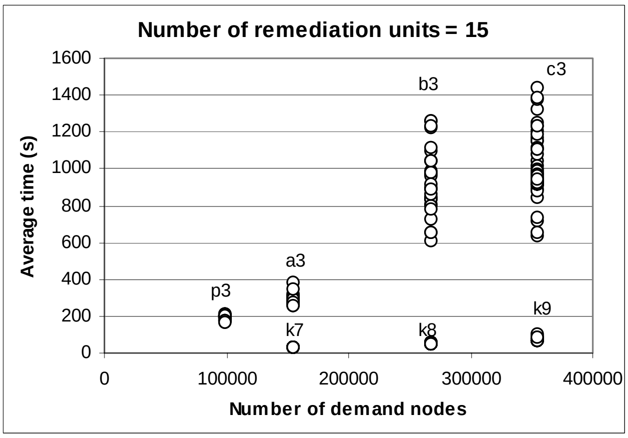

Figure 5.12 – Average run time for the 15 remediation units experiments.

In order to determine if k-means results would be more robust if we use the same computing time of SOM experiments instead of the same number of epochs, we have made another experiment (k9A), using the same conditions of k9, but increasing the number of epochs (we used 250 epochs instead of 15). The results obtained are very similar to the k9 results, with a small increase in the average distance to a remediation unit (we obtained an average distance of 20,799 km instead of 20,466 km) and the same geographic variability of the solutions (see Figure 5.9). As we had already seen with SOM results in Test 1, the increase in the number of epochs does not result in better solutions.

5.3 Discussion

The two tests have proved the SOM robustness to solve this problem, and consequently, its applicability.

In Test 2, the comparison of the two clustering methods tested allows us to conclude that both have advantages and disadvantages. One of the main drawbacks in using k-means is that it is quite sensitive to local minima, and to a certain degree we verified that in our tests. There are several ways of dealing with this problem, the most common of which is to re-initialize the k-means algorithm several times with different seeds, and then choose the best solution. The main advantage of this method, compared to SOM, is the reduced computing time allied to the fact that it allows to obtain best solutions in most cases. SOM has provided more robust and reproducible results, with the disadvantage of longer computing times.

The location of the remediation units obtained is very similar in the three years considered (2010, 2015 and 2020) in both tests. This can be explained by the fact that the location of the demand nodes is the same in those years; the only thing different is the amount of waste in each demand node. In fact this constitutes one limitation of the model, since we cannot predict and incorporate the changes in soil occupation with the pass of time. Although the CCA-treated wood waste varies in the three years considered, the origin or location of these wastes is always the same in this model formulation, which may not correspond to reality.

Comparing the grids used in the two tests, in the case of 10 and 15 remediation units, that is 5x2 vs. 10x1 and 5x3 vs. 15x1, respectively, we can see that the results are very similar (Figures 5.13 and 5.14), particularly for 10 remediation units. It was expected that the two-dimensional and three-dimensional grids would force proximity between the remediation units, which can be seen, to a certain level, in CC9 results (Figure 5.14) and was verified in Gomes *et al.* (2004). The results show that the 15x1 grid produces better results that the 5x3 grid, i.e., lower average distances to remediation units (about 1 km less). This can be explained by the fact that the "tension" exerted in each unit by the neighbouring units is much higher in the case of the matrix configuration (Bação *et al.*, 2004). This tension limits the plasticity of the SOM to adapt to the particular distribution of the dataset and, when using a small number of units, it is easier to adapt a line than a matrix.

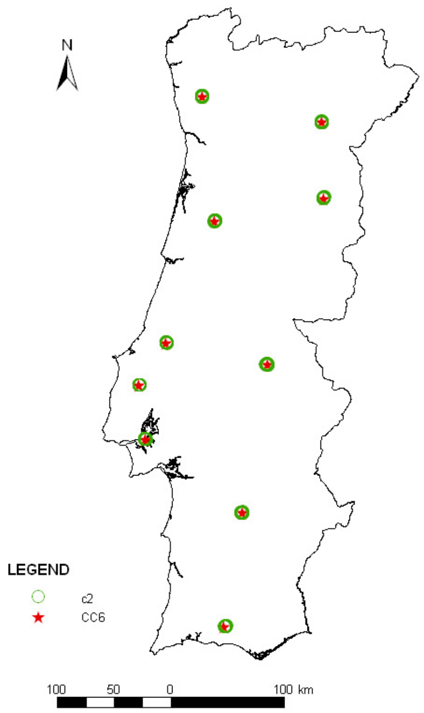

Figure 5.13 – Comparison of the 10x1 and 5x2 SOM grids.

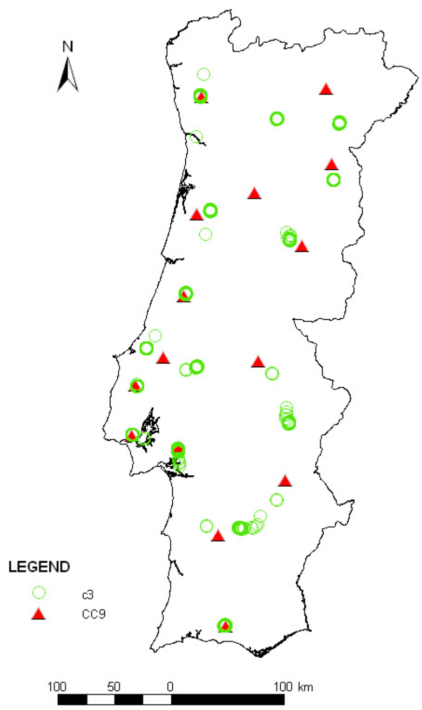

Figure 5.14 – Comparison of the 15x1 and 5x3 SOM grids.

Figures 5.15 to 5.17 present the final maps with all the results of Test 1 and Test 2 for 5; 10 or 15 remediation units. The overlay of all solutions allowed the delimitation of the potential "optimal" areas to implement the remediation units in order to assure distance minimization and, consequently, cost minimization. While it is quite simple to establish these areas in the case of 5 remediation units, when we have 10 and 15 remediation units, it is more difficult to define areas given the bigger dispersion of the solutions. Table 5.3 presents the municipalities covered by the potential areas.

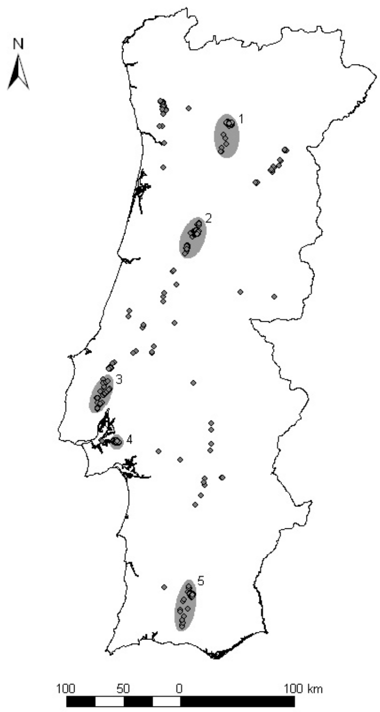

Figure 5.15 – Delimitation of the areas to implement 5 remediation units.

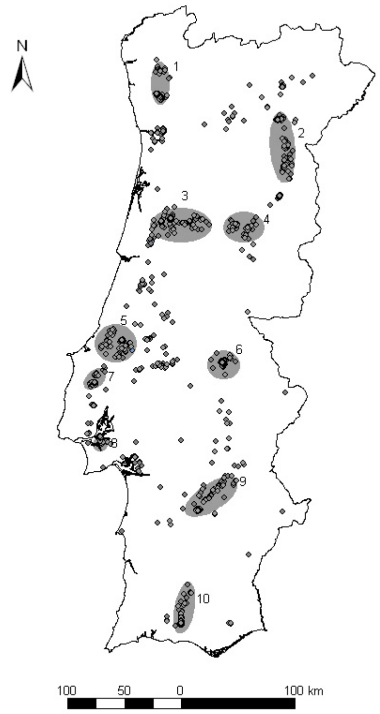

Figure 5.16 – Delimitation of the areas to implement 10 remediation units.

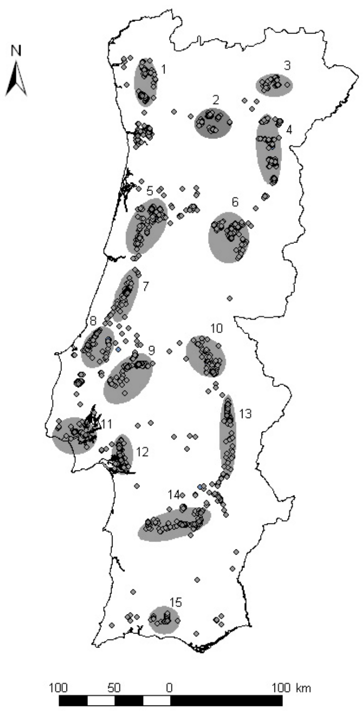

Figure 5.17 – Delimitation of the areas to implement 15 remediation units.

Figure 5.18 presents the overlay of the solutions found in all experiments with the initial location of the demand nodes (presented in Figure 4.7). It allows us to verify that the remediation units are located in the areas with bigger density of CCA-treated wood waste sources, which corresponds to the expected results.

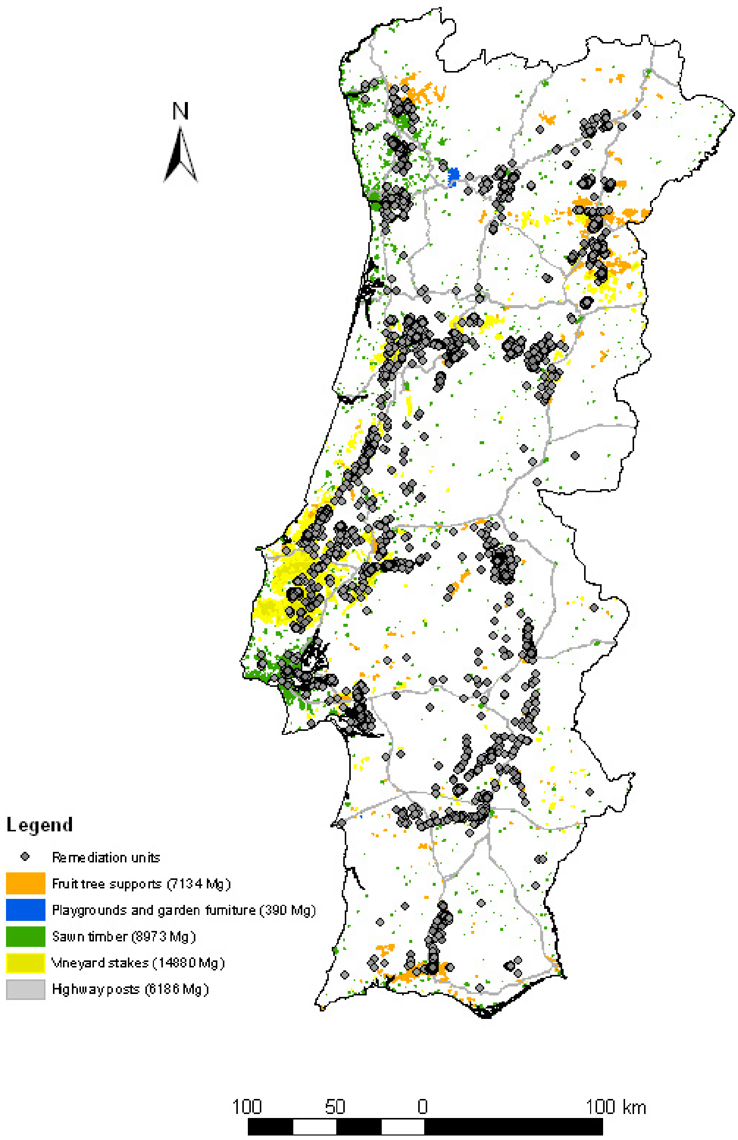

Figure 5.18 – Overlay of the treated wood locations with the remediation units location obtained in Test 1 and Test 2.

Table 5.3 – Municipalities covered by the potential areas to locate the remediation units.

Number of remediation units	Municipalities covered
5	Lousada, Baião, Lamego, Mortágua, Tondela, Tábua, Alenquer, Mafra, Bombarral, Moita, Alcochete, Almodovar, Lagoa
10	Braga, Vizela, Meda, Pinhel, Mira, Anadia, Mealhada, Seia, Covilhã, Fundão, Leiria, Pombal, Alcanena, Cadaval, Crato, Gavião, Alvito, Cuba, Portel, Lagoa
15	Braga, Vizela, Murça, Ribeira de Pena, Lousada, Baião, Lamego, Meda, Torre de Moncorvo, Seia, Covilhã, Fundão, Mira, Mealhada, Pombal, Leiria, Alcanena, Óbidos, Cadaval, Gavião, Crato, Arraiolos, Évora, Cartaxo, Alenquer, Setúbal, Moita, Seixal, Aljustrel, Ferreira do Alentejo, Lagoa

6. CONCLUSIONS AND FURTHER DEVELOPMENTS

This dissertation represents a first approach to the emerging issue of CCA-treated wood waste management. However, research is needed to address current and future disposal issues associated with integrated management of CCA-treated waste in Portugal, namely a more deep assessment of the estimates made through the mass-balance approach.

The main conclusions are now presented:

1. One of the most important conclusions is the **applicability of the tested methods to solve this location problem**. The solutions obtained with our data and with both clustering methods make sense and could be used to decide on the location of these units. The tested methods could also be used in other similar problems, like the location of the two Integrated Centres of Recovering, Valorisation and Elimination of Hazardous Wastes ("CIRVER") to manage hazardous waste in Portugal since the location of the demand nodes is known, as well as the waste amounts produced at each place.

2. **SOM has provided more robust and reproducible results that k-means, with the disadvantage of longer computing times**. The main advantage of k-means, compared to SOM, is the reduced computing time allied to the fact that it allows to obtain best solutions (minimum average distance to a remediation unit) in the majority of the cases, in spite of bigger variances and further geographical dispersion.

3. Another important conclusion is that the **population data can be a good approximation to this kind of location problems** where the demand is related to consumption and/or waste generation. In both tests, the solutions obtained with population data were in close proximity to the solutions obtained with CCA-treated wood waste demand nodes, particularly when the number of remediation units is larger. This can be important when information about the location or weighted demand of

demand nodes is not available. Population data can then be used as an approximation to the problem data.

4. Although the location model was not directly developed in GIS software, through an Avenue script for example, the **articulation between the GIS model and SOMToolbox for MATLAB was easy to implement**.

5. The approach used has great potential because it **can easily be adapted to a more complex (and realistic) formulation of our problem**. On one hand, **additional sources of preserved wood waste can be added**, like the locations of the telephone poles. On the other hand, both algorithms used can **apply distances derived from a distance matrix** with actual road distances between possible locations, thus turning the problem more realistic.

6. We could also verify in our tests that **the increase in the number of epochs, both in SOM and k-means, does not enhance the results** and increases significantly the computing time.

As further developments, a more realistic formulation of the problem would be the use of a network model and not a continuous location model, considering the distances on that network instead of the Euclidean distances. In this case, the remediation units would be located in the nodes of the network and we would be dealing with the classic p-median problem.

Furthermore, to obtain a more realistic formulation of the model, we could consider a capacitated location model, defining constraints about the capacity of the remediation units to treat the amounts of CCA-treated wood waste located in the demand nodes. The location of the wood recycling industries that will use the decontaminated CCA-treated wood as raw material could also be included in the model formulation, allowing including other cost minimization objective.

Another development to this location model would be to explore the results in GIS software using thematic geographic information to identify specific constraints to the location of the CCA-treated wood waste remediation units. This would improve the information available to the decision maker(s).

Finally, the model developed in this dissertation must be regarded as a decision aid tool that can help decision makers to better understand a situation, rather than as a black box that finds the optimum solution. The model can merely offer a number of solutions that may be considered satisfactory; however, the ultimate decision (and responsibility) always lies with the decision maker(s). As all models, it has limitations and only considers part of the immense number of factors that constitute the real world.

7. REFERENCES

Adamopoulos, S. & E. Voulgaridis (1998). Impregnation of timber and regulations applied to preservation practice in Greece. *Proceedings of the 4th International Symposium on Wood Preservation, Cannes - Mandelieu, France, 2nd-3rd February.* IRG Secretariat, Stockholm, Sweden: pp. 31-41.

Aerts, J. C. J. H. & G. B. M. Heuvelink (2002). Using simulated annealing for resource allocation. *International Journal of Geographical Information Science,* **16** (6): pp. 571-587.

Ahmadi, S. & I. H. Osman (in press). Greedy random adaptive memory programming search for the capacitated clustering problem. *European Journal of Operational Research*.

Ahn, H.K., S.W. Cheng, O. Cheong, M. Golin & R. Oostrum (2004). Competitive facility location: the Voronoi game. *Theoretical Computer Science,* **310**: pp. 457– 467.

Alp, O., E. Erkut & Z. Drezner (2003). An efficient genetic algorithm for the p-median problem. *Annals of Operations Research,* **122**: pp. 21–42.

Al-Sultan, K. S. & M. A. Al-Fawzan (1999). A tabu search approach to the uncapacitated facility location. *Annals of Operations Research,* **86**: pp. 91–103.

Andersson, G., R. L. Francis, T. Normark & M. B. Rayco (1998). Aggregation method experimentation for large-scale network location problems. *Location Science,* **6**: pp. 25-39.

Avella, P., A. Sassano & I. Vasil'ev (2003). *Computational study of large-scale p-median problems*. Technical Report 08-03, DIS - Universita di Roma "La Sapienza", Roma, Italia.

AWWA (2004). *Fact sheet on arsenic*. American Water Works Association. URL: http://www.awwa.org/Advocacy/pressroom/arsenic.cfm. Accessed 5th November 2003.

Bação, F., V. Lobo & M. Painho (2004). Clustering census data: comparing the performance of self-organising maps and k-means algorithms. *KDNet Symposium: Knowledge - Based Services for the Public Sector, 3rd-4th June, Bonn*, Germany. URL: http://www.isegi.unl.pt/docentes/vlobo/Publicacoes/publicacoes.htm. Accessed 29th September 2004.

Baldacci, R., E. Hadjiconstantinou, V. Maniezzo & A. Mingozzi (2002). A new method for solving capacitated location problems based on a set partitioning approach. *Computers & Operations Research,* **29**: pp. 365-386.

Beck, B. D. (2002). Potential human health risks from exposures to CCA-treated wood at residences and playgrounds. *American Wood Preservers Association (AWPA) Annual Meeting*, *Memphis, Tennessee, April 22nd*. URL: http://www.agctr.lsu.edu/enr/woodproducts/preservative_treated.asp Accessed 30th October 2003.

Bender, T., H. Hennes, J. Kalcsics, M. T. Melo & S. Nickel (2002). Location software and interface with GIS and supply chain management. *Facility Location: Applications and Theory*, Z. Dezner & H. W. Hamacher (Eds), Springer-Verlag, Heidelberg: pp. 233-274.

Berman, O., Z. Drezner & G. O. Wesolowsky (2002). The collection depots location problem on networks. *Naval Research Logistics,* **49**: pp.15-24.

Bespamyatnikh, S., B. Bhattacharya, M. Keil, D. Kirkpatrick & M. Segal (2002). Efficient algorithms for centers and medians in interval and circular-arc graphs. *Networks*, **39** (3): pp. 144-152.

Bicciato, S., M. Pandin, G. Didone & C. Bello (2001). Analysis of an associative memory neural network for pattern identification in gene expression data. *BIOKDD01 - Workshop on Data Mining in Bioinformatics*. *San Francisco, CA, USA, August 26th*. URL:http://www.cs.rpi.edu/~zaki/BIOKDD01/. Accessed 23rd August 2004.

Bishop, C. M. (1996). *Neural networks: a pattern recognition perspective*. Technical Report NCRG/96/001, Neural Computing Research Group, Aston University, Birmingham, UK.

Borrego, C., L. Arroja & M. A. Matos (2003). *Inventário de resíduos sólidos industriais*. Departamento de Ambiente e Ordenamento, Universidade de Aveiro, Aveiro.

Bowerman, R. L., P. H. Calamai & G. B. Hall (1999). The demand partitioning method for reducing aggregation errors in p-median problems. *Computers & Operations Research,* **26**: pp. 1097-1111.

Bozkaya, B., J. Zhang & E. Erkut (2002). An efficient genetic algorithm for the p-median problem. *Facility Location: Applications and Theory*, Z. Dezner & H. W. Hamacher (Eds), Springer-Verlag, Heidelberg: pp. 180-205.

Brandeau, M. L. & S. S. Chiu (1989). An overview of representative problems in location research. *Management Science,* **35** (6): pp. 645-674.

Breslin, V. T. & L. Adler-Ivanbrook (1998). Release of copper, chromium and arsenic from CCA-C treated lumber in estuaries. *Estuarine, Coastal and Shelf Science,* **46**: pp. 111–125.

Brimberg, J., P. Hansen, N. Mladenovic' & É. D. Taillard (2000). Improvements and comparison of heuristics for solving the multisource Weber problem. *Operations Research,* **48** (1): pp. 129-135.

Burkard, R. E., E. Çela & H. Dollani (2000). 2-medians in trees with pos/neg weights. *Discrete Applied Mathematics,* **105**: pp. 51-71.

Burkard, R. E. & H. Dollani (2001). Robust location problems with pos/neg weights on a tree. *Networks,* **38** (2): pp. 102-113.

Burrough, P. A. (1986). *Principles of geographic information systems for land resources assessment*. Oxford University Press, New York.

Campbell, J. F., A. T. Ernst & M. Krishnamoorthy (2002). Hub location problems. *Facility Location: Applications and Theory*, Z. Dezner & H. W. Hamacher (Eds), Springer-Verlag, Heidelberg: pp. 81-118.

Cappanera, P. (1999). *A survey on obnoxious facility location problems*. Technical Report: TR-99-11, Dipartamento di Informatica, Università di Pisa, Pisa, Italia.

Cappanera, P., G. Gallo & F. Maffioli (2004). Discrete facility location and routing of obnoxious activities. *Discrete Applied Mathematics,* **133**: pp. 3-28.

Carreras, M. & D. Serra (1999). On optimal location with threshold requirements. *Socio-Economic Planning Sciences,* **33**: pp. 91-103.

Carvalho, A. (1997). *Madeiras portuguesas: estrutura anatómica, propriedades, utilizações*. Volume II. Direcção-Geral das Florestas, Lisboa.

Chiou, Y. C. & L. W. Lan (2001). Genetic clustering algorithms. *European Journal of Operational Research,* **135**: pp. 413-427.

Chirenjea, T., L. Q. Maa, C. Clarkb & M. Reevesa (2003). Cu, Cr and As distribution in soils adjacent to pressure-treated decks, fences and poles. *Environmental Pollution,* **124**: pp. 407–417.

Chiyoshi, F. & R. D. Galvão (2000). A statistical analysis of simulated annealing applied to the p-median problem. *Annals of Operations Research,* **96**: pp. 61–74.

Church, R. L. (1999). Location modelling and GIS. *Geographic information systems: principles and technical issues*. Volume 1, P. A. Longley, M. F. Goodchild, D. J. Maguire & D. W. Rhind (Eds), John Wiley & Sons, New York: pp. 293-303.

Church, R. L. (2003). COBRA: A new formulation of the classic p-median location problem. *Annals of Operations Research,* **122**: pp. 103-120.

Church, R. L. & C. S. ReVelle (1976). Theoretical and computational links between the p-median location set-covering and the maximal covering location problem. *Geographical Analysis,* **8**: pp. 406-415.

Church, R. L. & P. Sorensen (1994). *Integrating normative location models into GIS: problem and prospects with p-median problem*. Technical Report

94-5, National Center for Geographic Information and Analysis, University of California, Santa Barbara, U.S.A..

Clausen, C. A. (2000). CCA removal from treated wood using a dual remediation process. *Waste Management & Research,* **18**: pp. 485-488.

Clausen, C. A., S. N. Kartal & J. Muehl (2001). Particleboard made from remediated CCA-treated wood: evaluation of panel properties. *Forest Products Journal,* **51** (7/8): pp. 61-64.

Clausen, C. A. & J. Muehl (2000). Properties of particleboard made from recycled CCA-treated wood. *31st Annual Meeting of the International Research Group on Wood Preservation*, *Kona, Hawaii, USA, 14th-19th May*. URL: http://www.fpl.fs.fed.us/documnts/pdf2000/claus00a.pdf. Accessed 8th December 2003.

Clausen, C. A. & R. L. Smith (1998). Removal of CCA from treated wood by oxalic acid extraction, steam explosion, and bacterial fermentation. *Journal of Industrial Microbiology & Biotechnology,* **20**: pp. 251–257.

Clausen, J. (1999). *Branch and bound algorithms: principles and examples*. Department of Computer Science, University of Copenhagen. Copenhagen, Denmark.

Cooper, L. (1963). Location-allocation problems. *Operations Research,* **11**: pp. 331-343.

Cooper, P. A. (1999). Future of wood preservation in Canada: disposal issues. *20th Annual Canadian Wood Preservation Association Conference*, *Vancouver BC, October 25th – 26th. URL:* http://www.forestry.utoronto.ca/treated_wood/future.pdf. Accessed 30th October 2003.

Cooper, P. A. (2003). A review of issues and technical options for managing spent CCA treated wood. *AWPA Annual Meeting*, AWPA. *Boston, Massachusetts, April 27th-29th.* URL: http://www.awpa.com/papers/Cooper_2003.pdf. Accessed 30th October 2003.

Correa, E. S., M. T. A. Steiner, A. A. Freitas & C. Carnieri (2001). A genetic algorithm for the p-median problem. *Genetic and Evolutionary Computation Conference (GECCO-2001)*, *San Francisco, USA, San Francisco, USA, July 7th-11th*, L. Spector & E. D. Goodman (Eds). Morgan Kaufmann Publishers, San Francisco: pp. 1268-1275.

Cortinhal, M. J. & M. E. Captivo (2003). Upper and lower bounds for the single source capacitated location problem. *European Journal of Operational Research,* **151**: pp. 333-351.

Cox, C. (1991). Chromated copper arsenate. *Journal of Pesticide Reform,* **11** (1): pp. 2-6.

Crainic, T. G., M. Gendreau, P. Hansen, N. Hoeb & N. Mladenovic´ (2001). Parallel variable neighbourhood search for the p-median. *Proceedings of the 4th Metaheuristics International Conference, Porto, MIC 2001 July 16th-20th,* J. P. de Sousa (Ed), Universidade do Porto, Porto: pp. 595-599.

Crawford, D. M., B. M. Woodward & C. A. Hatfield (2000). *Comparison of wood preservatives in stake tests: 2000 progress report.* Res. Note FPL-RN-02, United States Department of Agriculture, Forest Service, Forest Products Laboratory. Madison, W., U.S.A..

Cruz, F. R. B., G. R. Mateus & J. M. Smith (2003). A Branch-and-Bound algorithm to solve a multi-level network optimization problem. *Journal of Mathematical Modelling and Algorithms,* **2**: pp. 37-56.

Current, J., M. Daskin & D. Schilling (2002). Discrete network location models. *Facility Location: Applications and Theory*, Z. Dezner & H. W. Hamacher (Eds), Springer-Verlag, Heidelberg: pp. 81-118.

Current, J., H. Min & D. Schilling (1990). Multiobjective analysis of facility location decisions. *European Joumal of Operational Research,* **49**: pp. 295-307.

Current, J. R. & D. A. Schilling (1987). Elimination of source A and B errors in p-median location problems. *Geographical Analysis,* **19**: pp. 95-110.

Curry, B. & P. H. Morgan (2004). Evaluating Kohonen's learning rule: an approach through genetic algorithms. *European Journal of Operational Research,* **154**: pp. 191-205.

Densham, P. J. & G. Rushton (1991). *Designing and implementing strategies for solving large location-allocation problems with heuristic methods*. Report 91-10, National Center for Geographic Information and Analysis, University of California, Santa Barbara, U.S.A..

Deroubaix, G. (1998). Possible regulatory status of treated wood waste and implications. *Proceedings of the 4th International Symposium on Wood Preservation, Cannes - Mandelieu, France, 2nd-3rd February.* IRG Secretariat, Stockholm, Sweden: pp. 68-76.

Deroubaix, G. (2004). Impregnated wood waste management processes under development in France. *Final Conference COST Action E22 "Environmental Optimisation of Wood Protection"*, Lisboa, Portugal, 22nd-23rd March. URL: http://www.bfafh.de/cost22.htm. Accessed 3rd April 2004.

Deroubaix, G. & C. Cornillier (2004). Treated wood waste in Europe. Proposal for a management concept. *Final Conference COST Action E22 "Environmental Optimisation of Wood Protection"*, Lisboa, Portugal, 22nd-23rd March. URL: http://www.bfafh.de/cost22.htm. Accessed 3rd April 2004.

Díaz-Báñez, J. M., J. A. Mesa & A. Schobel (2004). Continuous location of dimensional structures. *European Journal of Operational Research,* **152**: pp. 22-44.

Drezner, T. & H. A. Eiselt (2002). Consumers in competitive location models. *Facility Location: Applications and Theory*, Z. Dezner & H. W. Hamacher (Eds), Springer-Verlag, Heidelberg: pp. 151-178.

Drezner, Z. (1995a). Dynamic facility location: the progressive p-median problem. *Location Science,* **3** (1): pp. 1-7.

Drezner, Z. (1995b). On the conditional p-median problem. *Computers & Operations Research,* **22** (5): pp. 525-530.

Drezner, Z., K. Klamroth, A. Schobel & G. O. Wesolowsky (2002). The Weber problem. *Facility Location: Applications and Theory*, Z. Dezner & H. W. Hamacher (Eds), Springer-Verlag, Heidelberg: pp. 1-36.

Eiselt, H. A. & G. Laport (1995). Objectives in location problems. *Facility Location: A survey of applications and methods*, Z. Drezner (Eds), Springer Series in Operations Research, New York: pp. 151-180.

Eiselt, H. A., G. Laporte & J. F. Thisse (1993). Competitive location models: a framework and bibliography. *Transportation Science,* **27** (1): pp. 44-54.

Erkut, E. & B. Bozkaya (1999). Analysis of aggregation errors for the p-median problem. *Computers & Operations Research,* **26**: pp. 1075-1096.

Estivill-Castro, V. & M. E. Houle (2001). Robust distance-based clustering with applications to spatial data mining. *Algorithmica,* **30** (2): pp. 216-242.

Evans, F. G. (2004). Waste management in Norway. *Final Conference COST Action E22 "Environmental Optimisation of Wood Protection"*, *Lisboa, Portugal, 22nd-23rd March.* URL: http://www.bfafh.de/cost22.htm. Accessed 3rd April 2004.

EWG & HBN (2001). *Poisoned playgrounds: arsenic in 'pressure-treated' wood*. Environmental Working Group (EWG) and Healthy Building Network (HBN), Washington D.C., U.S.A..

Felton, C. C. & R. C. D. Groot (1996). The recycling potential of preservative-treated wood. *Forest Products Journal,* **46** (7/8): pp. 37-46.

Fogel, D. B. (1999) *Evolutionary computation: toward a new philosophy of machine intelligence.* 2nd Edition. Wiley - IEEE Press, New York.

FPL (2000). *Environmental impact of preservative-treated wood in a wetland boardwalk*. FPL–RP–582, Forest Service, Forest Products Laboratory Forest Service, Forest Products Laboratory, United States Department of Agriculture, Madison, W.I., U.S.A..

Frasconi, P., M. Gori & A. Sperduti (1997). On the efficient classification of data structures by neural networks. *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI'97)*. *Aichi, Japan, August 23rd-29th*. URL: http://www.dsi.unifi.it/~paolo/datas/ijcai97.ps.gz. Accessed the 20th August 2004.

García-López, F., B. M. Batista, J. A. Moreno-Pérez & J. M. Moreno-Vega (2002). The parallel variable neighborhood search for the p-median problem. *Journal of Heuristics,* **8**: pp. 375–388.

García-López, F., B. Melián-Batista, J. A. Moreno-Pérez & J. M. Moreno-Vega (2003). Parallelization of the scatter search for the p-median problem. *Parallel Computing,* **29**: pp. 575–589.

Gendebien, A., A. Leavens, K. Blackmore, A. Godley, K. Lewin, B. Franke & A. Franke (2002). *Study on Hazardous Household Waste (HHW) with a main emphasis on Hazardous Household Chemicals (HHC)*. WRc Ref: CO 5089-2, Directorate General Environment, European Commission, Brussels, Belgium.

Ghiani, G., F. Guerriero & R. Musmanno (2002). The capacitated plant location problem with multiple facilities in the same site. *Computers & Operations Research,* **29**: pp. 1903–1912.

Ghosh, D. (2003). Neighborhood search heuristics for the uncapacitated facility location problem. *European Journal of Operational Research,* **150**: pp. 150-162.

Giannikos, I. (1998). A multiobjective programming model for locating treatment sites and routing hazardous wastes. *European Journal of Operational Research,* **104**: pp. 333-342.

Gil, L. & C. R. Duarte (1997). Tratamento de madeiras com resíduos da indústria corticeira. *Silva Lusitana,* **5** (1): pp. 59-69.

Glover, F. and M. Laguna. (1997). *Tabu search*. Kluwer Academic Publishers. Boston.

Glover, F. (1999). Scatter search and path relinking. *New Ideas in Optimization*, D. Corne, M. Dorigo & F. Glover (Eds), McGraw Hill, London: pp. 297-316.

Glover, F., M. Laguna & R. Martí (2004). Scatter search and path relinking: foundations and advanced designs. *New Optimization Techniques in Engineering*, G. Onwubolu & B. V. Babu (Eds), Springer-Verlag, Berlin - Heidelberg: pp. 87-100.

Gomes, H., V. Lobo & A. Ribeiro (2004). Application of clustering methods for optimizing the location of treated wood remediation units. *JOCLAD 2004 - XI Jornadas de Classificação e Análise de Dados*. *Lisboa, 31 de Março a*

3 de Abril 2004. Associação Portuguesa de Classificação e Análise de Dados, Lisboa: pp. 109-115.

Hakimi, S. L. (1964). Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research,* **12**: 450-459.

Hakimi, S. L. (1965). Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations Research,* **13**: pp. 462-475.

Hakimi, S. L., M. Labbe´ & E. F. Schmeichel (1999). Locations on time-varying networks. *Networks,* **34**: pp. 250–257.

Hale, T. S. & C. R. Moberg (2003). Location science research: a review. *Annals of Operations Research,* **123**: pp. 21-35.

Hamacher, H. W., M. Labbé, S. Nickel & A. J. V. Skriver (2002). Multicriteria Semi-obnoxious Network Location Problems (MSNLP) with sum and center objectives. *Annals of Operations Research,* **110**: pp. 33–53.

Hamacher, H. W. & S. Nickel (1996). Multicriteria planar location problems. *European Journal of Operational Research,* **94**: pp. 66-86.

Hamacher, H. W. & S. Nickel (1998). Classification of location models. *Location Science,* **6**: pp. 229-242.

Hansen, P. & N. Mladenovic' (2001). J-means: a new local search heuristic for minimum sum-of-squares clustering. *Pattern Recognition Letters,* **34** (2): pp. 405-413.

Hansen, P. & N. Mladenovic' (2003). Variable neighbourhood search. *State-of-the-art Handbook of Metaheuristics*, F. Glover & G. Kochenberger (Eds), Kluwer Academic Publishers, Dordrecht.

Hansen, P., N. Mladenovic' & É. Taillard (1998). Heuristic solution of the multisource Weber problem as a p-median problem. *Operations Research Letters,* **22**: pp. 55-62.

Hansen, P. & N. Mladenovic´ (1997). Variable neighborhood search for the p-median. *Location Science,* **5** (4): pp. 207-226.

Hansen, P., N. Mladenovic´ & D. Perez-Brito (2001). Variable neighborhood decomposition search. *Journal of Heuristics,* **7**: pp. 335–350.

Helsen, L., E. V. Bulck & J. S. Heryb (1998). Total recycling of CCA treated wood waste by low-temperature pyrolysis. *Waste Management & Research,* **18**: pp. 571-578.

Hidaka, K. & H. Okano (2003). An approximation algorithm for a large-scale facility location problem. *Algorithmica,* **35**: pp. 216–224.

Hodgson, M. J. & J. Hewko (2003). Aggregation and surrogation error in the p-median model. *Annals of Operations Research,* **123**: pp. 53–66.

Hohenthal, C. (2001). Waste management options in Finland. *Waste Management Options in Europe: Information Compiled in COST Action E9*. A. Merl (Ed), COST E9 Working Group 3 “End of life: recycling, disposal and energy production”, Brussels: pp. 23-34.

Hsieh, K.-H. & F.-C. Tien (2004). Self-organizing feature maps for solving location-allocation problems with rectilinear distances. *Computers & Operations Research,* **31**: pp.1017-1031.

Humar, M. & F. Pohleven (2001). Recycling of CCA or CCB treated waste wood. *Commodity Science in Global Quality Perspective: Products - Technology, Quality and Environment: Proceedings of the 13th IGWT Symposium*, *Maribor, Slovenia, September 2nd- 8th,* M. Denac, V. Musil & B. Pregrad (Eds), Ekonomsko-poslovna fakulteta, Maribor, Slovenia: pp. 1409-1413.

Illman, B. L., V. W. Yang & L. Ferge (2000). *Bioprocessing preservative-treated waste wood*. Report IRG/WP 00-50145, International Research Group on Wood Preservation, Stockholm, Sweden.

INE (2002). *Infra-estruturas rodoviárias 2001*. Instituto Nacional de Estatística. Lisboa.

INE (2004a). *Inquérito à base Belém*. Ficheiro das sociedades classificadas nas CAE 20102, 20201, 20202 e 20203 ordenadas por escalão de EVVN. Instituto Nacional de Estatística, Lisboa.

INE (2004b). *Inquérito à produção industrial 2002*. Instituto Nacional de Estatística, Lisboa.

INETI (2001). *Plano Nacional de Prevenção de Resíduos Industriais*: Volume II, Tomo I. Instituto Nacional de Engenharia e Tecnologia Industrial. Lisboa.

Jankowski, P., N. Andrienko & G. Andrienko (2001). Map-centred exploratory approach to multiple criteria spatial decision making. *International Journal of Geographical Information Science,* **15** (2): pp. 101-127.

Jermer, J., A. Ekvall, B. Bahr & C. Tullin (2004). Waste wood management in Sweden: an update. *Final Conference COST Action E22 "Environmental Optimisation of Wood Protection"*, *Lisboa, Portugal, 22nd-23rd March 2004*. URL: http://www.bfafh.de/cost22.htm. Accessed 3rd April 2004.

Kamdem, D. P., H. Jiang, W. Cui, J. Freed & L. M. Matuana (2004). Properties of wood plastic composites made of recycled HDPE and wood flour from CCA-treated wood removed from service. *Composites: Part A,* **35**: pp. 347–355.

Karasakal, O. & E. K. Karasakal (2004). A maximal covering location model in the presence of partial coverage. *Computers & Operations Research,* **31**: pp. 1515–1526.

Kariv, O. & S. L. Hakimi (1979). An algorithmic approach to network location problems. II: The p-medians. *SIAM Journal on Applied Mathematics*, **37**: pp. 539-560.

Kartal, S. N. (2003). Removal of copper, chromium, and arsenic from CCA-C treated wood by EDTA extraction. *Waste Management & Research,* **23**: pp. 537–546.

Kartal, S. N. & C. A. Clausen (2001). Leachability and decay resistance of particleboard made from acid extracted and bioremediated CCA-treated wood. *International Biodeterioration & Biodegradation,* **47**: pp.183–191.

Kartal, S. N., T. Kakitani & Y. Imamura (2004). Bioremediation of CCA-C treated wood by *Aspergillus niger* fermentation. *Holz Roh Werkst,* **62**: pp. 64–68.

Kartal, S. N. & C. Kose (2003). Remediation of CCA-C treated wood using chelating agents. *Holz als Roh- und Werkstoff,* **61**: pp. 382–387.

Kazi, F. K. M. & P. A. Cooper (1999). Recovery and reuse of Chromated Copper Arsenate (CCA) wood preservative from CCA treated wastes. *20th Annual Canadian Wood Preservation Association Conference*, *Vancouver BC, October 25th-26th.* URL: www.forestry.utoronto.ca/treated_wood/ccawaste.pdf. Accessed 11th November 2003.

Kim, Y. H. & S. Openshaw (1999a). Developing hybrid intelligent location optimisers for spatial modelling in GIS. *11th European Colloquium on Theoretical and Quantitative Geography*, *Durham, UK, September 3rd-7th*. URL: http://www.geog.leeds.ac.uk/presentations/99-1/. Accessed 20th October 2003.

Kim, Y. H. & S. Openshaw (1999b). Visualising performances of intelligent location model in a GIS. *GIS Research UAK'99*, *University of Southampton, UK, April 14th–16th*. URL: http://www.geog.leeds.ac.uk/presentations/99-3/. Accessed 20th October 2003.

Kohonen, T., J. Hynninen, J. Kangas & J. Laaksonen (1995). *SOM_PAK: the Self-Organizing Map program package*. SOM Programming Team of the Helsinki University of Technology, Laboratory of Computer and Information Science, Helsinki University of Technology, Helsinki, Finland.

Kohonen, T. (2001). *Self-organizing maps*. Springer, Berlin-Heidelberg.

Krarup, J., D. Pisinger & F. Plastria (2002). Discrete location problems with push-pull objectives. *Discrete Applied Mathematics,* **123**: pp. 267-378.

Laguna, M. (2002). Scatter search. *Handbook of Applied Optimization*, P. M. Pardalos & M. G. C. Resende (Eds), Oxford University Press, Oxford. pp. 183-193.

Lahdelma, R., P. Salminen & J. Hokkanen (2002). Locating a waste treatment facility by using stochastic multicriteria acceptability analysis with ordinal criteria. *European Journal of Operational Research,* **142**: pp. 345–356.

Leithoff, H., I. Stephan, M. T. Lenz & R. Peek (1995). *Growth of the copper tolerant brown-rot fungus Antrodia vaillantii on different substrates*. Document No: IRG/WP95-10121, The International Research Group on Wood Preservation, IRG Secretariat, Stockholm, Sweden.

Llurdés, J., D. Sauri & R. Cerdan (2003). Ten years wasted: the failure of siting waste facilities in central Catalonia, Spain. *Land Use Policy,* **20**: pp. 335-342.

Lorena, L. A. N. & E. L. F. Sene (2003). Local search heuristics for capacitated p-median problems. *Networks and Spatial Economics,* **3**: pp. 407-419.

Lorena, L. A. N. & E. L. F. Senne (2004). A column generation approach to capacitated p-median problems. *Computers & Operations Research,* **31**: pp. 863-876.

Lozano, S., F. Guerrero, L. Onieva & J. Larrañeta (1998). Kohonen maps for solving a class of location-allocation problems. *European Journal of Operational Research,* **108**: pp. 106-117.

Maas, R. P., S. C. Patch, A. M. Stork, J. F. Berkowitz & G. A. Stork (2002). *Release of total chromium, chromium VI and total arsenic from new and aged pressure treated lumber*. Report 02-093, Environmental Quality Institute, University of North Carolina-Asheville, Asheville, U.S.A..

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observation. *5th Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley: pp. 281-297.

Mangiameli, P., S. K. Chen & D. West (1996). A comparison of SOM neural networks and hierarchical clustering methods. *European Journal of Operational Research,* **93**: pp. 402-417.

Maniezzo, V., A. Mingozzi & R. Baldacci (1998). A bionomic approach to the capacitated p-median problem. *Journal of Heuristics,* **4**: pp. 263–280.

Marianov, V. & D. Serra (2002). Location problems in the public sector. *Facility Location: Applications and Theory*, Z. Dezner & H. W. Hamacher (Eds), Springer-Verlag, Heidelberg: pp. 119-150.

Martins, J. (2004). [*Pedido de informação*] [Email, 16-03-2004]. ISEGI-UNL, Lisboa.

McDarby, F. (2001). Waste management options in Ireland. *Waste Management Options in Europe: Information Compiled in COST Action E9*. A. Merl (Ed), COST E9 Working Group 3 “End of life: recycling, disposal and energy production”, Brussels: pp. 50-55.

McKeever, D. B., J. A. Youngquist & B. W. English (1995). Sources and availability of recovered wood and fiber for composite products. *Proceedings of the 29th International Particleboard/composite Materials Symposium, Pullman, April 4th-6th, Washington State University.* URL: http://wolcott.wsu.edu/papers.html. Accessed 30th October 2003.

McQueen, J., J. Stevens & D. P. Kamdem (1998). Recycling of CCA treated wood in the US. *Proceedings of the 4th International Symposium on Wood Preservation, Cannes - Mandelieu, France, 2nd-3rd February.* IRG Secretariat, Stockholm, Sweden: pp. 78-93.

Megiddo, N. & K. Supowit (1984). On the complexity of some common geometric location problems. *SIAM Journal on Computing,* **18**: pp. 182-196.

Melachrinoudis, E. & Z. Xanthopulos (2003). Semi-obnoxious single facility location in Euclidean space. *Computers & Operations Research,* **30**: pp. 2191-2209.

Mladenovic´, N., M. Labbe´ & P. Hansen (2003). Solving the p-center problem with tabu search and variable neighborhood search. *Networks,* **42** (1): pp. 48–64.

MTBI (2002). *Best management practices for the use of preservative-treated wood in aquatic environments in Michigan with special provisions and design criteria for engineers*. Michigan Timber Bridge Initiative, Michigan, U.S.A..

Murray, A. T. & R. A. Gerrard (1997). Capacitated service and regional constraints in location-allocation modeling. *Location Science,* **5** (2): pp. 103-118.

Narciso, M. G. & L. A. N. Lorena (2002). Uso de algoritmos genéticos em sistema de apoio à decisão para alocação de recursos no campo e na cidade. *Revista Brasileira de Agroinformática,* **4** (2): pp. 90-101.

Nema, A. K. & S. K. Gupta (1999). Optimization of regional hazardous waste management systems: an improved formulation. *Waste Management,* **19**: pp. 441-451.

Nickel, S. & J. Puerto (1999). A unified approach to network location. *Networks,* **34**: pp. 283-290.

Nour, M. A. & G. R. Madey (1996). Heuristic and optimization approaches to extending the Kohonen self organizing algorithm. *European Journal of Operational Research,* **93**: pp. 428-448.

OECD (2000). *Report of the OECD Workshop on Environmental Exposure Assessment to Wood Preservatives*. OECD, Belgirate, Italy.

OECD (2003). *Emission Scenario Document for Wood Preservatives*. OECD Series on Emission Scenario Documents. Number 2, Part 1. Environment Directorate Joint Meeting of the Chemicals Committee and the Working Party on Chemicals, Pesticides and Biotechnology.

Ohsawa, Y. (2000). Bicriteria euclidean location associated with maximin and minimax Criteria. *Naval Research Logistics,* **47**: pp. 581-592.

Okabe, A. & A. Suzuki (1997). Locational optimization problems solved through Voronoi diagrams. *European Journal of Operational Research,* **98**: pp. 445-456.

Ottosen, L. M. & I. V. Christensen (2004). Legislation and research concerning waste wood in Denmark. *Final Conference COST Action E22 "Environmental Optimisation of Wood Protection", Lisboa, Portugal, 22nd-23rd March*. URL: http://www.bfafh.de/cost22.htm. Accessed 3rd April 2004.

Ottosen, L. M., A. J. Pedersen, I. V. Kristensen & A. B. Ribeiro (2003). Removal of arsenic from toxic ash after combustion of impregnated wood. *Journal de Physique IV (France),* **107**: pp. 993-996.

Owen, S. H. & M. S. Daskin (1998). Strategic facility location: A review. *European Journal of Operational Research,* **111**: pp. 423-447.

Peek, R.-D. (1998). The waste wood situation in Germany: Legislative and practical consequences. *Proceedings of the 4th International Symposium on Wood Preservation, Cannes - Mandelieu, France, 2nd-3rd February.* IRG Secretariat, Stockholm, Sweden: pp. 56-66.

Pereira, M. A. & L. A. N. Lorena (2001). A heurística lagrangeana/surrogate aplicada ao problema de localização de máxima cobertura. *XXXIII Simpósio Brasileiro de Pesquisa Operacional - A Pesquisa Operacional e o Meio Ambiente, Campos do Jordão, 6 a 9 de Novembro de 2001.* URL: http://www.lac.inpe.br/~lorena/public.html. Accessed 15th September 2003.

Pérez, J. A. M., J. M. Moreno-Vega & I. R. Martín (2003). Variable neighborhood tabu search and its application to the median cycle problem. *European Journal of Operational Research,* **151**: pp. 365–378.

Piersma, N. (1999). A probabilistic analysis of the capacitated facility location problem. *Journal of Combinatorial Optimization,* **3**: pp. 31–50.

Plastia, F. (2002). Continuous covering location problems. *Facility Location: Applications and Theory,* Z. Drezner & H. W. Hamacher (Eds), Springer-Verlag, Heidelberg: pp. 37-80.

Plastria, F. & E. Carrizosa (2004). Optimal location and design of a competitive facility. *Mathematical Programming: Series A*, **100**, pp. 247-265

PMRA (2002). *Re-evaluation note of Chromated Copper Arsenate (CCA)*. REV2002-03, Pest Management Regulatory Agency, Ottawa, Ontario, Canada.

Procopiuc, C. M. (1997). *Applications of Clustering Problems*. URL: http://citeseer.ist.psu.edu/procopiuc97application.html. Accessed 2nd February 2004.

Pruszinski, A. (1999). *Review of the landfill disposal risks and the potential for recovery and recycling of preservative treated timber*. Report AD00555, Environmental Protection Agency, Government of South Australia, Adelaide, Australia.

Rakas, J., D. Teodorovic´ & T. Kim (2004). Multi-objective modeling for determining location of undesirable facilities. *Transportation Research: Part D,* **9**: pp. 125-138.

Read, D. (2003). *Report on Copper, Chromium and Arsenic (CCA) treated timber*. ERMA New Zealand, Wellington, New Zealand.

Reimão, D. (2003). [*Questões sobre madeira preservada*] [Email, 18-12-2003]. ISEGI-UNL, Lisboa.

Reimão, D. & R. Cockcroft (1985). *Wood Preservation in Portugal*. Information Nº487, The Swedish National Board for Technical Development, Stockholm, Sweden.

Resende, M. G. C. (1998). Computing approximate solutions of the maximum covering problem with GRASP. *Journal of Heuristics,* **4**: pp. 161–177.

Resende, M. G. C. & C. C. Ribeiro (2003). *GRASP and path-relinking: recent advances and applications*. AT&T Labs Research Technical Report Florham Park, N.J., U.S.A..

Resende, M. G. C. & R. F. Werneck (2002a). *A GRASP with path-relinking for the p-median problem*. Technical Report TD5E4QKA, AT&T Labs Research. Florham Park, N.J., U.S.A..

Resende, M. G. C. & R. F. Werneck (2002b). *On the implementation of a swap-based local search procedure for the p-median problem*. Technical Report TD-5E4QKA, AT&T Labs Research, Florham Park, N.J., U.S.A..

Resende, M. G. C. & R. F. Werneck (2004). A hybrid heuristic for the p-median problem. *Journal of Heuristics,* **10**: pp. 59–88.

Ribeiro, A. B. (1998). *Use of electrodialytic remediation technique for removal of selected heavy metals and metalloids from soils*. Department of Geology and Geotechnical Engineering, Technical University of Denmark. Denmark.

Ribeiro, A. B., E. P. Mateus, L. M. Ottosen & G. Bech-Nielsen (2000). Electrodialytic removal of Cu, Cr and As from chromated copper arsenate-treated timber waste. *Environmental Science & Technology,* **34** (5): pp. 784-788.

Righini, G. (1995). A double annealing algorithm for discrete location/allocation problems. *European Journal of Operational Research,* **86**: pp. 452-468.

Rooij, W. (2000). *The CARE algorithm: a heuristic method to solve capacitated set covering location problems on large-scale networks*. Department of Geography, University of Salford, Manchester, UK.

Rosing, K. E. & M. J. Hodgson (2002). Heuristic concentration for the p-median: an example demonstrating how and why it works. *Computers & Operations Research,* **29**: pp. 1317-1330.

Rosing, K. E. & C. S. ReVelle (1997). Heuristic concentration: two stage solution construction. *European Joumal of Operational Research,* **97**: pp. 75-86.

Rosing, K. E., C. S. ReVelle, E. Rolland, D. A. Schilling & J. R. Current (1998). Heuristic concentration and tabu search: a head to head comparison. *European Journal of Operational Research,* **104**: pp. 93-99.

Rosing, K. E., C. S. ReVelle & D. A. Schilling (1999). A gamma heuristic for the p-median problem. *European Journal of Operational Research,* **117**: pp. 522-532.

Salhi, S. (2002). Defining tabu list size and aspiration criterion within tabu search methods. *Computers & Operations Research,* **29**: pp. 67-86.

Sanches, R. (2004). [*Questões sobre madeira preservada*] [Email, 26-01-2004]. ISEGI-UNL, Lisboa.

Santana, F., L. Amaral, F. Ferreira, J. Monjardino & R. Maurício (2003). *Estudo de Inventariação de Resíduos Industriais*. Relatório Síntese, Instituto dos Resíduos. Lisboa.

Sarle, W. (2004). *How many kinds of Kohonen networks exist? (And what is k-means?)*. URL: http://www.faqs.org/faqs/ai-faq/neural-nets/part1/section-11.html. Accessed 20th September 2004.

Senne, E. L. F. & L. A. N. Lorena (2000). Lagrangean/surrogate heuristics for p-median problems. *Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research*, M. Laguna & J. L. Gonzalez-Velarde (Eds), Kluwer Academic Publishers, Dordrecht: pp. 115-130.

Serra, D. & V. Marianov (1998). The p-median problem in a changing network: the case of Barcelona. *Location Science,* **6**: pp. 383-394.

Shiau, R. J., R. L. Smith & B. Avellar (2000). Effects of steam explosion processing and organic acids on CCA removal from treated wood waste. *Wood Science and Technology,* **34**: pp. 377-388.

Simha, R., W. Cai & V. Spitkovsky (2000). Simulated N-Body: new particle physics-based heuristics for a Euclidean location-allocation problem. *Journal of Heuristics,* **7**: pp. 23–36.

Solo-Gabriele, H., J. Penha, V. Catilu, T. Townsend & T. Tolaymat (1998). *Generation, use, disposal and management options for CCA-treated wood*. Report #98-1, Florida Center for Solid and Hazardous Waste Management, Gainesville, U.S.A..

Solo-Gabriele, H., V. Calitu, M. Kormienko, T. Townsend & B. Messick (1999). *Disposal of CCA-treated wood: An evaluation of existing and alternative management options*. Report #99-6, Florida Center for Solid and Hazardous Waste Management, Gainesville, U.S.A..

Solo-Gabriele, H., M. Kormienko, K. Gary, T. Townsend, K. Stook & T. Tolaymat (2000). *Alternative chemicals and improved disposal-end management practices for CCA-treated wood*. Report #00-03, Florida Center for Solid and Hazardous Waste Management, Gainesville, U.S.A..

Solo-Gabriele, H., N. Hosein, G. Jacobi, T. Townsend, J. Jambeck, D. Hahn, T. Moskal & K. Iida (2001). *On-line sorting technologies for CCA-treated wood*. Sarasota County for Submission to the FDEP Innovative Recycling Grants Program, Sarasota, U.S.A..

Solo-Gabriele, H. M., T. G. Townsend, B. Messick & V. Calitu (2002). Characteristics of chromated copper arsenate-treated wood ash. *Journal of Hazardous Materials,* **B89**: pp. 213–232.

Solo-Gabriele, H., D. M. Sakura-Lemessy, T. Townsend, B. Dubey & J. Jambeck (2003). *Quantities of arsenic within the state of Florida*. Report #03-05, Florida Center for Solid and Hazardous Waste Management, Gainesville, U.S.A..

Spanos, K., P. Koukos & G. Skodras (2001). Waste management options in Greece. *Waste Management Options in Europe: Information Compiled in COST Action E9.* A. Merl (Ed), COST E9 Working Group 3 “End of life: recycling, disposal and energy production”, Brussels: pp. 43-49.

Speckels, L. & S. Springer (2001). Waste management options in Germany. *Waste Management Options in Europe: Information Compiled in COST Action E9*. A. Merl (Ed), COST E9 Working Group 3 “End of life: recycling, disposal and energy production”, Brussels: pp. 35-42.

Sridharan, R. (1995). The capacitated plant location problem. *European Journal of Operational Research,* **87**: pp. 203-213.

Stephan, I., H. Leithoff & R. D. Peek (1996). Microbial conversion of wood treated with salt preservatives. *Material u. Organismen,* **30**: pp. 179-200.

Stilwell, D., M. Toner & B. Sawhney (2003). Dislodgeable copper, chromium and arsenic from CCA-treated wood surfaces. *The Science of the Total Environment,* **312**: pp. 123–131.

Suttie, E. (2004). Wood waste management: UK update. *Final Conference COST Action E22 "Environmental Optimisation of Wood Protection"*, *Lisboa, Portugal, 22nd-23rd March.* URL: http://www.bfafh.de/cost22.htm. Accessed 3rd April 2004.

Suzuki, A. & A. Okabe (1995). Using Voronoi diagrams. *Facility Location: A Survey of Applications and Methods*, Z. Drezner (Ed), Springer Verlag, New York: pp. 103-118.

Suzuki, K. (1998). Tendency of the preservative use for impregnation industries in Japan. *Proceedings of the 4th International Symposium on Wood Preservation, Cannes - Mandelieu, France, 2nd-3rd February.* IRG Secretariat, Stockholm, Sweden: pp. 43-53.

Tãhkãlã, T. (2002). Waste and recycling. *Workshop 4 - COST E22*, *Tuusula, Finland, 3rd June*. URL: http://www.bfafh.de/inst4/43/download.htm. Accessed 1st November 2003.

Tame, N. W., B. Z. Dlugogorski & E. M. Kennedy (2003). Increased PCDD/F formation in the bottom ash from fires of CCA-treated wood. *Chemosphere,* **50**: pp. 1261–1263.

Tchobanoglous, G., H. Theisen & S. A. Virgil (1993). *Integrated solid waste management: engineering principles and management issues*. McGraw-Hill, Singapore.

Teitz, M. B. & P. Bart (1968). Heuristic methods for estimating the generalized vertex median of a weighted graph. *Operations Research,* **16**: pp. 955-961.

Townsend, T., K. Stook, T. Tolaymat, J. K. Song, H. Solo-Gabriele, N. Hosein & B. Khan (2001). *Metals concentrations in soils below decks made of CCA-treated wood*. Excerpts from Report #00-12, Florida Center for Solid and Hazardous Waste Management, Gainesville, U.S.A..

Townsend, T., K. Stook, M. Ward & H. Solo-Gabriele (2003a). *Leaching and toxicity of CCA-treated and alternative-treated wood products*. Report #02-4, Florida Center for Solid and Hazardous Waste Management, Gainesville, U.S.A..

Townsend, T. G., H. Solo-Gabriele, T. Tolaymat & K. Stook (2003b). Impact of chromated copper arsenate (CCA) in wood mulch. *The Science of the Total Environment,* **309**: pp. 173-185.

USEPA (2002). *Whitman announces transition from consumer use of treated wood containing arsenic*. United States Environmental Protection Agency. URL: http://www.epa.gov/epahome/headline_021202.htm. Accessed 30th October 2003.

Vesanto, J., J. Himberg, E. Alhoniemi & J. Parhankangas (2000). *SOM Toolbox for Matlab 5*. Helsinki University of Technology, Helsinki, Finland.

Villumsen, A. (2003). *Elektrodialytisk Rensning af CCA Impraegneret Affaldstrae*. LIFE Project Number LIFE ENV/000/369, Lyngby, Denmark.

WBG (1998). *Pollution prevention and abatement handbook: wood preserving industry*. Environment Department, World Bank Group, Washington D.C.: pp. 433-435.

Weis, J. S. & P. Weis (2002). Contamination of saltmarsh sediments and biota by CCA treated wood walkways. *Marine Pollution Bulletin,* **44**: pp. 504–510.

Zhao, P. & R. Batta (2000). An aggregation approach to solving the network p-median problem with link demands. *Networks,* **36** (4): pp. 233-241.

Zhou, J. (2000). Uncapacitated facility layout problem with stochastic demands. *Proceedings of the Sixth National Conference of Operations Research Society of China. Changsha, China, October 10th-15th*, X. Zhang, J. Wang, B. Liu & D. Liu. (Eds), Global-Link Publishing Company, Hong-Kong: pp. 904-911.

Appendix I.

Questionnaire sent to the preservation industries

QUESTIONNAIRE

Part 1 – Company Information

1.1 Company name: ____
1.2 Headquarters address: ____
1.3 Plant Address: ____
1.4. Contact person: ________________________ Post: _______________
1.5 Year of the beginning of operation: ________________
1.6 Would you like to receive a copy of the results from this survey? Yes ☐ No ☐

Part 2 – Process Information

2.1 Complete the information of the preservative used:

Product		**Amount used (annual mean in ton)**
Creosote	☐	________________
CCA (Type 1)[1]	☐	________________
CCA (Type 2)[2]	☐	________________
CCB	☐	________________
Other:________________	☐	________________

2.2 Characterization of the equipment used

Number of cylinders	1	2	3	4	4	5	7	8
Diameter (m)								
Length (m)								

Part 3 – Amounts of preserved wood produced

3.1. Amount produced in 2003 (m^3) __________
3.2. Annual mean amount (m^3) __________
3.3. Amount exported in 2003 (m^3) __________
3.4 Annual mean amount exported (m^3) __________

3.5 Which wood products are produced by your company?

Product		**Amount produced (annual mean in m^3)**
Telephone poles	☐	____________________
Railway sleepers	☐	____________________
Fence posts	☐	____________________
Highway posts	☐	____________________
Agricultural posts	☐	____________________
Sawn Timber	☐	____________________
Garden furniture and playgrounds	☐	____________________
Other:________________	☐	____________________

3.6 Which percentage of total production is rejected? ________________
Which is the final destination of these wastes?________________________
____

[1] 32.6% m/m de $CuSO_4.5H_2O$; 41.0 % m/m de $K_2Cr_2O_7$ ou $Na_2Cr_2O_7.2H_2O$ e 26.4% m/m de $As_2O_5.2H_2O$

[2] 35.0% m/m de $CuSO_4.5H_2O$; 45.0 % m/m de $K_2Cr_2O_7$ ou $Na_2Cr_2O_7.2H_2O$ e 20.0% m/m de $As_2O_5.2H_2O$

Appendix II.
MATLAB code

```
% TEST 1

uiimport

% Generated data patterns
W=weight_data(pop(:,1:2),pop(:,3));
W=[W rand(99146,1)];
W=sortrows(W,3);
W=W(:,1:2);

% P1
pa0=som_randinit(W,'msize',[5 1],'rect','sheet');
pa1=som_seqtrain(pa0,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',3,'trainlen_type','epochs','neigh','bubble');
pa2=som_seqtrain(pa1,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',6,'trainlen_type','epochs','neigh','bubble');
[qep1,tep1]=som_quality(pa2,W)

% P2
pb0=som_randinit(W,'msize',[5 1],'rect','sheet');
pb1=som_seqtrain(pb0,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',30, 'trainlen_type','epochs','neigh','bubble');
pb2=som_seqtrain(pb1,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',60,'trainlen_type','epochs','neigh','bubble');
[qep2,tep2]=som_quality(pb2,W)

% P3
pc0=som_randinit(W,'msize',[5 1],'rect','sheet');
pc1=som_seqtrain(pc0,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',300,'trainlen_type','epochs','neigh','bubble');
pc2=som_seqtrain(pc1,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',600,'trainlen_type','epochs','neigh','bubble');
[qep3,tep3]=som_quality(pc2,W)

% P4
pd0=som_randinit(W,'msize',[5 2],'rect','sheet');
pd1=som_seqtrain(pd0,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',3,'trainlen_type','epochs','neigh','bubble');
pd2=som_seqtrain(pd1,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',6,...
  'trainlen_type','epochs','neigh','bubble');
[qep4,tep4]=som_quality(pd2,W)

% P5
pe0=som_randinit(W,'msize',[5 2],'rect','sheet');
pe1=som_seqtrain(pe0,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',3,'trainlen_type','epochs','neigh','bubble');
pe2=som_seqtrain(pe1,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',6,'trainlen_type','epochs','neigh','bubble');
[qep5,tep5]=som_quality(pe2,W)

% P6
pf0=som_randinit(W,'msize',[5 2],'rect','sheet');
pf1=som_seqtrain(pf0,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',300, 'trainlen_type','epochs','neigh','bubble');
pf2=som_seqtrain(pf1,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',600,trainlen_type','epochs','neigh','bubble');
[qep6,tep6]=som_quality(pf2,W)

% P7
pg0=som_randinit(W,'msize',[5 3],'rect','sheet');
pg1=som_seqtrain(pg0,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',3,'trainlen_type','epochs','neigh','bubble');
pg2=som_seqtrain(pg1,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',6,'trainlen_type','epochs','neigh','bubble');
[qep4,tep4]=som_quality(pg2,W)
```

```
% P8
ph0=som_randinit(W,'msize',[5 3],'rect','sheet');
ph1=som_seqtrain(ph0,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',3,'trainlen_type','epochs','neigh','bubble');
ph2=som_seqtrain(ph1,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',6,'trainlen_type','epochs','neigh','bubble');
[qep5,tep5]=som_quality(ph2,W)

% P9
pi0=som_randinit(W,'msize',[5 3],'rect','sheet');
pi1=som_seqtrain(pi0,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',300,'trainlen_type','epochs','neigh','bubble');
pi2=som_seqtrain(pi1,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',600,'trainlen_type','epochs','neigh','bubble');
[qep6,tep6]=som_quality(pi2,W)

% Generated data patterns
W1=weight_data(data(:,1:2),data(:,3));
W1=[W1 rand(155044,1)];
W1=sortrows(W1,3);
W1=W1(:,1:2);

% CA1
ca10=som_randinit(W1,'msize',[5 1],'rect','sheet');
ca11=som_seqtrain(ca10,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',3,...
'trainlen_type','epochs','neigh','bubble');
ca12=som_seqtrain(ca11,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',6,'trainlen_type','epochs','neigh','bubble');
[qeca1,teca1]=som_quality(ca12,W1)

% CA2
ca20=som_randinit(W1,'msize',[5 1],'rect','sheet');
ca21=som_seqtrain(ca20,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',30,'trainlen_type','epochs','neigh','bubble');
ca22=som_seqtrain(ca21,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',60,'trainlen_type','epochs','neigh','bubble');
[qeca2,teca2]=som_quality(ca22,W1)

% CA3
ca30=som_randinit(W1,'msize',[5 1],'rect','sheet');
ca31=som_seqtrain(ca30,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',300,'trainlen_type','epochs','neigh','bubble');
ca32=som_seqtrain(ca31,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',600,'trainlen_type','epochs','neigh','bubble');
[qeca3,teca3]=som_quality(ca32,W1)

% CA4
ca40=som_randinit(W1,'msize',[5 2],'rect','sheet');
ca41=som_seqtrain(ca10,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',3,'trainlen_type','epochs','neigh','bubble');
ca42=som_seqtrain(ca11,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',6,'trainlen_type','epochs','neigh','bubble');
[qeca1,teca1]=som_quality(ca42,W1)

% CA5
ca50=som_randinit(W1,'msize',[5 2],'rect','sheet');
ca51=som_seqtrain(ca50,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',30,'trainlen_type','epochs','neigh','bubble');
ca52=som_seqtrain(ca51,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',60,'trainlen_type','epochs','neigh','bubble');
[qeca5,teca5]=som_quality(ca52,W1)

% CA6
ca60=som_randinit(W1,'msize',[5 2],'rect','sheet');
ca61=som_seqtrain(ca60,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',300,'trainlen_type','epochs','neigh','bubble');
ca62=som_seqtrain(ca61,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',600,'trainlen_type','epochs','neigh','bubble');
```

```
[qeca6,teca6]=som_quality(ca62,W1)

% CA7
ca70=som_randinit(W1,'msize',[5 3],'rect','sheet');
ca71=som_seqtrain(ca70,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',3,'trainlen_type','epochs','neigh','bubble');
ca72=som_seqtrain(ca71,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',6,'trainlen_type','epochs','neigh','bubble');
[qeca7,teca7]=som_quality(ca72,W1)

% CA8
ca80=som_randinit(W1,'msize',[5 3],'rect','sheet');
ca81=som_seqtrain(ca80,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',30,'trainlen_type','epochs','neigh','bubble');
ca82=som_seqtrain(ca81,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',60,'trainlen_type','epochs','neigh','bubble');
[qeca8,teca8]=som_quality(ca82,W1)

% CA9
ca90=som_randinit(W1,'msize',[5 3],'rect','sheet');
ca91=som_seqtrain(ca90,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',300,'trainlen_type','epochs','neigh','bubble');
ca92=som_seqtrain(ca91,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',600,'trainlen_type','epochs','neigh','bubble');
[qeca9,teca9]=som_quality(ca92,W1)

% Generated data patterns
W2=weight_data(data(:,1:2),data(:,4));
W2=[W2 rand(267876,1)];
W2=sortrows(W2,3);
W2=W2(:,1:2);

% CB1
cb10=som_randinit(W2,'msize',[5 1],'rect','sheet');
cb11=som_seqtrain(cb10,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',3,'trainlen_type','epochs','neigh','bubble');
cb12=som_seqtrain(cb11,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',6,'trainlen_type','epochs','neigh','bubble');
[qecb1,tecb1]=som_quality(cb12,W2)

% CB2
cb20=som_randinit(W2,'msize',[5 1],'rect','sheet');
cb21=som_seqtrain(cb20,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',30,'trainlen_type','epochs','neigh','bubble');
cb22=som_seqtrain(cb21,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',60,'trainlen_type','epochs','neigh','bubble');
[qecb2,tecb2]=som_quality(cb22,W2)

% CB3
cb30=som_randinit(W2,'msize',[5 1],'rect','sheet');
cb31=som_seqtrain(cb30,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',300,'trainlen_type','epochs','neigh','bubble');
cb32=som_seqtrain(cb31,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',600,'trainlen_type','epochs','neigh','bubble');
[qecb3,tecb3]=som_quality(cb32,W2)

% CB4
cb40=som_randinit(W2,'msize',[5 2],'rect','sheet');
cb41=som_seqtrain(cb40,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',3,'trainlen_type','epochs','neigh','bubble');
cb42=som_seqtrain(cb41,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',6,'trainlen_type','epochs','neigh','bubble');
[qecb4,tecb4]=som_quality(cb42,W2)

% CB5
cb50=som_randinit(W2,'msize',[5 2],'rect','sheet');
cb51=som_seqtrain(cb50,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',30,'trainlen_type','epochs','neigh','bubble');
```

```
cb52=som_seqtrain(cb51,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',60,'trainlen_type','epochs','neigh','bubble');
[qecb5,tecb5]=som_quality(cb52,W2)

% CB6
cb60=som_randinit(W2,'msize',[5 2],'rect','sheet');
cb61=som_seqtrain(cb60,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',300,'trainlen_type','epochs','neigh','bubble');
cb62=som_seqtrain(cb61,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',600,'trainlen_type','epochs','neigh','bubble');
[qecb6,tecb6]=som_quality(cb62,W2)

% CB7
cb70=som_randinit(W2,'msize',[5 3],'rect','sheet');
cb71=som_seqtrain(cb70,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',3,'trainlen_type','epochs','neigh','bubble');
cb72=som_seqtrain(cb71,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',6,'trainlen_type','epochs','neigh','bubble');
[qecb7,tecb7]=som_quality(cb72,W2)

% CB8
cb80=som_randinit(W2,'msize',[5 3],'rect','sheet');
cb81=som_seqtrain(cb80,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',30,'trainlen_type','epochs','neigh','bubble');
cb82=som_seqtrain(cb81,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',60,'trainlen_type','epochs','neigh','bubble');
[qecb8,tecb8]=som_quality(cb82,W2)

% CB9
cb90=som_randinit(W2,'msize',[5 3],'rect',sheet');
cb91=som_seqtrain(cb90,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',300,'trainlen_type','epochs','neigh','bubble');
cb92=som_seqtrain(cb91,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',600,'trainlen_type','epochs','neigh','bubble');
[qecb9,tecb9]=som_quality(cb92,W2)

% Generated data patterns
W3=weight_data(data(:,1:2),data(:,5));
W3=[W3 rand(355110,1)];
W3=sortrows(W3,3);
W3=W3(:,1:2);

% CC1
cc10=som_randinit(W3,'msize',[5 1],'rect','sheet');
cc11=som_seqtrain(cc10,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',3,'trainlen_type','epochs','neigh','bubble');
cc12=som_seqtrain(cc11,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',6,'trainlen_type','epochs','neigh','bubble');
[qecc1,tecc1]=som_quality(cc12,W3)

% CC2
cc20=som_randinit(W3,'msize',[5 1],'rect','sheet');
cc21=som_seqtrain(cc20,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',30,'trainlen_type','epochs','neigh','bubble');
cc22=som_seqtrain(cc21,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',60,'trainlen_type','epochs','neigh','bubble');
[qecc2,tecc2]=som_quality(cc22,W3)

% CC3
cc30=som_randinit(W3,'msize',[5 1],'rect','sheet');
cc31=som_seqtrain(cc30,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',300,'trainlen_type','epochs','neigh','bubble');
cc32=som_seqtrain(cc31,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',600,'trainlen_type','epochs','neigh','bubble');
[qecc3,tecc3]=som_quality(cc32,W3)

% CC4
cc40=som_randinit(W3,'msize',[5 2],'rect','sheet');
```

```
cc41=som_seqtrain(cc40,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',3,'trainlen_type','epochs','neigh','bubble');
cc42=som_seqtrain(cc41,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',6,'trainlen_type','epochs','neigh','bubble');
[qecc4,tecc4]=som_quality(cc42,W3)

% CC5
cc50=som_randinit(W3,'msize',[5 2],'rect','sheet');
cc51=som_seqtrain(cc50,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',30,'trainlen_type','epochs','neigh','bubble');
cc52=som_seqtrain(cc51,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',60,'trainlen_type','epochs','neigh','bubble');
[qecc5,tecc5]=som_quality(cc52,W3)

% CC6
cc60=som_randinit(W3,'msize',[5 2],'rect','sheet');
cc61=som_seqtrain(cc60,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',300,'trainlen_type','epochs','neigh','bubble');
cc62=som_seqtrain(cc61,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',600,'trainlen_type','epochs','neigh','bubble');
[qecc6,tecc6]=som_quality(cc62,W3)

% CC7
cc70=som_randinit(W3,'msize',[5 3],'rect','sheet');
cc71=som_seqtrain(cc70,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',3,'trainlen_type','epochs','neigh','bubble');
cc72=som_seqtrain(cc71,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',6,'trainlen_type','epochs','neigh','bubble');
[qecc7,tecc7]=som_quality(cc72,W3)

% CC8
cc80=som_randinit(W3,'msize',[5 3],'rect','sheet');
cc81=som_seqtrain(cc80,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',30,'trainlen_type','epochs','neigh','bubble');
cc82=som_seqtrain(cc81,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',60,'trainlen_type','epochs','neigh','bubble');
[qecc8,tecc8]=som_quality(cc82,W3)

% CC9
cc90=som_randinit(W3,'msize',[5 3],'rect','sheet');
cc91=som_seqtrain(cc90,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',300,'trainlen_type','epochs','neigh','bubble');
cc92=som_seqtrain(cc91,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05,
'trainlen',600,'trainlen_type','epochs','neigh','bubble');
[qecc9,tecc9]=som_quality(cc92,W3)

% TEST 2

uiimport

% Generated data patterns
W=weight_data(pop(:,1:2),pop(:,3));
W=[W rand(99146,1)];
W=sortrows(W,3);
W=W(:,1:2);

% P1
tic
p1010=som_randinit(W,'msize',[5 1],'rect','sheet');
p1011=som_seqtrain(p1010,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1012=som_seqtrain(p1011,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep01,tep01]=som_quality(p1012,W)
clear functions
tic
p1020=som_randinit(W,'msize',[5 1],'rect','sheet');
```

```
p1021=som_seqtrain(p1020,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1022=som_seqtrain(p1021,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep02,tep02]=som_quality(p1022,W)
clear functions

tic
p1030=som_randinit(W,'msize',[5 1],'rect','sheet');
p1031=som_seqtrain(p1030,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1032=som_seqtrain(p1031,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
'trainlen',10,'trainlen_type','epochs','neigh','bubble');
[qep03,tep03]=som_quality(p1032,W)
toc
clear functions

tic
p1040=som_randinit(W,'msize',[5 1],'rect','sheet');
p1041=som_seqtrain(p1040,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1042=som_seqtrain(p1041,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep04,tep04]=som_quality(p1042,W)
clear functions

tic
p1050=som_randinit(W,'msize',[5 1],'rect','sheet');
p1051=som_seqtrain(p1050,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1052=som_seqtrain(p1051,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep05,tep05]=som_quality(p1052,W)
clear functions

tic
p1060=som_randinit(W,'msize',[5 1],'rect','sheet');
p1061=som_seqtrain(p1060,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1062=som_seqtrain(p1061,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep06,tep06]=som_quality(p1062,W)
clear functions

tic
p1070=som_randinit(W,'msize',[5 1],'rect','sheet');
p1071=som_seqtrain(p1070,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1072=som_seqtrain(p1071,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep07,tep07]=som_quality(p1072,W)
clear functions
tic
p1080=som_randinit(W,'msize',[5 1],'rect','sheet');
p1081=som_seqtrain(p1080,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1082=som_seqtrain(p1081,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep08,tep08]=som_quality(p1082,W)
clear functions
tic
p1090=som_randinit(W,'msize',[5 1],'rect','sheet');
```

```
p1091=som_seqtrain(p1090,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1092=som_seqtrain(p1091,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep09,tep09]=som_quality(p1092,W)
clear functions

tic
p1100=som_randinit(W,'msize',[5 1],'rect','sheet');
p1101=som_seqtrain(p1100,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1102=som_seqtrain(p1101,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep10,tep10]=som_quality(p1102,W)
clear functions

tic
p1110=som_randinit(W,'msize',[5 1],'rect','sheet');
p1111=som_seqtrain(p1110,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1112=som_seqtrain(p1111,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep11,tep11]=som_quality(p1112,W)
clear functions

tic
p1120=som_randinit(W,'msize',[5 1],'rect','sheet');
p1121=som_seqtrain(p1120,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1122=som_seqtrain(p1121,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep12,tep12]=som_quality(p1122,W)
clear functions

tic
p1130=som_randinit(W,'msize',[5 1],'rect','sheet');
p1131=som_seqtrain(p1130,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1132=som_seqtrain(p1131,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep13,tep13]=som_quality(p1132,W)
clear functions

tic
p1140=som_randinit(W,'msize',[5 1],'rect','sheet');
p1141=som_seqtrain(p1140,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1142=som_seqtrain(p1141,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep14,tep14]=som_quality(p1142,W)
clear functions

tic
p1150=som_randinit(W,'msize',[5 1],'rect','sheet');
p1151=som_seqtrain(p1150,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1152=som_seqtrain(p1151,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep15,tep15]=som_quality(p1152,W)
clear functions

tic
```

```
p1160=som_randinit(W,'msize',[5 1],'rect','sheet');
p1161=som_seqtrain(p1160,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1162=som_seqtrain(p1161,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep16,tep16]=som_quality(p1162,W)
clear functions

tic
p1170=som_randinit(W,'msize',[5 1],'rect','sheet');
p1171=som_seqtrain(p1170,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1172=som_seqtrain(p1171,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep17,tep17]=som_quality(p1172,W)
clear functions

tic
p1180=som_randinit(W,'msize',[5 1],'rect','sheet');
p1181=som_seqtrain(p1180,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1182=som_seqtrain(p1181,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep18,tep18]=som_quality(p1182,W)
clear functions

tic
p1190=som_randinit(W,'msize',[5 1],'rect','sheet');
p1191=som_seqtrain(p1190,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1192=som_seqtrain(p1191,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep19,tep19]=som_quality(p1192,W)
clear functions

tic
p1200=som_randinit(W,'msize',[5 1],'rect','sheet');
p1201=som_seqtrain(p1200,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1202=som_seqtrain(p1201,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep20,tep20]=som_quality(p1202,W)
clear functions

tic
p1210=som_randinit(W,'msize',[5 1],'rect','sheet');
p1211=som_seqtrain(p1210,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1212=som_seqtrain(p1211,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep21,tep21]=som_quality(p1212,W)
clear functions

tic
p1220=som_randinit(W,'msize',[5 1],'rect','sheet');
p1221=som_seqtrain(p1220,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1222=som_seqtrain(p1221,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep22,tep22]=som_quality(p1222,W)
clear functions
```

```
tic
p1230=som_randinit(W,'msize',[5 1],'rect','sheet');
p1231=som_seqtrain(p1230,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1232=som_seqtrain(p1231,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep23,tep23]=som_quality(p1232,W)
clear functions

tic
p1240=som_randinit(W,'msize',[5 1],'rect','sheet');
p1241=som_seqtrain(p1240,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1242=som_seqtrain(p1241,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep24,tep24]=som_quality(p1242,W)
clear functions

tic
p1250=som_randinit(W,'msize',[5 1],'rect','sheet');
p1251=som_seqtrain(p1250,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1252=som_seqtrain(p1251,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep25,tep25]=som_quality(p1252,W)
clear functions

tic
p1260=som_randinit(W,'msize',[5 1],'rect','sheet');
p1261=som_seqtrain(p1260,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1262=som_seqtrain(p1261,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep26,tep26]=som_quality(p1262,W)
clear functions

tic
p1270=som_randinit(W,'msize',[5 1],'rect','sheet');
p1271=som_seqtrain(p1270,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1272=som_seqtrain(p1271,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep27,tep27]=som_quality(p1272,W)
clear functions

tic
p1280=som_randinit(W,'msize',[5 1],'rect','sheet');
p1281=som_seqtrain(p1280,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1282=som_seqtrain(p1281,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep28,tep28]=som_quality(p1282,W)
clear functions

tic
p1290=som_randinit(W,'msize',[5 1],'rect','sheet');
p1291=som_seqtrain(p1290,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1292=som_seqtrain(p1291,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep29,tep29]=som_quality(p1292,W)
clear functions
```

```
tic
p1300=som_randinit(W,'msize',[5 1],'rect','sheet');
p1301=som_seqtrain(p1300,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p1302=som_seqtrain(p1301,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep30,tep30]=som_quality(p1302,W)
clear functions

% P2

tic
p2010=som_randinit(W,'msize',[10 1],'rect','sheet');
p2011=som_seqtrain(p2010,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2012=som_seqtrain(p2011,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep01,tep01]=som_quality(p2012,W)
clear functions

tic
p2020=som_randinit(W,'msize',[10 1],'rect','sheet');
p2021=som_seqtrain(p2020,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2022=som_seqtrain(p2021,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep02,tep02]=som_quality(p2022,W)
clear functions

tic
p2030=som_randinit(W,'msize',[10 1],'rect','sheet');
p2031=som_seqtrain(p2030,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2032=som_seqtrain(p2031,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep03,tep03]=som_quality(p2032,W)
clear functions

tic
p2040=som_randinit(W,'msize',[10 1],'rect','sheet');
p2041=som_seqtrain(p2040,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2042=som_seqtrain(p2041,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep04,tep04]=som_quality(p2042,W)
clear functions

tic
p2050=som_randinit(W,'msize',[10 1],'rect','sheet');
p2051=som_seqtrain(p2050,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2052=som_seqtrain(p2051,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep05,tep05]=som_quality(p2052,W)
clear functions

tic
p2060=som_randinit(W,'msize',[10 1],'rect','sheet');
p2061=som_seqtrain(p2060,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2062=som_seqtrain(p2061,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
```

```
[qep06,tep06]=som_quality(p2062,W)
clear functions

tic
p2070=som_randinit(W,'msize',[10 1],'rect','sheet');
p2071=som_seqtrain(p2070,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2072=som_seqtrain(p2071,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep07,tep07]=som_quality(p2072,W)
clear functions

tic
p2080=som_randinit(W,'msize',[10 1],'rect','sheet');
p2081=som_seqtrain(p2080,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2082=som_seqtrain(p2081,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep08,tep08]=som_quality(p2082,W)
clear functions

tic
p2090=som_randinit(W,'msize',[10 1],'rect','sheet');
p2091=som_seqtrain(p2090,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2092=som_seqtrain(p2091,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep09,tep09]=som_quality(p2092,W)
clear functions

tic
p2100=som_randinit(W,'msize',[10 1],'rect','sheet');
p2101=som_seqtrain(p2100,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2102=som_seqtrain(p2101,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep10,tep10]=som_quality(p2102,W)
clear functions

tic
p2110=som_randinit(W,'msize',[10 1],'rect','sheet');
p2111=som_seqtrain(p2110,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2112=som_seqtrain(p2111,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep11,tep11]=som_quality(p2112,W)
clear functions

tic
p2120=som_randinit(W,'msize',[10 1],'rect','sheet');
p2121=som_seqtrain(p2120,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2122=som_seqtrain(p2121,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep12,tep12]=som_quality(p2122,W)
clear functions

tic
p2130=som_randinit(W,'msize',[10 1],'rect','sheet');
p2131=som_seqtrain(p2130,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2132=som_seqtrain(p2131,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
```

```
toc
[qep13,tep13]=som_quality(p2132,W)
clear functions

tic
p2140=som_randinit(W,'msize',[10 1],'rect','sheet');
p2141=som_seqtrain(p2140,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2142=som_seqtrain(p2141,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep14,tep14]=som_quality(p2142,W)
clear functions

tic
p2150=som_randinit(W,'msize',[10 1],'rect','sheet');
p2151=som_seqtrain(p2150,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2152=som_seqtrain(p2151,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep15,tep15]=som_quality(p2152,W)
clear functions

tic
p2160=som_randinit(W,'msize',[10 1],'rect','sheet');
p2161=som_seqtrain(p2160,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2162=som_seqtrain(p2161,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep16,tep16]=som_quality(p2162,W)
clear functions

tic
p2170=som_randinit(W,'msize',[10 1],'rect','sheet');
p2171=som_seqtrain(p2170,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2172=som_seqtrain(p2171,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep17,tep17]=som_quality(p2172,W)
clear functions

tic
p2180=som_randinit(W,'msize',[10 1],'rect','sheet');
p2181=som_seqtrain(p2180,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2182=som_seqtrain(p2181,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep18,tep18]=som_quality(p2182,W)
clear functions

tic
p2190=som_randinit(W,'msize',[10 1],'rect','sheet');
p2191=som_seqtrain(p2190,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2192=som_seqtrain(p2191,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep19,tep19]=som_quality(p2192,W)
clear functions

tic
p2200=som_randinit(W,'msize',[10 1],'rect','sheet');
p2201=som_seqtrain(p2200,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
```

```
p2202=som_seqtrain(p2201,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep20,tep20]=som_quality(p2202,W)
clear functions

tic
p2210=som_randinit(W,'msize',[10 1],'rect','sheet');
p2211=som_seqtrain(p2210,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2212=som_seqtrain(p2211,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep21,tep21]=som_quality(p2212,W)
clear functions

tic
p2220=som_randinit(W,'msize',[10 1],'rect','sheet');
p2221=som_seqtrain(p2220,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2222=som_seqtrain(p2221,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep22,tep22]=som_quality(p2222,W)
clear functions

tic
p2230=som_randinit(W,'msize',[10 1],'rect','sheet');
p2231=som_seqtrain(p2230,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2232=som_seqtrain(p2231,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep23,tep23]=som_quality(p2232,W)
clear functions

tic
p2240=som_randinit(W,'msize',[10 1],'rect','sheet');
p2241=som_seqtrain(p2240,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2242=som_seqtrain(p2241,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep24,tep24]=som_quality(p2242,W)
clear functions

tic
p2250=som_randinit(W,'msize',[10 1],'rect','sheet');
p2251=som_seqtrain(p2250,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2252=som_seqtrain(p2251,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep25,tep25]=som_quality(p2252,W)
clear functions

tic
p2260=som_randinit(W,'msize',[10 1],'rect','sheet');
p2261=som_seqtrain(p2260,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2262=som_seqtrain(p2261,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep26,tep26]=som_quality(p2262,W)
clear functions

tic
p2270=som_randinit(W,'msize',[10 1],'rect','sheet');
```

```
p2271=som_seqtrain(p2270,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2272=som_seqtrain(p2271,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep27,tep27]=som_quality(p2272,W)
clear functions

tic
p2280=som_randinit(W,'msize',[10 1],'rect','sheet');
p2281=som_seqtrain(p2280,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2282=som_seqtrain(p2281,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep28,tep28]=som_quality(p2282,W)
clear functions

tic
p2290=som_randinit(W,'msize',[10 1],'rect','sheet');
p2291=som_seqtrain(p2290,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2292=som_seqtrain(p2291,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep29,tep29]=som_quality(p2292,W)
clear functions

tic
p2300=som_randinit(W,'msize',[10 1],'rect','sheet');
p2301=som_seqtrain(p2300,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p2302=som_seqtrain(p2301,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep30,tep30]=som_quality(p2302,W)
clear functions

% P3

tic
p3010=som_randinit(W,'msize',[15 1],'rect','sheet');
p3011=som_seqtrain(p3010,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3012=som_seqtrain(p3011,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep01,tep01]=som_quality(p3012,W)
clear functions

tic
p3020=som_randinit(W,'msize',[15 1],'rect','sheet');
p3021=som_seqtrain(p3020,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3022=som_seqtrain(p3021,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep02,tep02]=som_quality(p3022,W)
clear functions

tic
p3030=som_randinit(W,'msize',[15 1],'rect','sheet');
p3031=som_seqtrain(p3030,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3032=som_seqtrain(p3031,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep03,tep03]=som_quality(p3032,W)
clear functions
```

```
tic
p3040=som_randinit(W,'msize',[15 1],'rect','sheet');
p3041=som_seqtrain(p3040,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3042=som_seqtrain(p3041,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep04,tep04]=som_quality(p3042,W)
clear functions

tic
p3050=som_randinit(W,'msize',[15 1],'rect','sheet');
p3051=som_seqtrain(p3050,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3052=som_seqtrain(p3051,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep05,tep05]=som_quality(p3052,W)
clear functions

tic
p3060=som_randinit(W,'msize',[15 1],'rect','sheet');
p3061=som_seqtrain(p3060,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3062=som_seqtrain(p3061,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep06,tep06]=som_quality(p3062,W)
clear functions

tic
p3070=som_randinit(W,'msize',[15 1],'rect','sheet');
p3071=som_seqtrain(p3070,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3072=som_seqtrain(p3071,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep07,tep07]=som_quality(p3072,W)
clear functions

tic
p3080=som_randinit(W,'msize',[15 1],'rect','sheet');
p3081=som_seqtrain(p3080,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3082=som_seqtrain(p3081,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep08,tep08]=som_quality(p3082,W)
clear functions

tic
p3090=som_randinit(W,'msize',[13 1],'rect','sheet');
p3091=som_seqtrain(p3090,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3092=som_seqtrain(p3091,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep09,tep09]=som_quality(p3092,W)
clear functions

tic
p3100=som_randinit(W,'msize',[15 1],'rect','sheet');
p3101=som_seqtrain(p3100,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3102=som_seqtrain(p3101,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep10,tep10]=som_quality(p3102,W)
clear functions
```

```
tic
p3110=som_randinit(W,'msize',[15 1],'rect','sheet');
p3111=som_seqtrain(p3110,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3112=som_seqtrain(p3111,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep11,tep11]=som_quality(p3112,W)
clear functions

tic
p3120=som_randinit(W,'msize',[15 1],'rect','sheet');
p3121=som_seqtrain(p3120,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3122=som_seqtrain(p3121,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep12,tep12]=som_quality(p3122,W)
clear functions

tic
p3130=som_randinit(W,'msize',[15 1],'rect','sheet');
p3131=som_seqtrain(p3130,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3132=som_seqtrain(p3131,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep13,tep13]=som_quality(p3132,W)
clear functions

tic
p3140=som_randinit(W,'msize',[15 1],'rect','sheet');
p3141=som_seqtrain(p3140,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3142=som_seqtrain(p3141,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep14,tep14]=som_quality(p3142,W)
clear functions

tic
p3150=som_randinit(W,'msize',[15 1],'rect','sheet');
p3151=som_seqtrain(p3150,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3152=som_seqtrain(p3151,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep15,tep15]=som_quality(p3152,W)
clear functions

tic
p3160=som_randinit(W,'msize',[15 1],'rect','sheet');
p3161=som_seqtrain(p3160,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3162=som_seqtrain(p3161,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep16,tep16]=som_quality(p3162,W)
clear functions

tic
p3170=som_randinit(W,'msize',[15 1],'rect','sheet');
p3171=som_seqtrain(p3170,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3172=som_seqtrain(p3171,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep17,tep17]=som_quality(p3172,W)
clear functions
```

```
tic
p3180=som_randinit(W,'msize',[15 1],'rect','sheet');
p3181=som_seqtrain(p3180,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3182=som_seqtrain(p3181,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep18,tep18]=som_quality(p3182,W)
clear functions

tic
p3190=som_randinit(W,'msize',[15 1],'rect','sheet');
p3191=som_seqtrain(p3190,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3192=som_seqtrain(p3191,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep19,tep19]=som_quality(p3192,W)
clear functions

tic
p3200=som_randinit(W,'msize',[15 1],'rect','sheet');
p3201=som_seqtrain(p3200,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3202=som_seqtrain(p3201,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep20,tep20]=som_quality(p3202,W)
clear functions

tic
p3210=som_randinit(W,'msize',[15 1],'rect','sheet');
p3211=som_seqtrain(p3210,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3212=som_seqtrain(p3211,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep21,tep21]=som_quality(p3212,W)
clear functions

tic
p3220=som_randinit(W,'msize',[15 1],'rect','sheet');
p3221=som_seqtrain(p3220,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3222=som_seqtrain(p3221,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep22,tep22]=som_quality(p3222,W)
clear functions

tic
p3230=som_randinit(W,'msize',[15 1],'rect','sheet');
p3231=som_seqtrain(p3230,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3232=som_seqtrain(p3231,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep23,tep23]=som_quality(p3232,W)
clear functions

tic
p3240=som_randinit(W,'msize',[15 1],'rect','sheet');
p3241=som_seqtrain(p3240,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3242=som_seqtrain(p3241,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep24,tep24]=som_quality(p3242,W)
clear functions
```

```
tic
p3250=som_randinit(W,'msize',[15 1],'rect','sheet');
p3251=som_seqtrain(p3250,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3252=som_seqtrain(p3251,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep25,tep25]=som_quality(p3252,W)
clear functions

tic
p3260=som_randinit(W,'msize',[15 1],'rect','sheet');
p3261=som_seqtrain(p3260,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3262=som_seqtrain(p3261,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep26,tep26]=som_quality(p3262,W)
clear functions

tic
p3270=som_randinit(W,'msize',[15 1],'rect','sheet');
p3271=som_seqtrain(p3270,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3272=som_seqtrain(p3271,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep27,tep27]=som_quality(p3272,W)
clear functions

tic
p3280=som_randinit(W,'msize',[15 1],'rect','sheet');
p3281=som_seqtrain(p3280,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3282=som_seqtrain(p3281,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep28,tep28]=som_quality(p3282,W)
clear functions

tic
p3290=som_randinit(W,'msize',[15 1],'rect','sheet');
p3291=som_seqtrain(p3290,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3292=som_seqtrain(p3291,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep29,tep29]=som_quality(p3292,W)
clear functions

tic
p3300=som_randinit(W,'msize',[15 1],'rect','sheet');
p3301=som_seqtrain(p3300,W,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3,
'trainlen',5,'trainlen_type','epochs','neigh','bubble');
p3302=som_seqtrain(p3301,W,'radius_ini',2,'radius_fin',0,'alpha_ini',0.05
,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep30,tep30]=som_quality(p3302,W)
clear functions

% A1

tic
a1010=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1011=som_seqtrain(a1010,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1012=som_seqtrain(a1011,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
```

```
[qep01,tep01]=som_quality(a1012,W1)
clear functions

tic
a1020=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1021=som_seqtrain(a1020,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1022=som_seqtrain(a1021,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep02,tep02]=som_quality(a1022,W1)
clear functions

tic
a1030=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1031=som_seqtrain(a1030,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1032=som_seqtrain(a1031,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep03,tep03]=som_quality(a1032,W1)
clear functions

tic
a1040=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1041=som_seqtrain(a1040,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1042=som_seqtrain(a1041,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep04,tep04]=som_quality(a1042,W1)
clear functions

tic
a1050=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1051=som_seqtrain(a1050,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1052=som_seqtrain(a1051,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep05,tep05]=som_quality(a1052,W1)
clear functions

tic
a1060=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1061=som_seqtrain(a1060,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1062=som_seqtrain(a1061,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep06,tep06]=som_quality(a1062,W1)
clear functions

tic
a1070=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1071=som_seqtrain(a1070,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1072=som_seqtrain(a1071,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep07,tep07]=som_quality(a1072,W1)
clear functions

tic
a1080=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1081=som_seqtrain(a1080,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1082=som_seqtrain(a1081,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
```

```
toc
[qep08,tep08]=som_quality(a1082,W1)
clear functions

tic
a1090=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1091=som_seqtrain(a1090,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1092=som_seqtrain(a1091,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep09,tep09]=som_quality(a1092,W1)
clear functions

tic
a1100=som_randinit(W1'msize',[5 1],'rect','sheet');
a1101=som_seqtrain(a1100,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1102=som_seqtrain(a1101,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep10,tep10]=som_quality(a1102,W1)
clear functions

tic
a1110=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1111=som_seqtrain(a1110,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1112=som_seqtrain(a1111,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep11,tep11]=som_quality(a1112,W1)
clear functions

tic
a1120=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1121=som_seqtrain(a1120,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1122=som_seqtrain(a1121,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep12,tep12]=som_quality(a1122,W1)
clear functions

tic
a1130=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1131=som_seqtrain(a1130,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1132=som_seqtrain(a1131,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep13,tep13]=som_quality(a1132,W1)
clear functions

tic
a1140=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1141=som_seqtrain(a1140,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1142=som_seqtrain(a1141,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep14,tep14]=som_quality(a1142,W1)
clear functions

tic
a1150=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1151=som_seqtrain(a1150,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
```

```
a1152=som_seqtrain(a1151,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep15,tep15]=som_quality(a1152,W1)
clear functions

tic
a1160=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1161=som_seqtrain(a1160,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1162=som_seqtrain(a1161,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep16,tep16]=som_quality(a1162,W1)
clear functions

tic
a1170=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1171=som_seqtrain(a1170,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1172=som_seqtrain(a1171,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep17,tep17]=som_quality(a1172,W1)
clear functions

tic
a1180=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1181=som_seqtrain(a1180,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1182=som_seqtrain(a1181,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep18,tep18]=som_quality(a1182,W1)
clear functions

tic
a1190=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1191=som_seqtrain(a1190,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1192=som_seqtrain(a1191,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep19,tep19]=som_quality(a1192,W1)
clear functions

tic
a1200=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1201=som_seqtrain(a1200,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1202=som_seqtrain(a1201,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep20,tep20]=som_quality(a1202,W1)
clear functions

tic
a1210=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1211=som_seqtrain(a1210,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1212=som_seqtrain(a1211,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep21,tep21]=som_quality(a1212,W1)
clear functions

tic
a1220=som_randinit(W1,'msize',[5 1],'rect','sheet');
```

```
a1221=som_seqtrain(a1220,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1222=som_seqtrain(a1221,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep22,tep22]=som_quality(a1222,W1)
clear functions

tic
a1230=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1231=som_seqtrain(a1230,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1232=som_seqtrain(a1231,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep23,tep23]=som_quality(a1232,W1)
clear functions

tic
a1240=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1241=som_seqtrain(a1240,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1242=som_seqtrain(a1241,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep24,tep24]=som_quality(a1242,W1)
clear functions

tic
a1250=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1251=som_seqtrain(a1250,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1252=som_seqtrain(a1251,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep25,tep25]=som_quality(a1252,W1)
clear functions

tic
a1260=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1261=som_seqtrain(a1260,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1262=som_seqtrain(a1261,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep26,tep26]=som_quality(a1262,W1)
clear functions

tic
a1270=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1271=som_seqtrain(a1270,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1272=som_seqtrain(a1271,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep27,tep27]=som_quality(a1272,W1)
clear functions

tic
a1280=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1281=som_seqtrain(a1280,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1282=som_seqtrain(a1281,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep28,tep28]=som_quality(a1282,W1)
clear functions

tic
```

```
a1290=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1291=som_seqtrain(a1290,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1292=som_seqtrain(a1291,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep29,tep29]=som_quality(a1292,W1)
clear functions

tic
a1300=som_randinit(W1,'msize',[5 1],'rect','sheet');
a1301=som_seqtrain(a1300,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a1302=som_seqtrain(a1301,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep30,tep30]=som_quality(a1302,W1)
clear functions

% A2

tic
a2010=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2011=som_seqtrain(a2010,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2012=som_seqtrain(a2011,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep01,tep01]=som_quality(a2012,W1)
clear functions

tic
a2020=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2021=som_seqtrain(a2020,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2022=som_seqtrain(a2021,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep02,tep02]=som_quality(a2022,W1)
clear functions

tic
a2030=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2031=som_seqtrain(a2030,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2032=som_seqtrain(a2031,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep03,tep03]=som_quality(a2032,W1)
clear functions

tic
a2040=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2041=som_seqtrain(a2040,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2042=som_seqtrain(a2041,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep04,tep04]=som_quality(a2042,W1)
clear functions

tic
a2050=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2051=som_seqtrain(a2050,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2052=som_seqtrain(a2051,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep05,tep05]=som_quality(a2052,W1)
```

```
clear functions

tic
a2060=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2061=som_seqtrain(a2060,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2062=som_seqtrain(a2061,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep06,tep06]=som_quality(a2062,W1)
clear functions

tic
a2070=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2071=som_seqtrain(a2070,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2072=som_seqtrain(a2071,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep07,tep07]=som_quality(a2072,W1)
clear functions

tic
a2080=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2081=som_seqtrain(a2080,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2082=som_seqtrain(a2081,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep08,tep08]=som_quality(a2082,W1)
clear functions

tic
a2090=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2091=som_seqtrain(a2090,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2092=som_seqtrain(a2091,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep09,tep09]=som_quality(a2092,W1)
clear functions

tic
a2100=som_randinit(W1'msize',[10 1],'rect','sheet');
a2101=som_seqtrain(a2100,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2102=som_seqtrain(a2101,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep10,tep10]=som_quality(a2102,W1)
clear functions

tic
a2110=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2111=som_seqtrain(a2110,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2112=som_seqtrain(a2111,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep11,tep11]=som_quality(a2112,W1)
clear functions

tic
a2120=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2121=som_seqtrain(a2120,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2122=som_seqtrain(a2121,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
```

```
[qep12,tep12]=som_quality(a2122,W1)
clear functions

tic
a2130=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2131=som_seqtrain(a2130,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2132=som_seqtrain(a2131,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep13,tep13]=som_quality(a2132,W1)
clear functions

tic
a2140=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2141=som_seqtrain(a2140,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2142=som_seqtrain(a2141,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep14,tep14]=som_quality(a2142,W1)
clear functions

tic
a2150=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2151=som_seqtrain(a2150,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2152=som_seqtrain(a2151,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep15,tep15]=som_quality(a2152,W1)
clear functions

tic
a2160=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2161=som_seqtrain(a2160,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2162=som_seqtrain(a2161,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep16,tep16]=som_quality(a2162,W1)
clear functions

tic
a2170=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2171=som_seqtrain(a2170,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2172=som_seqtrain(a2171,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep17,tep17]=som_quality(a2172,W1)
clear functions

tic
a2180=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2181=som_seqtrain(a2180,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2182=som_seqtrain(a2181,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep18,tep18]=som_quality(a2182,W1)
clear functions

tic
a2190=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2191=som_seqtrain(a2190,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2192=som_seqtrain(a2191,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
```

```
toc
[qep19,tep19]=som_quality(a2192,W1)
clear functions

tic
a2200=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2201=som_seqtrain(a2200,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2202=som_seqtrain(a2201,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep20,tep20]=som_quality(a2202,W1)
clear functions

tic
a2210=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2211=som_seqtrain(a2210,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2212=som_seqtrain(a2211,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep21,tep21]=som_quality(a2212,W1)
clear functions

tic
a2220=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2221=som_seqtrain(a2220,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2222=som_seqtrain(a2221,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep22,tep22]=som_quality(a2222,W1)
clear functions

tic
a2230=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2231=som_seqtrain(a2230,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2232=som_seqtrain(a2231,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep23,tep23]=som_quality(a2232,W1)
clear functions

tic
a2240=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2241=som_seqtrain(a2240,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2242=som_seqtrain(a2241,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep24,tep24]=som_quality(a2242,W1)
clear functions

tic
a2250=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2251=som_seqtrain(a2250,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2252=som_seqtrain(a2251,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep25,tep25]=som_quality(a2252,W1)
clear functions

tic
a2260=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2261=som_seqtrain(a2260,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
```

```
a2262=som_seqtrain(a2261,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep26,tep26]=som_quality(a2262,W1)
clear functions

tic
a2270=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2271=som_seqtrain(a2270,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2272=som_seqtrain(a2271,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep27,tep27]=som_quality(a2272,W1)
clear functions

tic
a2280=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2281=som_seqtrain(a2280,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2282=som_seqtrain(a2281,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep28,tep28]=som_quality(a2282,W1)
clear functions

tic
a2290=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2291=som_seqtrain(a2290,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2292=som_seqtrain(a2291,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep29,tep29]=som_quality(a2292,W1)
clear functions

tic
a2300=som_randinit(W1,'msize',[10 1],'rect','sheet');
a2301=som_seqtrain(a2300,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a2302=som_seqtrain(a2301,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep30,tep30]=som_quality(a2302,W1)
clear functions

% A3

tic
a3010=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3011=som_seqtrain(a3010,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3012=som_seqtrain(a3011,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep01,tep01]=som_quality(a3012,W1)
clear functions

tic
a3020=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3021=som_seqtrain(a3020,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3022=som_seqtrain(a3021,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep02,tep02]=som_quality(a3022,W1)
clear functions

tic
```

```
a3030=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3031=som_seqtrain(a3030,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3032=som_seqtrain(a3031,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep03,tep03]=som_quality(a3032,W1)
clear functions

tic
a3040=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3041=som_seqtrain(a3040,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3042=som_seqtrain(a3041,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep04,tep04]=som_quality(a3042,W1)
clear functions

tic
a3050=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3051=som_seqtrain(a3050,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3052=som_seqtrain(a3051,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep05,tep05]=som_quality(a3052,W1)
clear functions

tic
a3060=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3061=som_seqtrain(a3060,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3062=som_seqtrain(a3061,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep06,tep06]=som_quality(a3062,W1)
clear functions

tic
a3070=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3071=som_seqtrain(a3070,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3072=som_seqtrain(a3071,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep07,tep07]=som_quality(a3072,W1)
clear functions

tic
a3080=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3081=som_seqtrain(a3080,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3082=som_seqtrain(a3081,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep08,tep08]=som_quality(a3082,W1)
clear functions

tic
a3090=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3091=som_seqtrain(a3090,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3092=som_seqtrain(a3091,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep09,tep09]=som_quality(a3092,W1)
clear functions
```

```
tic
a3100=som_randinit(W1'msize',[15 1],'rect','sheet');
a3101=som_seqtrain(a3100,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3102=som_seqtrain(a3101,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep10,tep10]=som_quality(a3102,W1)
clear functions

tic
a3110=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3111=som_seqtrain(a3110,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3112=som_seqtrain(a3111,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep11,tep11]=som_quality(a3112,W1)
clear functions

tic
a3120=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3121=som_seqtrain(a3120,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3122=som_seqtrain(a3121,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep12,tep12]=som_quality(a3122,W1)
clear functions

tic
a3130=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3131=som_seqtrain(a3130,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3132=som_seqtrain(a3131,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep13,tep13]=som_quality(a3132,W1)
clear functions

tic
a3140=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3141=som_seqtrain(a3140,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3142=som_seqtrain(a3141,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep14,tep14]=som_quality(a3142,W1)
clear functions

tic
a3150=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3151=som_seqtrain(a3150,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3152=som_seqtrain(a3151,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep15,tep15]=som_quality(a3152,W1)
clear functions

tic
a3160=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3161=som_seqtrain(a3160,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3162=som_seqtrain(a3161,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep16,tep16]=som_quality(a3162,W1)
clear functions
```

```
tic
a3170=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3171=som_seqtrain(a3170,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3172=som_seqtrain(a3171,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep17,tep17]=som_quality(a3172,W1)
clear functions

tic
a3180=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3181=som_seqtrain(a3180,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3182=som_seqtrain(a3181,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep18,tep18]=som_quality(a3182,W1)
clear functions

tic
a3190=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3191=som_seqtrain(a3190,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3192=som_seqtrain(a3191,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep19,tep19]=som_quality(a3192,W1)
clear functions

tic
a3200=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3201=som_seqtrain(a3200,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3202=som_seqtrain(a3201,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep20,tep20]=som_quality(a3202,W1)
clear functions

tic
a3210=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3211=som_seqtrain(a3210,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3212=som_seqtrain(a3211,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep21,tep21]=som_quality(a3212,W1)
clear functions

tic
a3220=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3221=som_seqtrain(a3220,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3222=som_seqtrain(a3221,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep22,tep22]=som_quality(a3222,W1)
clear functions

tic
a3230=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3231=som_seqtrain(a3230,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3232=som_seqtrain(a3231,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep23,tep23]=som_quality(a3232,W1)
clear functions
```

```
tic
a3240=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3241=som_seqtrain(a3240,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3242=som_seqtrain(a3241,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep24,tep24]=som_quality(a3242,W1)
clear functions

tic
a3250=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3251=som_seqtrain(a3250,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3252=som_seqtrain(a3251,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep25,tep25]=som_quality(a3252,W1)
clear functions

tic
a3260=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3261=som_seqtrain(a3260,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3262=som_seqtrain(a3261,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep26,tep26]=som_quality(a3262,W1)
clear functions

tic
a3270=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3271=som_seqtrain(a3270,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3272=som_seqtrain(a3271,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep27,tep27]=som_quality(a3272,W1)
clear functions

tic
a3280=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3281=som_seqtrain(a3280,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3282=som_seqtrain(a3281,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep28,tep28]=som_quality(a3282,W1)
clear functions

tic
a3290=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3291=som_seqtrain(a3290,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3292=som_seqtrain(a3291,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep29,tep29]=som_quality(a3292,W1)
clear functions

tic
a3300=som_randinit(W1,'msize',[15 1],'rect','sheet');
a3301=som_seqtrain(a3300,W1,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
a3302=som_seqtrain(a3301,W1,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep30,tep30]=som_quality(a3302,W1)
clear functions
```

```
% B1

tic
b1010=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1011=som_seqtrain(b1010,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1012=som_seqtrain(b1011,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep01,tep01]=som_quality(b1012,W2)
clear functions

tic
b1020=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1021=som_seqtrain(b1020,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1022=som_seqtrain(b1021,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep02,tep02]=som_quality(b1022,W2)
clear functions

tic
b1030=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1031=som_seqtrain(b1030,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1032=som_seqtrain(b1031,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep03,tep03]=som_quality(b1032,W2)
clear functions

tic
b1040=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1041=som_seqtrain(b1040,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1042=som_seqtrain(b1041,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep04,tep04]=som_quality(b1042,W2)
clear functions

tic
b1050=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1051=som_seqtrain(b1050,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1052=som_seqtrain(b1051,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep05,tep05]=som_quality(b1052,W2)
clear functions

tic
b1060=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1061=som_seqtrain(b1060,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1062=som_seqtrain(b1061,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep06,tep06]=som_quality(b1062,W2)
clear functions

tic
b1070=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1071=som_seqtrain(b1070,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1072=som_seqtrain(b1071,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
```

```
[qep07,tep07]=som_quality(b1072,W2)
clear functions

tic
b1080=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1081=som_seqtrain(b1080,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1082=som_seqtrain(b1081,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep08,tep08]=som_quality(b1082,W2)
clear functions

tic
b1090=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1091=som_seqtrain(b1090,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1092=som_seqtrain(b1091,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep09,tep09]=som_quality(b1092,W2)
clear functions

tic
b1100=som_randinit(W2'msize',[5 1],'rect','sheet');
b1101=som_seqtrain(b1100,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1102=som_seqtrain(b1101,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep10,tep10]=som_quality(b1102,W2)
clear functions

tic
b1110=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1111=som_seqtrain(b1110,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1112=som_seqtrain(b1111,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep11,tep11]=som_quality(b1112,W2)
clear functions

tic
b1120=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1121=som_seqtrain(b1120,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1122=som_seqtrain(b1121,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep12,tep12]=som_quality(b1122,W2)
clear functions

tic
b1130=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1131=som_seqtrain(b1130,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1132=som_seqtrain(b1131,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep13,tep13]=som_quality(b1132,W2)
clear functions

tic
b1140=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1141=som_seqtrain(b1140,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1142=som_seqtrain(b1141,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
```

```
toc
[qep14,tep14]=som_quality(b1142,W2)
clear functions

tic
b1150=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1151=som_seqtrain(b1150,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1152=som_seqtrain(b1151,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep15,tep15]=som_quality(b1152,W2)
clear functions

tic
b1160=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1161=som_seqtrain(b1160,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1162=som_seqtrain(b1161,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep16,tep16]=som_quality(b1162,W2)
clear functions

tic
b1170=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1171=som_seqtrain(b1170,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1172=som_seqtrain(b1171,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep17,tep17]=som_quality(b1172,W2)
clear functions

tic
b1180=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1181=som_seqtrain(b1180,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1182=som_seqtrain(b1181,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep18,tep18]=som_quality(b1182,W2)
clear functions

tic
b1190=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1191=som_seqtrain(b1190,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1192=som_seqtrain(b1191,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep19,tep19]=som_quality(b1192,W2)
clear functions

tic
b1200=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1201=som_seqtrain(b1200,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1202=som_seqtrain(b1201,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep20,tep20]=som_quality(b1202,W2)
clear functions

tic
b1210=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1211=som_seqtrain(b1210,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
```

```
b1212=som_seqtrain(b1211,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep21,tep21]=som_quality(b1212,W2)
clear functions

tic
b1220=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1221=som_seqtrain(b1220,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1222=som_seqtrain(b1221,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep22,tep22]=som_quality(b1222,W2)
clear functions

tic
b1230=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1231=som_seqtrain(b1230,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1232=som_seqtrain(b1231,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep23,tep23]=som_quality(b1232,W2)
clear functions

tic
b1240=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1241=som_seqtrain(b1240,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1242=som_seqtrain(b1241,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep24,tep24]=som_quality(b1242,W2)
clear functions

tic
b1250=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1251=som_seqtrain(b1250,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1252=som_seqtrain(b1251,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep25,tep25]=som_quality(b1252,W2)
clear functions

tic
b1260=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1261=som_seqtrain(b1260,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1262=som_seqtrain(b1261,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep26,tep26]=som_quality(b1262,W2)
clear functions

tic
b1270=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1271=som_seqtrain(b1270,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1272=som_seqtrain(b1271,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep27,tep27]=som_quality(b1272,W2)
clear functions

tic
b1280=som_randinit(W2,'msize',[5 1],'rect','sheet');
```

```
b1281=som_seqtrain(b1280,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1282=som_seqtrain(b1281,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep28,tep28]=som_quality(b1282,W2)
clear functions

tic
b1290=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1291=som_seqtrain(b1290,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1292=som_seqtrain(b1291,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep29,tep29]=som_quality(b1292,W2)
clear functions

tic
b1300=som_randinit(W2,'msize',[5 1],'rect','sheet');
b1301=som_seqtrain(b1300,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b1302=som_seqtrain(b1301,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep30,tep30]=som_quality(b1302,W2)
clear functions

% B2

tic
b2010=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2011=som_seqtrain(b2010,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2012=som_seqtrain(b2011,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep01,tep01]=som_quality(b2012,W2)
clear functions

tic
b2020=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2021=som_seqtrain(b2020,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2022=som_seqtrain(b2021,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep02,tep02]=som_quality(b2022,W2)
clear functions

tic
b2030=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2031=som_seqtrain(b2030,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2032=som_seqtrain(b2031,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep03,tep03]=som_quality(b2032,W2)
clear functions

tic
b2040=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2041=som_seqtrain(b2040,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2042=som_seqtrain(b2041,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep04,tep04]=som_quality(b2042,W2)
clear functions
```

```
tic
b2050=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2051=som_seqtrain(b2050,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2052=som_seqtrain(b2051,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep05,tep05]=som_quality(b2052,W2)
clear functions

tic
b2060=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2061=som_seqtrain(b2060,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2062=som_seqtrain(b2061,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep06,tep06]=som_quality(b2062,W2)
clear functions

tic
b2070=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2071=som_seqtrain(b2070,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2072=som_seqtrain(b2071,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep07,tep07]=som_quality(b2072,W2)
clear functions

tic
b2080=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2081=som_seqtrain(b2080,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2082=som_seqtrain(b2081,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep08,tep08]=som_quality(b2082,W2)
clear functions

tic
b2090=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2091=som_seqtrain(b2090,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2092=som_seqtrain(b2091,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep09,tep09]=som_quality(b2092,W2)
clear functions

tic
b2100=som_randinit(W2'msize',[10 1],'rect','sheet');
b2101=som_seqtrain(b2100,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2102=som_seqtrain(b2101,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep10,tep10]=som_quality(b2102,W2)
clear functions

tic
b2110=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2111=som_seqtrain(b2110,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2112=som_seqtrain(b2111,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep11,tep11]=som_quality(b2112,W2)
clear functions
```

```
tic
b2120=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2121=som_seqtrain(b2120,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2122=som_seqtrain(b2121,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep12,tep12]=som_quality(b2122,W2)
clear functions

tic
b2130=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2131=som_seqtrain(b2130,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2132=som_seqtrain(b2131,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep13,tep13]=som_quality(b2132,W2)
clear functions

tic
b2140=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2141=som_seqtrain(b2140,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2142=som_seqtrain(b2141,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep14,tep14]=som_quality(b2142,W2)
clear functions

tic
b2150=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2151=som_seqtrain(b2150,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2152=som_seqtrain(b2151,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep15,tep15]=som_quality(b2152,W2)
clear functions

tic
b2160=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2161=som_seqtrain(b2160,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2162=som_seqtrain(b2161,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep16,tep16]=som_quality(b2162,W2)
clear functions

tic
b2170=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2171=som_seqtrain(b2170,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2172=som_seqtrain(b2171,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep17,tep17]=som_quality(b2172,W2)
clear functions

tic
b2180=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2181=som_seqtrain(b2180,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2182=som_seqtrain(b2181,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep18,tep18]=som_quality(b2182,W2)
clear functions
```

```
tic
b2190=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2191=som_seqtrain(b2190,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2192=som_seqtrain(b2191,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep19,tep19]=som_quality(b2192,W2)
clear functions

tic
b2200=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2201=som_seqtrain(b2200,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2202=som_seqtrain(b2201,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep20,tep20]=som_quality(b2202,W2)
clear functions

tic
b2210=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2211=som_seqtrain(b2210,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2212=som_seqtrain(b2211,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep21,tep21]=som_quality(b2212,W2)
clear functions

tic
b2220=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2221=som_seqtrain(b2220,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2222=som_seqtrain(b2221,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep22,tep22]=som_quality(b2222,W2)
clear functions

tic
b2230=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2231=som_seqtrain(b2230,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2232=som_seqtrain(b2231,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep23,tep23]=som_quality(b2232,W2)
clear functions

tic
b2240=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2241=som_seqtrain(b2240,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2242=som_seqtrain(b2241,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep24,tep24]=som_quality(b2242,W2)
clear functions

tic
b2250=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2251=som_seqtrain(b2250,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2252=som_seqtrain(b2251,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep25,tep25]=som_quality(b2252,W2)
clear functions
```

```
tic
b2260=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2261=som_seqtrain(b2260,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2262=som_seqtrain(b2261,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep26,tep26]=som_quality(b2262,W2)
clear functions

tic
b2270=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2271=som_seqtrain(b2270,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2272=som_seqtrain(b2271,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep27,tep27]=som_quality(b2272,W2)
clear functions

tic
b2280=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2281=som_seqtrain(b2280,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2282=som_seqtrain(b2281,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep28,tep28]=som_quality(b2282,W2)
clear functions

tic
b2290=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2291=som_seqtrain(b2290,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2292=som_seqtrain(b2291,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep29,tep29]=som_quality(b2292,W2)
clear functions

tic
b2300=som_randinit(W2,'msize',[10 1],'rect','sheet');
b2301=som_seqtrain(b2300,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b2302=som_seqtrain(b2301,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep30,tep30]=som_quality(b2302,W2)
clear functions

% B3
tic
b3010=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3011=som_seqtrain(b3010,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3012=som_seqtrain(b3011,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep01,tep01]=som_quality(b3012,W2)
clear functions

tic
b3020=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3021=som_seqtrain(b3020,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3022=som_seqtrain(b3021,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep02,tep02]=som_quality(b3022,W2)
```

```
clear functions

tic
b3030=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3031=som_seqtrain(b3030,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3032=som_seqtrain(b3031,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep03,tep03]=som_quality(b3032,W2)
clear functions

tic
b3040=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3041=som_seqtrain(b3040,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3042=som_seqtrain(b3041,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep04,tep04]=som_quality(b3042,W2)
clear functions

tic
b3050=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3051=som_seqtrain(b3050,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3052=som_seqtrain(b3051,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep05,tep05]=som_quality(b3052,W2)
clear functions

tic
b3060=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3061=som_seqtrain(b3060,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3062=som_seqtrain(b3061,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep06,tep06]=som_quality(b3062,W2)
clear functions

tic
b3070=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3071=som_seqtrain(b3070,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3072=som_seqtrain(b3071,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep07,tep07]=som_quality(b3072,W2)
clear functions

tic
b3080=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3081=som_seqtrain(b3080,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3082=som_seqtrain(b3081,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep08,tep08]=som_quality(b3082,W2)
clear functions

tic
b3090=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3091=som_seqtrain(b3090,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3092=som_seqtrain(b3091,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
```

```
[qep09,tep09]=som_quality(b3092,W2)
clear functions

tic
b3100=som_randinit(W2'msize',[15 1],'rect','sheet');
b3101=som_seqtrain(b3100,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3102=som_seqtrain(b3101,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep10,tep10]=som_quality(b3102,W2)
clear functions

tic
b3110=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3111=som_seqtrain(b3110,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3112=som_seqtrain(b3111,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep11,tep11]=som_quality(b3112,W2)
clear functions

tic
b3120=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3121=som_seqtrain(b3120,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3122=som_seqtrain(b3121,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep12,tep12]=som_quality(b3122,W2)
clear functions

tic
b3130=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3131=som_seqtrain(b3130,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3132=som_seqtrain(b3131,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep13,tep13]=som_quality(b3132,W2)
clear functions

tic
b3140=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3141=som_seqtrain(b3140,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3142=som_seqtrain(b3141,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep14,tep14]=som_quality(b3142,W2)
clear functions

tic
b3150=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3151=som_seqtrain(b3150,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3152=som_seqtrain(b3151,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep15,tep15]=som_quality(b3152,W2)
clear functions

tic
b3160=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3161=som_seqtrain(b3160,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3162=som_seqtrain(b3161,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
```

```
toc
[qep16,tep16]=som_quality(b3162,W2)
clear functions

tic
b3170=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3171=som_seqtrain(b3170,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3172=som_seqtrain(b3171,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep17,tep17]=som_quality(b3172,W2)
clear functions

tic
b3180=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3181=som_seqtrain(b3180,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3182=som_seqtrain(b3181,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep18,tep18]=som_quality(b3182,W2)
clear functions

tic
b3190=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3191=som_seqtrain(b3190,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3192=som_seqtrain(b3191,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep19,tep19]=som_quality(b3192,W2)
clear functions

tic
b3200=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3201=som_seqtrain(b3200,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3202=som_seqtrain(b3201,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep20,tep20]=som_quality(b3202,W2)
clear functions

tic
b3210=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3211=som_seqtrain(b3210,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3212=som_seqtrain(b3211,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep21,tep21]=som_quality(b3212,W2)
clear functions

tic
b3220=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3221=som_seqtrain(b3220,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3222=som_seqtrain(b3221,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep22,tep22]=som_quality(b3222,W2)
clear functions

tic
b3230=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3231=som_seqtrain(b3230,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
```

```
b3232=som_seqtrain(b3231,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep23,tep23]=som_quality(b3232,W2)
clear functions

tic
b3240=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3241=som_seqtrain(b3240,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3242=som_seqtrain(b3241,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep24,tep24]=som_quality(b3242,W2)
clear functions

tic
b3250=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3251=som_seqtrain(b3250,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3252=som_seqtrain(b3251,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep25,tep25]=som_quality(b3252,W2)
clear functions

tic
b3260=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3261=som_seqtrain(b3260,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3262=som_seqtrain(b3261,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep26,tep26]=som_quality(b3262,W2)
clear functions

tic
b3270=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3271=som_seqtrain(b3270,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3272=som_seqtrain(b3271,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep27,tep27]=som_quality(b3272,W2)
clear functions

tic
b3280=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3281=som_seqtrain(b3280,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3282=som_seqtrain(b3281,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep28,tep28]=som_quality(b3282,W2)
clear functions

tic
b3290=som_randinit(W2,'msize',[15 1],'rect','sheet');
b3291=som_seqtrain(b3290,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3292=som_seqtrain(b3291,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep29,tep29]=som_quality(b3292,W2)
clear functions

tic
b3300=som_randinit(W2,'msize',[15 1],'rect','sheet');
```

```
b3301=som_seqtrain(b3300,W2,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
b3302=som_seqtrain(b3301,W2,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep30,tep30]=som_quality(b3302,W2)
clear functions

% C1

tic
c1010=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1011=som_seqtrain(c1010,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1012=som_seqtrain(c1011,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep01,tep01]=som_quality(c1012,W3)
clear functions

tic
c1020=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1021=som_seqtrain(c1020,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1022=som_seqtrain(c1021,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep02,tep02]=som_quality(c1022,W3)
clear functions

tic
c1030=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1031=som_seqtrain(c1030,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1032=som_seqtrain(c1031,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep03,tep03]=som_quality(c1032,W3)
clear functions

tic
c1040=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1041=som_seqtrain(c1040,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1042=som_seqtrain(c1041,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep04,tep04]=som_quality(c1042,W3)
clear functions

tic
c1050=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1051=som_seqtrain(c1050,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1052=som_seqtrain(c1051,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep05,tep05]=som_quality(c1052,W3)
clear functions

tic
c1060=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1061=som_seqtrain(c1060,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1062=som_seqtrain(c1061,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep06,tep06]=som_quality(c1062,W3)
clear functions
```

```
tic
c1070=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1071=som_seqtrain(c1070,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1072=som_seqtrain(c1071,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep07,tep07]=som_quality(c1072,W3)
clear functions

tic
c1080=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1081=som_seqtrain(c1080,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1082=som_seqtrain(c1081,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep08,tep08]=som_quality(c1082,W3)
clear functions

tic
c1090=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1091=som_seqtrain(c1090,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1092=som_seqtrain(c1091,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep09,tep09]=som_quality(c1092,W3)
clear functions

tic
c1100=som_randinit(W3'msize',[5 1],'rect','sheet');
c1101=som_seqtrain(c1100,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1102=som_seqtrain(c1101,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep10,tep10]=som_quality(c1102,W3)
clear functions

tic
c1110=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1111=som_seqtrain(c1110,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1112=som_seqtrain(c1111,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep11,tep11]=som_quality(c1112,W3)
clear functions

tic
c1120=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1121=som_seqtrain(c1120,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1122=som_seqtrain(c1121,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep12,tep12]=som_quality(c1122,W3)
clear functions

tic
c1130=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1131=som_seqtrain(c1130,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1132=som_seqtrain(c1131,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep13,tep13]=som_quality(c1132,W3)
clear functions
```

```
tic
c1140=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1141=som_seqtrain(c1140,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1142=som_seqtrain(c1141,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep14,tep14]=som_quality(c1142,W3)
clear functions

tic
c1150=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1151=som_seqtrain(c1150,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1152=som_seqtrain(c1151,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep15,tep15]=som_quality(c1152,W3)
clear functions

tic
c1160=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1161=som_seqtrain(c1160,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1162=som_seqtrain(c1161,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep16,tep16]=som_quality(c1162,W3)
clear functions

tic
c1170=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1171=som_seqtrain(c1170,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1172=som_seqtrain(c1171,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep17,tep17]=som_quality(c1172,W3)
clear functions

tic
c1180=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1181=som_seqtrain(c1180,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1182=som_seqtrain(c1181,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep18,tep18]=som_quality(c1182,W3)
clear functions

tic
c1190=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1191=som_seqtrain(c1190,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1192=som_seqtrain(c1191,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep19,tep19]=som_quality(c1192,W3)
clear functions

tic
c1200=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1201=som_seqtrain(c1200,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1202=som_seqtrain(c1201,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep20,tep20]=som_quality(c1202,W3)
clear functions
```

```
tic
c1210=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1211=som_seqtrain(c1210,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1212=som_seqtrain(c1211,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep21,tep21]=som_quality(c1212,W3)
clear functions

tic
c1220=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1221=som_seqtrain(c1220,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1222=som_seqtrain(c1221,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep22,tep22]=som_quality(c1222,W3)
clear functions

tic
c1230=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1231=som_seqtrain(c1230,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1232=som_seqtrain(c1231,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep23,tep23]=som_quality(c1232,W3)
clear functions

tic
c1240=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1241=som_seqtrain(c1240,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1242=som_seqtrain(c1241,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep24,tep24]=som_quality(c1242,W3)
clear functions

tic
c1250=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1251=som_seqtrain(c1250,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1252=som_seqtrain(c1251,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep25,tep25]=som_quality(c1252,W3)
clear functions

tic
c1260=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1261=som_seqtrain(c1260,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1262=som_seqtrain(c1261,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep26,tep26]=som_quality(c1262,W3)
clear functions

tic
c1270=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1271=som_seqtrain(c1270,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1272=som_seqtrain(c1271,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep27,tep27]=som_quality(c1272,W3)
clear functions
```

```
tic
c1280=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1281=som_seqtrain(c1280,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1282=som_seqtrain(c1281,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep28,tep28]=som_quality(c1282,W3)
clear functions

tic
c1290=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1291=som_seqtrain(c1290,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1292=som_seqtrain(c1291,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep29,tep29]=som_quality(c1292,W3)
clear functions

tic
c1300=som_randinit(W3,'msize',[5 1],'rect','sheet');
c1301=som_seqtrain(c1300,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c1302=som_seqtrain(c1301,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep30,tep30]=som_quality(c1302,W3)
clear functions

% C2

tic
c2010=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2011=som_seqtrain(c2010,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2012=som_seqtrain(c2011,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep01,tep01]=som_quality(c2012,W3)
clear functions

tic
c2020=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2021=som_seqtrain(c2020,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2022=som_seqtrain(c2021,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep02,tep02]=som_quality(c2022,W3)
clear functions

tic
c2030=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2031=som_seqtrain(c2030,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2032=som_seqtrain(c2031,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep03,tep03]=som_quality(c2032,W3)
clear functions

tic
c2040=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2041=som_seqtrain(c2040,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2042=som_seqtrain(c2041,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
```

```
[qep04,tep04]=som_quality(c2042,W3)
clear functions

tic
c2050=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2051=som_seqtrain(c2050,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2052=som_seqtrain(c2051,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep05,tep05]=som_quality(c2052,W3)
clear functions

tic
c2060=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2061=som_seqtrain(c2060,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2062=som_seqtrain(c2061,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep06,tep06]=som_quality(c2062,W3)
clear functions

tic
c2070=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2071=som_seqtrain(c2070,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2072=som_seqtrain(c2071,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep07,tep07]=som_quality(c2072,W3)
clear functions

tic
c2080=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2081=som_seqtrain(c2080,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2082=som_seqtrain(c2081,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep08,tep08]=som_quality(c2082,W3)
clear functions

tic
c2090=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2091=som_seqtrain(c2090,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2092=som_seqtrain(c2091,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep09,tep09]=som_quality(c2092,W3)
clear functions

tic
c2100=som_randinit(W3'msize',[10 1],'rect','sheet');
c2101=som_seqtrain(c2100,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2102=som_seqtrain(c2101,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep10,tep10]=som_quality(c2102,W3)
clear functions

tic
c2110=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2111=som_seqtrain(c2110,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2112=som_seqtrain(c2111,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
```

```
toc
[qep11,tep11]=som_quality(c2112,W3)
clear functions

tic
c2120=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2121=som_seqtrain(c2120,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2122=som_seqtrain(c2121,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep12,tep12]=som_quality(c2122,W3)
clear functions

tic
c2130=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2131=som_seqtrain(c2130,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2132=som_seqtrain(c2131,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep13,tep13]=som_quality(c2132,W3)
clear functions

tic
c2140=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2141=som_seqtrain(c2140,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2142=som_seqtrain(c2141,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep14,tep14]=som_quality(c2142,W3)
clear functions

tic
c2150=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2151=som_seqtrain(c2150,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2152=som_seqtrain(c2151,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep15,tep15]=som_quality(c2152,W3)
clear functions

tic
c2160=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2161=som_seqtrain(c2160,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2162=som_seqtrain(c2161,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep16,tep16]=som_quality(c2162,W3)
clear functions

tic
c2170=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2171=som_seqtrain(c2170,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2172=som_seqtrain(c2171,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep17,tep17]=som_quality(c2172,W3)
clear functions

tic
c2180=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2181=som_seqtrain(c2180,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
```

```
c2182=som_seqtrain(c2181,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep18,tep18]=som_quality(c2182,W3)
clear functions

tic
c2190=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2191=som_seqtrain(c2190,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2192=som_seqtrain(c2191,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep19,tep19]=som_quality(c2192,W3)
clear functions

tic
c2200=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2201=som_seqtrain(c2200,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2202=som_seqtrain(c2201,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep20,tep20]=som_quality(c2202,W3)
clear functions

tic
c2210=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2211=som_seqtrain(c2210,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2212=som_seqtrain(c2211,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep21,tep21]=som_quality(c2212,W3)
clear functions

tic
c2220=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2221=som_seqtrain(c2220,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2222=som_seqtrain(c2221,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep22,tep22]=som_quality(c2222,W3)
clear functions

tic
c2230=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2231=som_seqtrain(c2230,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2232=som_seqtrain(c2231,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep23,tep23]=som_quality(c2232,W3)
clear functions

tic
c2240=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2241=som_seqtrain(c2240,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2242=som_seqtrain(c2241,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep24,tep24]=som_quality(c2242,W3)
clear functions

tic
c2250=som_randinit(W3,'msize',[10 1],'rect','sheet');
```

```
c2251=som_seqtrain(c2250,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2252=som_seqtrain(c2251,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep25,tep25]=som_quality(c2252,W3)
clear functions

tic
c2260=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2261=som_seqtrain(c2260,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2262=som_seqtrain(c2261,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep26,tep26]=som_quality(c2262,W3)
clear functions

tic
c2270=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2271=som_seqtrain(c2270,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2272=som_seqtrain(c2271,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep27,tep27]=som_quality(c2272,W3)
clear functions

tic
c2280=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2281=som_seqtrain(c2280,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2282=som_seqtrain(c2281,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep28,tep28]=som_quality(c2282,W3)
clear functions

tic
c2290=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2291=som_seqtrain(c2290,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2292=som_seqtrain(c2291,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep29,tep29]=som_quality(c2292,W3)
clear functions

tic
c2300=som_randinit(W3,'msize',[10 1],'rect','sheet');
c2301=som_seqtrain(c2300,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c2302=som_seqtrain(c2301,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep30,tep30]=som_quality(c2302,W3)
clear functions

% C3

tic
c3010=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3011=som_seqtrain(c3010,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3012=som_seqtrain(c3011,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep01,tep01]=som_quality(c3012,W3)
clear functions
```

```
tic
c3020=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3021=som_seqtrain(c3020,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3022=som_seqtrain(c3021,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep02,tep02]=som_quality(c3022,W3)
clear functions

tic
c3030=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3031=som_seqtrain(c3030,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3032=som_seqtrain(c3031,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep03,tep03]=som_quality(c3032,W3)
clear functions

tic
c3040=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3041=som_seqtrain(c3040,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3042=som_seqtrain(c3041,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep04,tep04]=som_quality(c3042,W3)
clear functions

tic
c3050=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3051=som_seqtrain(c3050,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3052=som_seqtrain(c3051,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep05,tep05]=som_quality(c3052,W3)
clear functions

tic
c3060=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3061=som_seqtrain(c3060,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3062=som_seqtrain(c3061,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep06,tep06]=som_quality(c3062,W3)
clear functions

tic
c3070=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3071=som_seqtrain(c3070,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3072=som_seqtrain(c3071,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep07,tep07]=som_quality(c3072,W3)
clear functions

tic
c3080=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3081=som_seqtrain(c3080,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3082=som_seqtrain(c3081,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep08,tep08]=som_quality(c3082,W3)
clear functions
```

```
tic
c3090=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3091=som_seqtrain(c3090,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3092=som_seqtrain(c3091,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep09,tep09]=som_quality(c3092,W3)
clear functions

tic
c3100=som_randinit(W3'msize',[15 1],'rect','sheet');
c3101=som_seqtrain(c3100,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3102=som_seqtrain(c3101,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep10,tep10]=som_quality(c3102,W3)
clear functions

tic
c3110=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3111=som_seqtrain(c3110,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3112=som_seqtrain(c3111,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep11,tep11]=som_quality(c3112,W3)
clear functions

tic
c3120=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3121=som_seqtrain(c3120,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3122=som_seqtrain(c3121,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep12,tep12]=som_quality(c3122,W3)
clear functions

tic
c3130=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3131=som_seqtrain(c3130,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3132=som_seqtrain(c3131,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep13,tep13]=som_quality(c3132,W3)
clear functions

tic
c3140=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3141=som_seqtrain(c3140,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3142=som_seqtrain(c3141,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep14,tep14]=som_quality(c3142,W3)
clear functions

tic
c3150=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3151=som_seqtrain(c3150,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3152=som_seqtrain(c3151,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep15,tep15]=som_quality(c3152,W3)
clear functions
```

```
tic
c3160=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3161=som_seqtrain(c3160,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3162=som_seqtrain(c3161,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep16,tep16]=som_quality(c3162,W3)
clear functions

tic
c3170=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3171=som_seqtrain(c3170,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3172=som_seqtrain(c3171,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep17,tep17]=som_quality(c3172,W3)
clear functions

tic
c3180=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3181=som_seqtrain(c3180,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3182=som_seqtrain(c3181,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep18,tep18]=som_quality(c3182,W3)
clear functions

tic
c3190=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3191=som_seqtrain(c3190,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3192=som_seqtrain(c3191,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep19,tep19]=som_quality(c3192,W3)
clear functions

tic
c3200=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3201=som_seqtrain(c3200,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3202=som_seqtrain(c3201,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep20,tep20]=som_quality(c3202,W3)
clear functions

tic
c3210=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3211=som_seqtrain(c3210,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3212=som_seqtrain(c3211,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep21,tep21]=som_quality(c3212,W3)
clear functions

tic
c3220=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3221=som_seqtrain(c3220,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3222=som_seqtrain(c3221,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep22,tep22]=som_quality(c3222,W3)
clear functions
```

```
tic
c3230=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3231=som_seqtrain(c3230,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3232=som_seqtrain(c3231,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep23,tep23]=som_quality(c3232,W3)
clear functions

tic
c3240=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3241=som_seqtrain(c3240,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3242=som_seqtrain(c3241,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep24,tep24]=som_quality(c3242,W3)
clear functions

tic
c3250=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3251=som_seqtrain(c3250,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3252=som_seqtrain(c3251,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep25,tep25]=som_quality(c3252,W3)
clear functions

tic
c3260=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3261=som_seqtrain(c3260,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3262=som_seqtrain(c3261,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep26,tep26]=som_quality(c3262,W3)
clear functions

tic
c3270=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3271=som_seqtrain(c3270,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3272=som_seqtrain(c3271,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep27,tep27]=som_quality(c3272,W3)
clear functions

tic
c3280=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3281=som_seqtrain(c3280,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3282=som_seqtrain(c3281,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep28,tep28]=som_quality(c3282,W3)
clear functions

tic
c3290=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3291=som_seqtrain(c3290,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3292=som_seqtrain(c3291,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep29,tep29]=som_quality(c3292,W3)
clear functions
```

```
tic
c3300=som_randinit(W3,'msize',[15 1],'rect','sheet');
c3301=som_seqtrain(c3300,W3,'radius_ini',3,'radius_fin',0,'alpha_ini',0.3
,'trainlen',5,'trainlen_type','epochs','neigh','bubble');
c3302=som_seqtrain(c3301,W3,'radius_ini',2,'radius_fin',0,'alpha_ini',0.0
5,'trainlen',10,'trainlen_type','epochs','neigh','bubble');
toc
[qep30,tep30]=som_quality(c3302,W3)
clear functions

% K-MEANS

% K1
tic; [k101,c101,err101]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k102,c102,err102]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k103,c103,err103]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k104,c104,err104]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k105,c105,err105]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k106,c106,err106]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k107,c107,err107]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k108,c108,err108]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k109,c109,err109]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k110,c110,err110]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k111,c111,err111]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k112,c112,err112]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k113,c113,err113]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k114,c114,err114]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k115,c115,err115]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k116,c116,err116]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k117,c117,err117]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k118,c118,err118]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k119,c119,err119]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k120,c120,err120]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k121,c121,err121]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k122,c122,err122]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k123,c123,err123]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k124,c124,err124]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k125,c125,err125]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k126,c126,err126]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k127,c127,err127]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k128,c128,err128]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k129,c129,err129]=som_kmeans('batch',W1,5,15,'verbose');toc
tic; [k130,c130,err130]=som_kmeans('batch',W1,5,15,'verbose');toc

kM101=som_map_struct(2,'msize',[1
5],'sheet');kM101.codebook=k101;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM101,sW1)
kM102=som_map_struct(2,'msize',[1
5],'sheet');kM102.codebook=k102;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM102,sW1)
kM103=som_map_struct(2,'msize',[1
5],'sheet');kM103.codebook=k103;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM103,sW1)
kM104=som_map_struct(2,'msize',[1
5],'sheet');kM104.codebook=k104;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM104,sW1)
kM105=som_map_struct(2,'msize',[1
5],'sheet');kM105.codebook=k105;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM105,sW1)
kM106=som_map_struct(2,'msize',[1
5],'sheet');kM106.codebook=k106;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM106,sW1)
kM107=som_map_struct(2,'msize',[1
5],'sheet');kM107.codebook=k107;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM107,sW1)
kM108=som_map_struct(2,'msize',[1
5],'sheet');kM108.codebook=k108;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM108,sW1)
```

```
kM109=som_map_struct(2,'msize',[1
5],'sheet');kM109.codebook=k104;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM109,sW1)
kM110=som_map_struct(2,'msize',[1
5],'sheet');kM110.codebook=k110;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM110,sW1)
kM111=som_map_struct(2,'msize',[1
5],'sheet');kM111.codebook=k111;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM111,sW1)
kM112=som_map_struct(2,'msize',[1
5],'sheet');kM112.codebook=k112;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM112,sW1)
kM113=som_map_struct(2,'msize',[1
5],'sheet');kM113.codebook=k113;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM113,sW1)
kM114=som_map_struct(2,'msize',[1
5],'sheet');kM114.codebook=k114;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM114,sW1)
kM115=som_map_struct(2,'msize',[1
5],'sheet');kM115.codebook=k115;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM115,sW1)
kM116=som_map_struct(2,'msize',[1
5],'sheet');kM116.codebook=k116;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM116,sW1)
kM117=som_map_struct(2,'msize',[1
5],'sheet');kM117.codebook=k117;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM117,sW1)
kM118=som_map_struct(2,'msize',[1
5],'sheet');kM118.codebook=k118;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM118,sW1)
kM119=som_map_struct(2,'msize',[1
5],'sheet');kM119.codebook=k119;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM119,sW1)
kM120=som_map_struct(2,'msize',[1
5],'sheet');kM120.codebook=k120;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM120,sW1)
kM121=som_map_struct(2,'msize',[1
5],'sheet');kM121.codebook=k121;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM121,sW1)
kM122=som_map_struct(2,'msize',[1
5],'sheet');kM122.codebook=k122;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM122,sW1)
kM123=som_map_struct(2,'msize',[1
5],'sheet');kM123.codebook=k123;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM123,sW1)
kM124=som_map_struct(2,'msize',[1
5],'sheet');kM124.codebook=k124;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM124,sW1)
kM125=som_map_struct(2,'msize',[1
5],'sheet');kM125.codebook=k125;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM125,sW1)
kM126=som_map_struct(2,'msize',[1
5],'sheet');kM126.codebook=k126;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM126,sW1)
kM127=som_map_struct(2,'msize',[1
5],'sheet');kM127.codebook=k127;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM127,sW1)
kM128=som_map_struct(2,'msize',[1
5],'sheet');kM128.codebook=k128;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM128,sW1)
kM129=som_map_struct(2,'msize',[1
5],'sheet');kM129.codebook=k129;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM129,sW1)
kM130=som_map_struct(2,'msize',[1
5],'sheet');kM130.codebook=k130;sW1=som_data_struct(W1);[qe,te]=som_quali
ty(kM130,sW1)

%K2
tic; [k201,c201,err201]=som_kmeans('batch',W2,5,15,'verbose');toc
```

```
tic; [k202,c202,err202]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k203,c203,err203]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k204,c204,err204]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k205,c205,err205]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k206,c206,err206]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k207,c207,err207]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k208,c208,err208]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k209,c209,err209]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k210,c210,err210]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k211,c211,err211]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k212,c212,err212]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k213,c213,err213]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k214,c214,err214]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k215,c215,err215]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k216,c216,err216]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k217,c217,err217]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k218,c218,err218]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k219,c219,err219]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k220,c220,err220]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k221,c221,err221]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k222,c222,err222]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k223,c223,err223]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k224,c224,err224]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k225,c225,err225]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k226,c226,err226]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k227,c227,err227]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k228,c228,err228]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k229,c229,err229]=som_kmeans('batch',W2,5,15,'verbose');toc
tic; [k230,c230,err230]=som_kmeans('batch',W2,5,15,'verbose');toc

kM201=som_map_struct(2,'msize',[1
5],'sheet');kM201.codebook=k201;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM201,sW2)
kM202=som_map_struct(2,'msize',[1
5],'sheet');kM202.codebook=k202;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM202,sW2)
kM203=som_map_struct(2,'msize',[1
5],'sheet');kM203.codebook=k203;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM203,sW2)
kM204=som_map_struct(2,'msize',[1
5],'sheet');kM204.codebook=k204;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM204,sW2)
kM205=som_map_struct(2,'msize',[1
5],'sheet');kM205.codebook=k205;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM205,sW2)
kM206=som_map_struct(2,'msize',[1
5],'sheet');kM206.codebook=k206;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM206,sW2)
kM207=som_map_struct(2,'msize',[1
5],'sheet');kM207.codebook=k207;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM207,sW2)
kM208=som_map_struct(2,'msize',[1
5],'sheet');kM208.codebook=k208;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM208,sW2)
kM209=som_map_struct(2,'msize',[1
5],'sheet');kM209.codebook=k204;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM209,sW2)
kM210=som_map_struct(2,'msize',[1
5],'sheet');kM210.codebook=k210;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM210,sW2)
kM211=som_map_struct(2,'msize',[1
5],'sheet');kM211.codebook=k211;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM211,sW2)
kM212=som_map_struct(2,'msize',[1
5],'sheet');kM212.codebook=k212;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM212,sW2)
kM213=som_map_struct(2,'msize',[1
5],'sheet');kM213.codebook=k213;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM213,sW2)
```

```
kM214=som_map_struct(2,'msize',[1
5],'sheet');kM214.codebook=k214;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM214,sW2)
kM215=som_map_struct(2,'msize',[1
5],'sheet');kM215.codebook=k215;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM215,sW2)
kM216=som_map_struct(2,'msize',[1
5],'sheet');kM216.codebook=k216;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM216,sW2)
kM217=som_map_struct(2,'msize',[1
5],'sheet');kM217.codebook=k217;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM217,sW2)
kM218=som_map_struct(2,'msize',[1
5],'sheet');kM218.codebook=k218;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM218,sW2)
kM219=som_map_struct(2,'msize',[1
5],'sheet');kM219.codebook=k219;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM219,sW2)
kM220=som_map_struct(2,'msize',[1
5],'sheet');kM220.codebook=k220;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM220,sW2)
kM221=som_map_struct(2,'msize',[1
5],'sheet');kM221.codebook=k221;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM221,sW2)
kM222=som_map_struct(2,'msize',[1
5],'sheet');kM222.codebook=k222;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM222,sW2)
kM223=som_map_struct(2,'msize',[1
5],'sheet');kM223.codebook=k223;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM223,sW2)
kM224=som_map_struct(2,'msize',[1
5],'sheet');kM224.codebook=k224;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM224,sW2)
kM225=som_map_struct(2,'msize',[1
5],'sheet');kM225.codebook=k225;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM225,sW2)
kM226=som_map_struct(2,'msize',[1
5],'sheet');kM226.codebook=k226;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM226,sW2)
kM227=som_map_struct(2,'msize',[1
5],'sheet');kM227.codebook=k227;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM227,sW2)
kM228=som_map_struct(2,'msize',[1
5],'sheet');kM228.codebook=k228;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM228,sW2)
kM229=som_map_struct(2,'msize',[1
5],'sheet');kM229.codebook=k229;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM229,sW2)
kM230=som_map_struct(2,'msize',[1
5],'sheet');kM230.codebook=k230;sW2=som_data_struct(W2);[qe,te]=som_quali
ty(kM230,sW2)

% K3
tic; [k301,c301,err301]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k302,c302,err302]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k303,c303,err303]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k304,c304,err304]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k305,c305,err305]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k306,c306,err306]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k307,c307,err307]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k308,c308,err308]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k309,c309,err309]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k310,c310,err310]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k311,c311,err311]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k312,c312,err312]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k313,c313,err313]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k314,c314,err314]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k315,c315,err315]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k316,c316,err316]=som_kmeans('batch',W3,5,15,'verbose');toc
```

```
tic; [k317,c317,err317]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k318,c318,err318]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k319,c319,err319]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k320,c320,err320]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k321,c321,err321]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k322,c322,err322]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k323,c323,err323]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k324,c324,err324]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k325,c325,err325]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k326,c326,err326]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k327,c327,err327]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k328,c328,err328]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k329,c329,err329]=som_kmeans('batch',W3,5,15,'verbose');toc
tic; [k330,c330,err330]=som_kmeans('batch',W3,5,15,'verbose');toc

kM301=som_map_struct(2,'msize',[1
5],'sheet');kM301.codebook=k301;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM301,sW3)
kM302=som_map_struct(2,'msize',[1
5],'sheet');kM302.codebook=k302;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM302,sW3)
kM303=som_map_struct(2,'msize',[1
5],'sheet');kM303.codebook=k303;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM303,sW3)
kM304=som_map_struct(2,'msize',[1
5],'sheet');kM304.codebook=k304;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM304,sW3)
kM305=som_map_struct(2,'msize',[1
5],'sheet');kM305.codebook=k305;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM305,sW3)
kM306=som_map_struct(2,'msize',[1
5],'sheet');kM306.codebook=k306;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM306,sW3)
kM307=som_map_struct(2,'msize',[1
5],'sheet');kM307.codebook=k307;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM307,sW3)
kM308=som_map_struct(2,'msize',[1
5],'sheet');kM308.codebook=k308;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM308,sW3)
kM309=som_map_struct(2,'msize',[1
5],'sheet');kM309.codebook=k304;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM309,sW3)
kM310=som_map_struct(2,'msize',[1
5],'sheet');kM310.codebook=k310;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM310,sW3)
kM311=som_map_struct(2,'msize',[1
5],'sheet');kM311.codebook=k311;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM311,sW3)
kM312=som_map_struct(2,'msize',[1
5],'sheet');kM312.codebook=k312;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM312,sW3)
kM313=som_map_struct(2,'msize',[1
5],'sheet');kM313.codebook=k313;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM313,sW3)
kM314=som_map_struct(2,'msize',[1
5],'sheet');kM314.codebook=k314;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM314,sW3)
kM315=som_map_struct(2,'msize',[1
5],'sheet');kM315.codebook=k315;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM315,sW3)
kM316=som_map_struct(2,'msize',[1
5],'sheet');kM316.codebook=k316;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM316,sW3)
kM317=som_map_struct(2,'msize',[1
5],'sheet');kM317.codebook=k317;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM317,sW3)
kM318=som_map_struct(2,'msize',[1
5],'sheet');kM318.codebook=k318;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM318,sW3)
```

```
kM319=som_map_struct(2,'msize',[1
5],'sheet');kM319.codebook=k319;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM319,sW3)
kM320=som_map_struct(2,'msize',[1
5],'sheet');kM320.codebook=k320;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM320,sW3)
kM321=som_map_struct(2,'msize',[1
5],'sheet');kM321.codebook=k321;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM321,sW3)
kM322=som_map_struct(2,'msize',[1
5],'sheet');kM322.codebook=k322;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM322,sW3)
kM323=som_map_struct(2,'msize',[1
5],'sheet');kM323.codebook=k323;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM323,sW3)
kM324=som_map_struct(2,'msize',[1
5],'sheet');kM324.codebook=k324;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM324,sW3)
kM325=som_map_struct(2,'msize',[1
5],'sheet');kM325.codebook=k325;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM325,sW3)
kM326=som_map_struct(2,'msize',[1
5],'sheet');kM326.codebook=k326;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM326,sW3)
kM327=som_map_struct(2,'msize',[1
5],'sheet');kM327.codebook=k327;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM327,sW3)
kM328=som_map_struct(2,'msize',[1
5],'sheet');kM328.codebook=k328;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM328,sW3)
kM329=som_map_struct(2,'msize',[1
5],'sheet');kM329.codebook=k329;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM329,sW3)
kM330=som_map_struct(2,'msize',[1
5],'sheet');kM330.codebook=k330;sW3=som_data_struct(W3);[qe,te]=som_quali
ty(kM330,sW3)

% K4
tic; [k401,c401,err401]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k402,c402,err402]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k403,c403,err403]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k404,c404,err404]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k405,c405,err405]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k406,c406,err406]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k407,c407,err407]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k408,c408,err408]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k409,c409,err409]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k410,c410,err410]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k411,c411,err411]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k412,c412,err412]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k413,c413,err413]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k414,c414,err414]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k415,c415,err415]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k416,c416,err146]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k417,c417,err417]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k418,c418,err418]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k419,c419,err419]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k420,c420,err420]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k421,c421,err421]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k422,c422,err422]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k423,c423,err423]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k424,c424,err424]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k425,c425,err425]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k426,c426,err426]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k427,c427,err427]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k428,c428,err428]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k429,c429,err429]=som_kmeans('batch',W1,10,15,'verbose');toc
tic; [k430,c430,err430]=som_kmeans('batch',W1,10,15,'verbose');toc
```

```
kM401=som_map_struct(2,'msize',[1
10],'sheet');kM401.codebook=k401;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM401,sW1)
kM402=som_map_struct(2,'msize',[1
10],'sheet');kM402.codebook=k402;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM402,sW1)
kM403=som_map_struct(2,'msize',[1
10],'sheet');kM403.codebook=k403;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM403,sW1)
kM404=som_map_struct(2,'msize',[1
10],'sheet');kM404.codebook=k404;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM404,sW1)
kM405=som_map_struct(2,'msize',[1
10],'sheet');kM405.codebook=k405;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM405,sW1)
kM406=som_map_struct(2,'msize',[1
10],'sheet');kM406.codebook=k406;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM406,sW1)
kM407=som_map_struct(2,'msize',[1
10],'sheet');kM407.codebook=k407;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM407,sW1)
kM408=som_map_struct(2,'msize',[1
10],'sheet');kM408.codebook=k408;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM408,sW1)
kM409=som_map_struct(2,'msize',[1
10],'sheet');kM409.codebook=k404;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM409,sW1)
kM410=som_map_struct(2,'msize',[1
10],'sheet');kM410.codebook=k410;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM410,sW1)
kM411=som_map_struct(2,'msize',[1
10],'sheet');kM411.codebook=k411;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM411,sW1)
kM412=som_map_struct(2,'msize',[1
10],'sheet');kM412.codebook=k412;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM412,sW1)
kM413=som_map_struct(2,'msize',[1
10],'sheet');kM413.codebook=k413;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM413,sW1)
kM414=som_map_struct(2,'msize',[1
10],'sheet');kM414.codebook=k414;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM414,sW1)
kM415=som_map_struct(2,'msize',[1
10],'sheet');kM415.codebook=k415;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM415,sW1)
kM416=som_map_struct(2,'msize',[1
10],'sheet');kM416.codebook=k416;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM416,sW1)
kM417=som_map_struct(2,'msize',[1
10],'sheet');kM417.codebook=k417;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM417,sW1)
kM418=som_map_struct(2,'msize',[1
10],'sheet');kM418.codebook=k418;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM418,sW1)
kM419=som_map_struct(2,'msize',[1
10],'sheet');kM419.codebook=k419;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM419,sW1)
kM420=som_map_struct(2,'msize',[1
10],'sheet');kM420.codebook=k420;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM420,sW1)
kM421=som_map_struct(2,'msize',[1
10],'sheet');kM421.codebook=k421;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM421,sW1)
kM422=som_map_struct(2,'msize',[1
10],'sheet');kM422.codebook=k422;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM422,sW1)
kM423=som_map_struct(2,'msize',[1
10],'sheet');kM423.codebook=k423;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM423,sW1)
```

```
kM424=som_map_struct(2,'msize',[1
10],'sheet');kM424.codebook=k424;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM424,sW1)
kM425=som_map_struct(2,'msize',[1
10],'sheet');kM425.codebook=k425;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM425,sW1)
kM426=som_map_struct(2,'msize',[1
10],'sheet');kM426.codebook=k426;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM426,sW1)
kM427=som_map_struct(2,'msize',[1
10],'sheet');kM427.codebook=k427;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM427,sW1)
kM428=som_map_struct(2,'msize',[1
10],'sheet');kM428.codebook=k428;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM428,sW1)
kM429=som_map_struct(2,'msize',[1
10],'sheet');kM429.codebook=k429;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM429,sW1)
kM430=som_map_struct(2,'msize',[1
10],'sheet');kM430.codebook=k430;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM430,sW1)

% K5
tic; [k501,c501,err501]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k502,c502,err502]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k503,c503,err503]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k504,c504,err504]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k505,c505,err505]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k506,c506,err506]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k507,c507,err507]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k508,c508,err508]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k509,c509,err509]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k510,c510,err510]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k511,c511,err511]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k512,c512,err512]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k513,c513,err513]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k514,c514,err514]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k515,c515,err515]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k516,c516,err516]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k517,c517,err517]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k518,c518,err518]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k519,c519,err519]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k520,c520,err520]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k521,c521,err521]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k522,c522,err522]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k523,c523,err523]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k524,c524,err524]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k525,c525,err525]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k526,c526,err526]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k527,c527,err527]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k528,c528,err528]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k529,c529,err529]=som_kmeans('batch',W2,10,15,'verbose');toc
tic; [k530,c530,err530]=som_kmeans('batch',W2,10,15,'verbose');toc

kM501=som_map_struct(2,'msize',[1
10],'sheet');kM501.codebook=k501;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM501,sW2)
kM502=som_map_struct(2,'msize',[1
10],'sheet');kM502.codebook=k502;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM502,sW2)
kM503=som_map_struct(2,'msize',[1
10],'sheet');kM503.codebook=k503;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM503,sW2)
kM504=som_map_struct(2,'msize',[1
10],'sheet');kM504.codebook=k504;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM504,sW2)
kM505=som_map_struct(2,'msize',[1
10],'sheet');kM505.codebook=k505;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM505,sW2)
```

```
kM506=som_map_struct(2,'msize',[1
10],'sheet');kM506.codebook=k506;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM506,sW2)
kM507=som_map_struct(2,'msize',[1
10],'sheet');kM507.codebook=k507;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM507,sW2)
kM508=som_map_struct(2,'msize',[1
10],'sheet');kM508.codebook=k508;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM508,sW2)
kM509=som_map_struct(2,'msize',[1
10],'sheet');kM509.codebook=k504;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM509,sW2)
kM510=som_map_struct(2,'msize',[1
10],'sheet');kM510.codebook=k510;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM510,sW2)
kM511=som_map_struct(2,'msize',[1
10],'sheet');kM511.codebook=k511;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM511,sW2)
kM512=som_map_struct(2,'msize',[1
10],'sheet');kM512.codebook=k512;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM512,sW2)
kM513=som_map_struct(2,'msize',[1
10],'sheet');kM513.codebook=k513;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM513,sW2)
kM514=som_map_struct(2,'msize',[1
10],'sheet');kM514.codebook=k514;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM514,sW2)
kM515=som_map_struct(2,'msize',[1
10],'sheet');kM515.codebook=k515;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM515,sW2)
kM516=som_map_struct(2,'msize',[1
10],'sheet');kM516.codebook=k516;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM516,sW2)
kM517=som_map_struct(2,'msize',[1
10],'sheet');kM517.codebook=k517;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM517,sW2)
kM518=som_map_struct(2,'msize',[1
10],'sheet');kM518.codebook=k518;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM518,sW2)
kM519=som_map_struct(2,'msize',[1
10],'sheet');kM519.codebook=k519;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM519,sW2)
kM520=som_map_struct(2,'msize',[1
10],'sheet');kM520.codebook=k520;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM520,sW2)
kM521=som_map_struct(2,'msize',[1
10],'sheet');kM521.codebook=k521;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM521,sW2)
kM522=som_map_struct(2,'msize',[1
10],'sheet');kM522.codebook=k522;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM522,sW2)
kM523=som_map_struct(2,'msize',[1
10],'sheet');kM523.codebook=k523;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM523,sW2)
kM524=som_map_struct(2,'msize',[1
10],'sheet');kM524.codebook=k524;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM524,sW2)
kM525=som_map_struct(2,'msize',[1
10],'sheet');kM525.codebook=k525;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM525,sW2)
kM526=som_map_struct(2,'msize',[1
10],'sheet');kM526.codebook=k526;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM526,sW2)
kM527=som_map_struct(2,'msize',[1
10],'sheet');kM527.codebook=k527;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM527,sW2)
kM528=som_map_struct(2,'msize',[1
10],'sheet');kM528.codebook=k528;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM528,sW2)
```

```
kM529=som_map_struct(2,'msize',[1
10],'sheet');kM529.codebook=k529;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM529,sW2)
kM530=som_map_struct(2,'msize',[1
10],'sheet');kM530.codebook=k530;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM530,sW2)

% K6
tic; [k601,c601,err601]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k602,c602,err602]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k603,c603,err603]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k604,c604,err604]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k605,c605,err605]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k606,c606,err606]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k607,c607,err607]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k608,c608,err608]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k609,c609,err609]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k610,c610,err610]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k611,c611,err611]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k612,c612,err612]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k613,c613,err613]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k614,c614,err614]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k615,c615,err615]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k616,c616,err616]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k617,c617,err617]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k618,c618,err618]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k619,c619,err619]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k620,c620,err620]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k621,c621,err621]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k622,c622,err622]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k623,c623,err623]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k624,c624,err624]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k625,c625,err625]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k626,c626,err626]=som_kmeans('batch',W3,10,15,'verbose');toc
tic; [k627,c627,err627]=som_kmeans('batch',W3,10,15,'verbose');toc

kM601=som_map_struct(2,'msize',[1
10],'sheet');kM601.codebook=k601;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM601,sW3)
kM602=som_map_struct(2,'msize',[1
10],'sheet');kM602.codebook=k602;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM602,sW3)
kM603=som_map_struct(2,'msize',[1
10],'sheet');kM603.codebook=k603;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM603,sW3)
kM604=som_map_struct(2,'msize',[1
10],'sheet');kM604.codebook=k604;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM604,sW3)
kM605=som_map_struct(2,'msize',[1
10],'sheet');kM605.codebook=k605;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM605,sW3)
kM606=som_map_struct(2,'msize',[1
10],'sheet');kM606.codebook=k606;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM606,sW3)
kM607=som_map_struct(2,'msize',[1
10],'sheet');kM607.codebook=k607;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM607,sW3)
kM608=som_map_struct(2,'msize',[1
10],'sheet');kM608.codebook=k608;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM608,sW3)
kM609=som_map_struct(2,'msize',[1
10],'sheet');kM609.codebook=k604;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM609,sW3)
kM610=som_map_struct(2,'msize',[1
10],'sheet');kM610.codebook=k610;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM610,sW3)
kM611=som_map_struct(2,'msize',[1
10],'sheet');kM611.codebook=k611;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM611,sW3)
```

```
kM612=som_map_struct(2,'msize',[1
10],'sheet');kM612.codebook=k612;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM612,sW3)
kM613=som_map_struct(2,'msize',[1
10],'sheet');kM613.codebook=k613;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM613,sW3)
kM614=som_map_struct(2,'msize',[1
10],'sheet');kM614.codebook=k614;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM614,sW3)
kM615=som_map_struct(2,'msize',[1
10],'sheet');kM615.codebook=k615;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM615,sW3)
kM616=som_map_struct(2,'msize',[1
10],'sheet');kM616.codebook=k616;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM616,sW3)
kM617=som_map_struct(2,'msize',[1
10],'sheet');kM617.codebook=k617;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM617,sW3)
kM618=som_map_struct(2,'msize',[1
10],'sheet');kM618.codebook=k618;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM618,sW3)
kM619=som_map_struct(2,'msize',[1
10],'sheet');kM619.codebook=k619;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM619,sW3)
kM620=som_map_struct(2,'msize',[1
10],'sheet');kM620.codebook=k620;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM620,sW3)
kM621=som_map_struct(2,'msize',[1
10],'sheet');kM621.codebook=k621;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM621,sW3)
kM622=som_map_struct(2,'msize',[1
10],'sheet');kM622.codebook=k622;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM622,sW3)
kM623=som_map_struct(2,'msize',[1
10],'sheet');kM623.codebook=k623;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM623,sW3)
kM624=som_map_struct(2,'msize',[1
10],'sheet');kM624.codebook=k624;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM624,sW3)
kM625=som_map_struct(2,'msize',[1
10],'sheet');kM625.codebook=k625;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM625,sW3)
kM626=som_map_struct(2,'msize',[1
10],'sheet');kM626.codebook=k626;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM626,sW3)
kM627=som_map_struct(2,'msize',[1
10],'sheet');kM627.codebook=k627;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM627,sW3)
kM628=som_map_struct(2,'msize',[1
10],'sheet');kM628.codebook=k628;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM628,sW3)
kM629=som_map_struct(2,'msize',[1
10],'sheet');kM629.codebook=k629;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM629,sW3)
kM630=som_map_struct(2,'msize',[1
10],'sheet');kM630.codebook=k630;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM630,sW3)

% K7
tic; [k701,c701,err701]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k702,c702,err702]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k703,c703,err703]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k704,c704,err704]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k705,c705,err705]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k706,c706,err706]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k707,c707,err707]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k708,c708,err708]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k709,c709,err709]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k710,c710,err710]=som_kmeans('batch',W1,15,15,'verbose');toc
```

```
tic; [k711,c711,err711]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k712,c712,err712]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k713,c713,err713]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k717,c717,err717]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k718,c718,err718]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k719,c719,err719]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k720,c720,err720]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k721,c721,err721]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k722,c722,err722]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k723,c723,err723]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k724,c724,err724]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k725,c725,err725]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k726,c726,err726]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k727,c727,err727]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k728,c728,err728]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k729,c729,err729]=som_kmeans('batch',W1,15,15,'verbose');toc
tic; [k730,c730,err730]=som_kmeans('batch',W1,15,15,'verbose');toc

kM701=som_map_struct(2,'msize',[1
15],'sheet');kM701.codebook=k701;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM701,sW1)
kM702=som_map_struct(2,'msize',[1
15],'sheet');kM702.codebook=k702;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM702,sW1)
kM703=som_map_struct(2,'msize',[1
15],'sheet');kM703.codebook=k703;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM703,sW1)
kM704=som_map_struct(2,'msize',[1
15],'sheet');kM704.codebook=k704;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM704,sW1)
kM705=som_map_struct(2,'msize',[1
15],'sheet');kM705.codebook=k705;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM705,sW1)
kM706=som_map_struct(2,'msize',[1
15],'sheet');kM706.codebook=k706;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM706,sW1)
kM707=som_map_struct(2,'msize',[1
15],'sheet');kM707.codebook=k707;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM707,sW1)
kM708=som_map_struct(2,'msize',[1
15],'sheet');kM708.codebook=k708;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM708,sW1)
kM709=som_map_struct(2,'msize',[1
15],'sheet');kM709.codebook=k704;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM709,sW1)
kM710=som_map_struct(2,'msize',[1
15],'sheet');kM710.codebook=k710;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM710,sW1)
kM711=som_map_struct(2,'msize',[1
15],'sheet');kM711.codebook=k711;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM711,sW1)
kM712=som_map_struct(2,'msize',[1
15],'sheet');kM712.codebook=k712;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM712,sW1)
kM713=som_map_struct(2,'msize',[1
15],'sheet');kM713.codebook=k713;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM713,sW1)
kM714=som_map_struct(2,'msize',[1
15],'sheet');kM714.codebook=k714;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM714,sW1)
kM715=som_map_struct(2,'msize',[1
15],'sheet');kM715.codebook=k715;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM715,sW1)
kM716=som_map_struct(2,'msize',[1
15],'sheet');kM716.codebook=k716;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM716,sW1)
kM717=som_map_struct(2,'msize',[1
15],'sheet');kM717.codebook=k717;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM717,sW1)
```

```
kM718=som_map_struct(2,'msize',[1
15],'sheet');kM718.codebook=k718;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM718,sW1)
kM719=som_map_struct(2,'msize',[1
15],'sheet');kM719.codebook=k719;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM719,sW1)
kM720=som_map_struct(2,'msize',[1
15],'sheet');kM720.codebook=k720;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM720,sW1)
kM721=som_map_struct(2,'msize',[1
15],'sheet');kM721.codebook=k721;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM721,sW1)
kM722=som_map_struct(2,'msize',[1
15],'sheet');kM722.codebook=k722;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM722,sW1)
kM723=som_map_struct(2,'msize',[1
15],'sheet');kM723.codebook=k723;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM723,sW1)
kM724=som_map_struct(2,'msize',[1
15],'sheet');kM724.codebook=k724;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM724,sW1)
kM725=som_map_struct(2,'msize',[1
15],'sheet');kM725.codebook=k725;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM725,sW1)
kM726=som_map_struct(2,'msize',[1
15],'sheet');kM726.codebook=k726;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM726,sW1)
kM727=som_map_struct(2,'msize',[1
15],'sheet');kM727.codebook=k727;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM727,sW1)
kM728=som_map_struct(2,'msize',[1
15],'sheet');kM728.codebook=k728;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM728,sW1)
kM729=som_map_struct(2,'msize',[1
15],'sheet');kM729.codebook=k729;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM729,sW1)
kM730=som_map_struct(2,'msize',[1
15],'sheet');kM730.codebook=k730;sW1=som_data_struct(W1);[qe,te]=som_qual
ity(kM730,sW1)

% K8
tic; [k801,c801,err801]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k802,c802,err802]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k803,c803,err803]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k804,c804,err804]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k805,c805,err805]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k806,c806,err806]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k807,c807,err807]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k808,c808,err808]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k809,c809,err809]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k810,c810,err810]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k811,c811,err811]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k812,c812,err812]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k813,c813,err813]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k814,c814,err814]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k815,c815,err815]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k816,c816,err816]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k817,c817,err817]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k818,c818,err818]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k819,c819,err819]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k820,c820,err820]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k821,c821,err821]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k822,c822,err822]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k823,c823,err823]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k824,c824,err824]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k825,c825,err825]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k826,c826,err826]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k827,c827,err827]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k828,c828,err828]=som_kmeans('batch',W2,15,15,'verbose');toc
```

```
tic; [k829,c829,err829]=som_kmeans('batch',W2,15,15,'verbose');toc
tic; [k830,c830,err830]=som_kmeans('batch',W2,15,15,'verbose');toc

kM801=som_map_struct(2,'msize',[1
15],'sheet');kM801.codebook=k801;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM801,sW2)
kM802=som_map_struct(2,'msize',[1
15],'sheet');kM802.codebook=k802;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM802,sW2)
kM803=som_map_struct(2,'msize',[1
15],'sheet');kM803.codebook=k803;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM803,sW2)
kM804=som_map_struct(2,'msize',[1
15],'sheet');kM804.codebook=k804;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM804,sW2)
kM805=som_map_struct(2,'msize',[1
15],'sheet');kM805.codebook=k805;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM805,sW2)
kM806=som_map_struct(2,'msize',[1
15],'sheet');kM806.codebook=k806;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM806,sW2)
kM807=som_map_struct(2,'msize',[1
15],'sheet');kM807.codebook=k807;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM807,sW2)
kM808=som_map_struct(2,'msize',[1
15],'sheet');kM808.codebook=k808;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM808,sW2)
kM809=som_map_struct(2,'msize',[1
15],'sheet');kM809.codebook=k804;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM809,sW2)
kM810=som_map_struct(2,'msize',[1
15],'sheet');kM810.codebook=k810;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM810,sW2)
kM811=som_map_struct(2,'msize',[1
15],'sheet');kM811.codebook=k811;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM811,sW2)
kM812=som_map_struct(2,'msize',[1
15],'sheet');kM812.codebook=k812;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM812,sW2)
kM813=som_map_struct(2,'msize',[1
15],'sheet');kM813.codebook=k813;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM813,sW2)
kM814=som_map_struct(2,'msize',[1
15],'sheet');kM814.codebook=k814;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM814,sW2)
kM815=som_map_struct(2,'msize',[1
15],'sheet');kM815.codebook=k815;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM815,sW2)
kM816=som_map_struct(2,'msize',[1
15],'sheet');kM816.codebook=k816;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM816,sW2)
kM817=som_map_struct(2,'msize',[1
15],'sheet');kM817.codebook=k817;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM817,sW2)
kM818=som_map_struct(2,'msize',[1
15],'sheet');kM818.codebook=k818;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM818,sW2)
kM819=som_map_struct(2,'msize',[1
15],'sheet');kM819.codebook=k819;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM819,sW2)
kM820=som_map_struct(2,'msize',[1
15],'sheet');kM820.codebook=k820;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM820,sW2)
kM821=som_map_struct(2,'msize',[1
15],'sheet');kM821.codebook=k821;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM821,sW2)
kM822=som_map_struct(2,'msize',[1
15],'sheet');kM822.codebook=k822;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM822,sW2)
```

```
kM823=som_map_struct(2,'msize',[1
15],'sheet');kM823.codebook=k823;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM823,sW2)
kM824=som_map_struct(2,'msize',[1
15],'sheet');kM824.codebook=k824;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM824,sW2)
kM825=som_map_struct(2,'msize',[1
15],'sheet');kM825.codebook=k825;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM825,sW2)
kM826=som_map_struct(2,'msize',[1
15],'sheet');kM826.codebook=k826;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM826,sW2)
kM827=som_map_struct(2,'msize',[1
15],'sheet');kM827.codebook=k827;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM827,sW2)
kM828=som_map_struct(2,'msize',[1
15],'sheet');kM828.codebook=k828;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM828,sW2)
kM829=som_map_struct(2,'msize',[1
15],'sheet');kM829.codebook=k829;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM829,sW2)
kM830=som_map_struct(2,'msize',[1
15],'sheet');kM830.codebook=k830;sW2=som_data_struct(W2);[qe,te]=som_qual
ity(kM830,sW2)

% K9
tic; [k901,c901,err901]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k902,c902,err902]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k903,c903,err903]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k904,c904,err904]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k905,c905,err905]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k906,c906,err906]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k907,c907,err907]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k908,c908,err908]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k909,c909,err909]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k910,c910,err910]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k911,c911,err911]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k912,c912,err912]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k913,c913,err913]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k914,c914,err914]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k915,c915,err915]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k916,c916,err916]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k917,c917,err917]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k918,c918,err918]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k919,c919,err919]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k920,c920,err920]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k921,c921,err921]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k922,c922,err922]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k923,c923,err923]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k924,c924,err924]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k925,c925,err925]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k926,c926,err926]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k927,c927,err927]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k928,c928,err928]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k929,c929,err929]=som_kmeans('batch',W3,15,15,'verbose');toc
tic; [k930,c930,err930]=som_kmeans('batch',W3,15,15,'verbose');toc

kM901=som_map_struct(2,'msize',[1
15],'sheet');kM901.codebook=k901;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM901,sW3)
kM902=som_map_struct(2,'msize',[1
15],'sheet');kM902.codebook=k902;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM902,sW3)
kM903=som_map_struct(2,'msize',[1
15],'sheet');kM903.codebook=k903;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM903,sW3)
kM904=som_map_struct(2,'msize',[1
15],'sheet');kM904.codebook=k904;sW3=som_data_struct(W3);[qe,te]=som_qual
ity(kM904,sW3)
```

```
kM905=som_map_struct(2,'msize',[1 15],'sheet'); kM905.codebook=k905;
sW3=som_data_struct(W3); [qe,te]=som_quality(kM905,sW3)
kM906=som_map_struct(2,'msize',[1 15],'sheet'); kM906.codebook=k906;
sW3=som_data_struct(W3); [qe,te]=som_quality(kM906,sW3)
kM907=som_map_struct(2,'msize',[1 15],'sheet'); kM907.codebook=k907;
sW3=som_data_struct(W3); [qe,te]=som_quality(kM907,sW3)
kM908=som_map_struct(2,'msize',[1 15],'sheet'); kM908.codebook=k908;
sW3=som_data_struct(W3); [qe,te]=som_quality(kM908,sW3)
kM909=som_map_struct(2,'msize',[1 15],'sheet'); kM909.codebook=k904;
sW3=som_data_struct(W3); [qe,te]=som_quality(kM909,sW3)
kM910=som_map_struct(2,'msize',[1 15],'sheet'); kM910.codebook=k910;
sW3=som_data_struct(W3); [qe,te]=som_quality(kM910,sW3)
kM911=som_map_struct(2,'msize',[1 15],'sheet'); kM911.codebook=k911;
sW3=som_data_struct(W3); [qe,te]=som_quality(kM911,sW3)
kM912=som_map_struct(2,'msize',[1 15],'sheet'); kM912.codebook=k912;
sW3=som_data_struct(W3); [qe,te]=som_quality(kM912,sW3)
kM913=som_map_struct(2,'msize',[1 15],'sheet'); kM913.codebook=k913;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM913,sW3)
kM914=som_map_struct(2,'msize',[1 15],'sheet'); kM914.codebook=k914;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM914,sW3)
kM915=som_map_struct(2,'msize',[1 15],'sheet'); kM915.codebook=k915;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM915,sW3)
kM916=som_map_struct(2,'msize',[1 15],'sheet'); kM916.codebook=k916;
sW3=som_data_struct(W3); [qe,te]=som_quality(kM916,sW3)
kM917=som_map_struct(2,'msize',[1 15],'sheet'); kM917.codebook=k917;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM917,sW3)
kM918=som_map_struct(2,'msize',[1 15],'sheet'); kM918.codebook=k918;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM918,sW3)
kM919=som_map_struct(2,'msize',[1 15],'sheet'); kM919.codebook=k919;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM919,sW3)
kM920=som_map_struct(2,'msize',[1 15],'sheet'); kM920.codebook=k920;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM920,sW3)
kM921=som_map_struct(2,'msize',[1 15],'sheet'); kM921.codebook=k921;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM921,sW3)
kM922=som_map_struct(2,'msize',[1 15],'sheet'); kM922.codebook=k922;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM922,sW3)
kM923=som_map_struct(2,'msize',[1 15],'sheet'); kM923.codebook=k923;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM923,sW3)
kM924=som_map_struct(2,'msize',[1 15],'sheet'); kM924.codebook=k924;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM924,sW3)
kM925=som_map_struct(2,'msize',[1 15],'sheet'); kM925.codebook=k925;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM925,sW3)
kM926=som_map_struct(2,'msize',[1 15],'sheet'); kM926.codebook=k926;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM926,sW3)
kM927=som_map_struct(2,'msize',[1 15],'sheet'); kM927.codebook=k927;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM927,sW3)
kM928=som_map_struct(2,'msize',[1 15],'sheet'); kM928.codebook=k928;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM928,sW3)
kM929=som_map_struct(2,'msize',[1 15],'sheet'); kM929.codebook=k929;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM929,sW3)
kM930=som_map_struct(2,'msize',[1 15],'sheet'); kM930.codebook=k930;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM930,sW3)

% k9A
tic; [k9A01,c901,err901]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A02,c902,err902]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A03,c903,err903]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A04,c904,err904]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A05,c905,err905]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A06,c906,err906]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A07,c907,err907]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A08,c908,err908]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A09,c909,err909]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A10,c910,err910]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A11,c911,err911]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A12,c912,err912]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A13,c913,err913]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A14,c914,err914]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A15,c915,err915]=som_kmeans('batch',W3,15,250,'verbose');toc
```

```
tic; [k9A16,c916,err916]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A17,c917,err917]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A18,c918,err918]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A19,c919,err919]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A20,c920,err920]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A21,c921,err921]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A22,c922,err922]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A23,c923,err923]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A24,c924,err924]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A25,c925,err925]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A26,c926,err926]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A27,c927,err927]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A28,c928,err928]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A29,c929,err929]=som_kmeans('batch',W3,15,250,'verbose');toc
tic; [k9A30,c930,err930]=som_kmeans('batch',W3,15,250,'verbose');toc

kM9A01=som_map_struct(2,'msize',[1 15],'sheet');kM9A01.codebook=k9A01;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A01,sW3)
kM9A02=som_map_struct(2,'msize',[1 15],'sheet');kM9A02.codebook=k9A02;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A02,sW3)
kM9A03=som_map_struct(2,'msize',[1 15],'sheet');kM9A03.codebook=k9A03;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A03,sW3)
kM9A04=som_map_struct(2,'msize',[1 15],'sheet');kM9A04.codebook=k9A04;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A04,sW3)
kM9A05=som_map_struct(2,'msize',[1 15],'sheet');kM9A05.codebook=k9A05;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A05,sW3)
kM9A06=som_map_struct(2,'msize',[1 15],'sheet');kM9A06.codebook=k9A06;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A06,sW3)
kM9A07=som_map_struct(2,'msize',[1 15],'sheet');kM9A07.codebook=k9A07;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A07,sW3)
kM9A08=som_map_struct(2,'msize',[1 15],'sheet');kM9A08.codebook=k9A08;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A08,sW3)
kM9A09=som_map_struct(2,'msize',[1 15],'sheet');kM9A09.codebook=k9A04;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A09,sW3)
kM9A10=som_map_struct(2,'msize',[1 15],'sheet');kM9A10.codebook=k9A10;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A10,sW3)
kM9A11=som_map_struct(2,'msize',[1 15],'sheet');kM9A11.codebook=k9A11;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A11,sW3)
kM9A12=som_map_struct(2,'msize',[1 15],'sheet');kM9A12.codebook=k9A12;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A12,sW3)
kM9A13=som_map_struct(2,'msize',[1 15],'sheet');kM9A13.codebook=k9A13;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A13,sW3)
kM9A14=som_map_struct(2,'msize',[1 15],'sheet');kM9A14.codebook=k9A14;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A14,sW3)
kM9A15=som_map_struct(2,'msize',[1 15],'sheet');kM9A15.codebook=k9A15;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A15,sW3)
kM9A16=som_map_struct(2,'msize',[1 15],'sheet');kM9A16.codebook=k9A16;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A16,sW3)
kM9A17=som_map_struct(2,'msize',[1 15],'sheet');kM9A17.codebook=k9A17;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A17,sW3)
kM9A18=som_map_struct(2,'msize',[1 15],'sheet');kM9A18.codebook=k9A18;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A18,sW3)
kM9A19=som_map_struct(2,'msize',[1 15],'sheet');kM9A19.codebook=k9A19;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A19,sW3)
kM9A20=som_map_struct(2,'msize',[1 15],'sheet');kM9A20.codebook=k9A20;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A20,sW3)
kM9A21=som_map_struct(2,'msize',[1 15],'sheet');kM9A21.codebook=k9A21;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A21,sW3)
kM9A22=som_map_struct(2,'msize',[1 15],'sheet');kM9A22.codebook=k9A22;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A22,sW3)
kM9A23=som_map_struct(2,'msize',[1 15],'sheet');kM9A23.codebook=k9A23;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A23,sW3)
kM9A24=som_map_struct(2,'msize',[1 15],'sheet');kM9A24.codebook=k9A24;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A24,sW3)
kM9A25=som_map_struct(2,'msize',[1 15],'sheet');kM9A25.codebook=k9A25;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A25,sW3)
kM9A26=som_map_struct(2,'msize',[1 15],'sheet');kM9A26.codebook=k9A26;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A26,sW3)
```

```
kM9A27=som_map_struct(2,'msize',[1 15],'sheet');kM9A27.codebook=k9A27;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A27,sW3)
kM9A28=som_map_struct(2,'msize',[1 15],'sheet');kM9A28.codebook=k9A28;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A28,sW3)
kM9A29=som_map_struct(2,'msize',[1 15],'sheet');kM9A29.codebook=k9A29;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A29,sW3)
kM9A30=som_map_struct(2,'msize',[1 15],'sheet');kM9A30.codebook=k9A30;
sW3=som_data_struct(W3);[qe,te]=som_quality(kM9A30,sW3)
```

Appendix III.

CCA-treated wood waste remediation units location

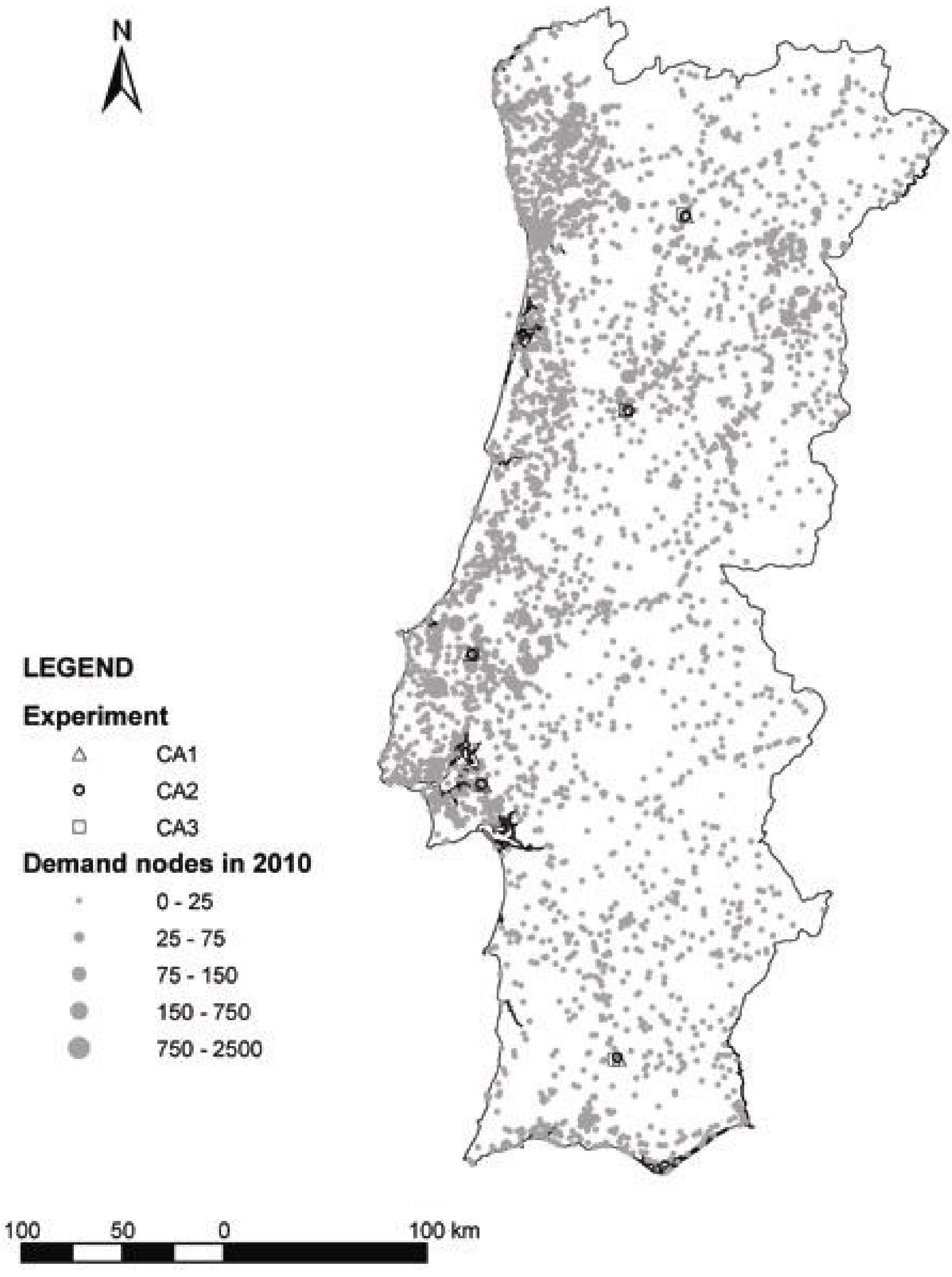

Figure III. 1 – Results from Test 1, for 2010 and considering 5 remediation units.

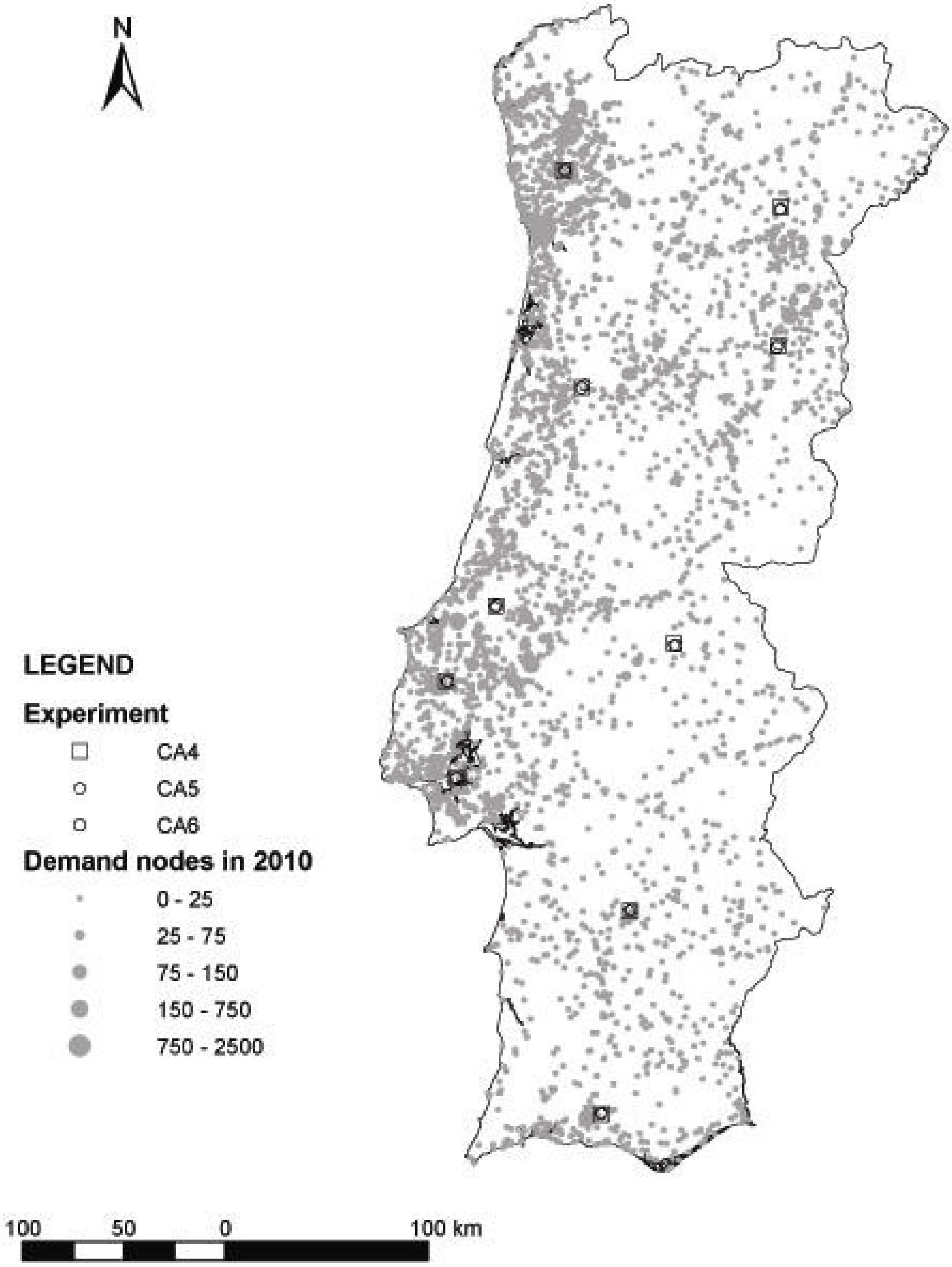

Figure III. 2 – Results from Test 1, for 2010 and considering 10 remediation units.

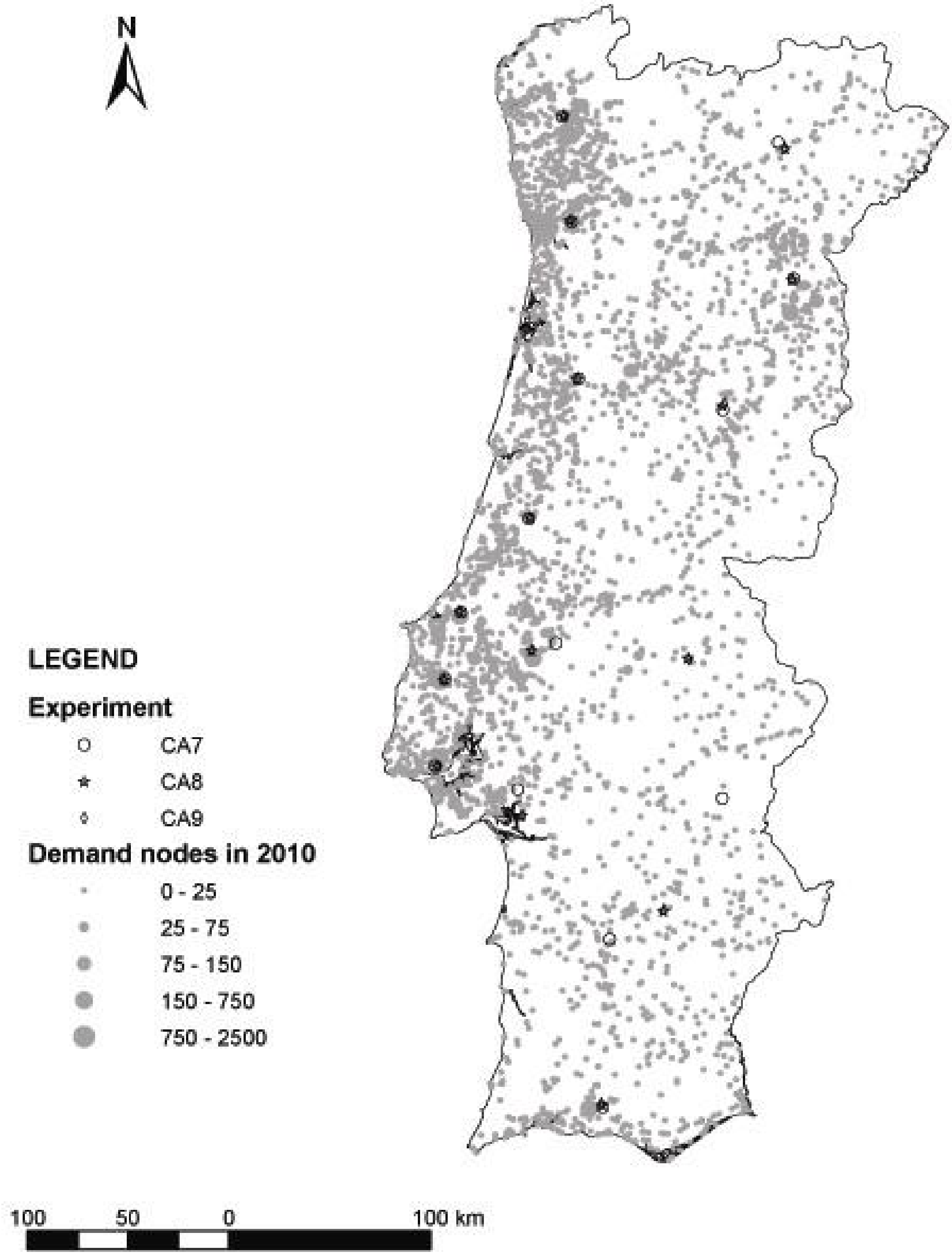

Figure III. 3 – Results from Test 1, for 2010 and considering 15 remediation units.

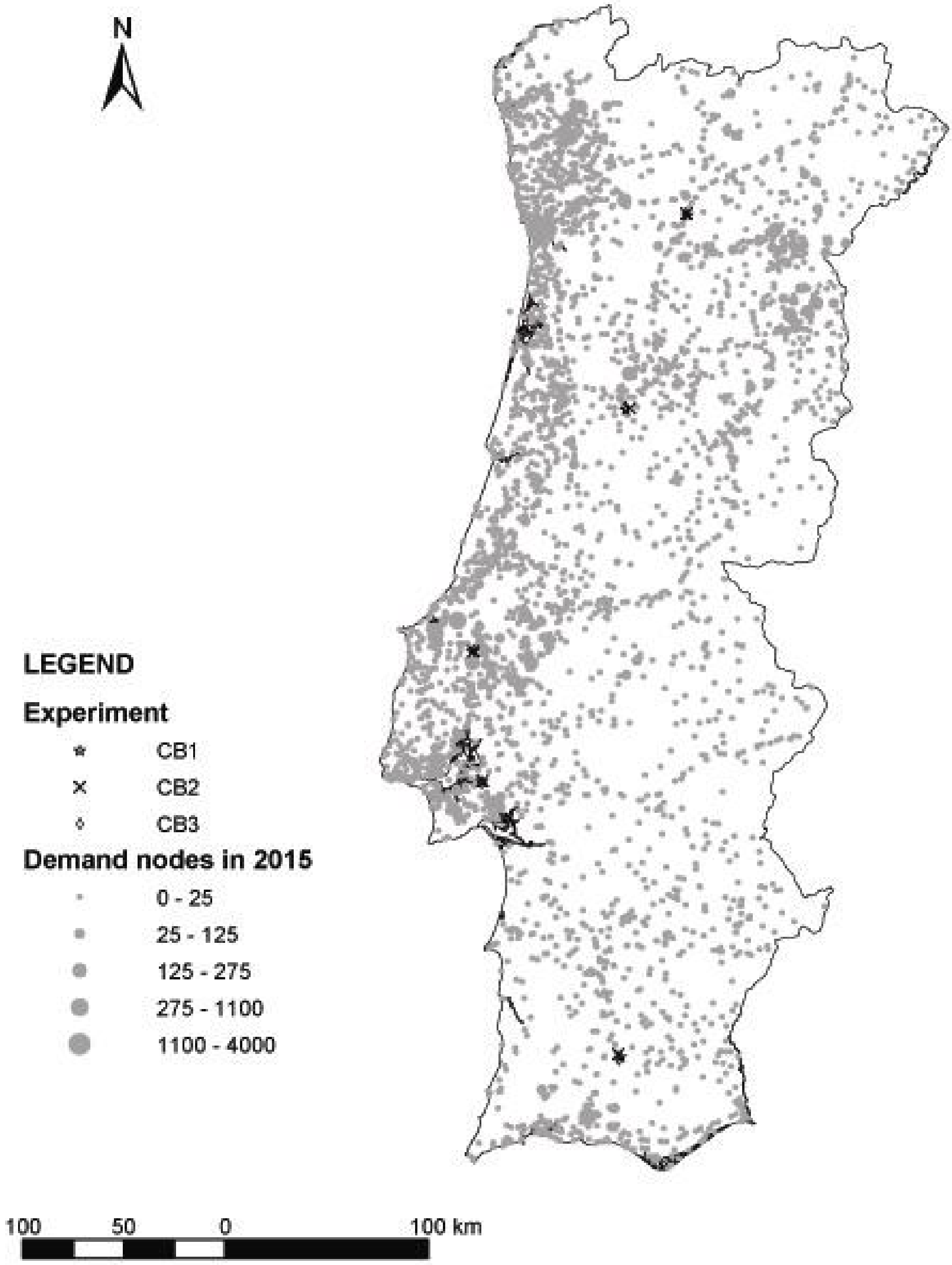

Figure III. 4 – Results from Test 1, for 2015 and considering 5 remediation units.

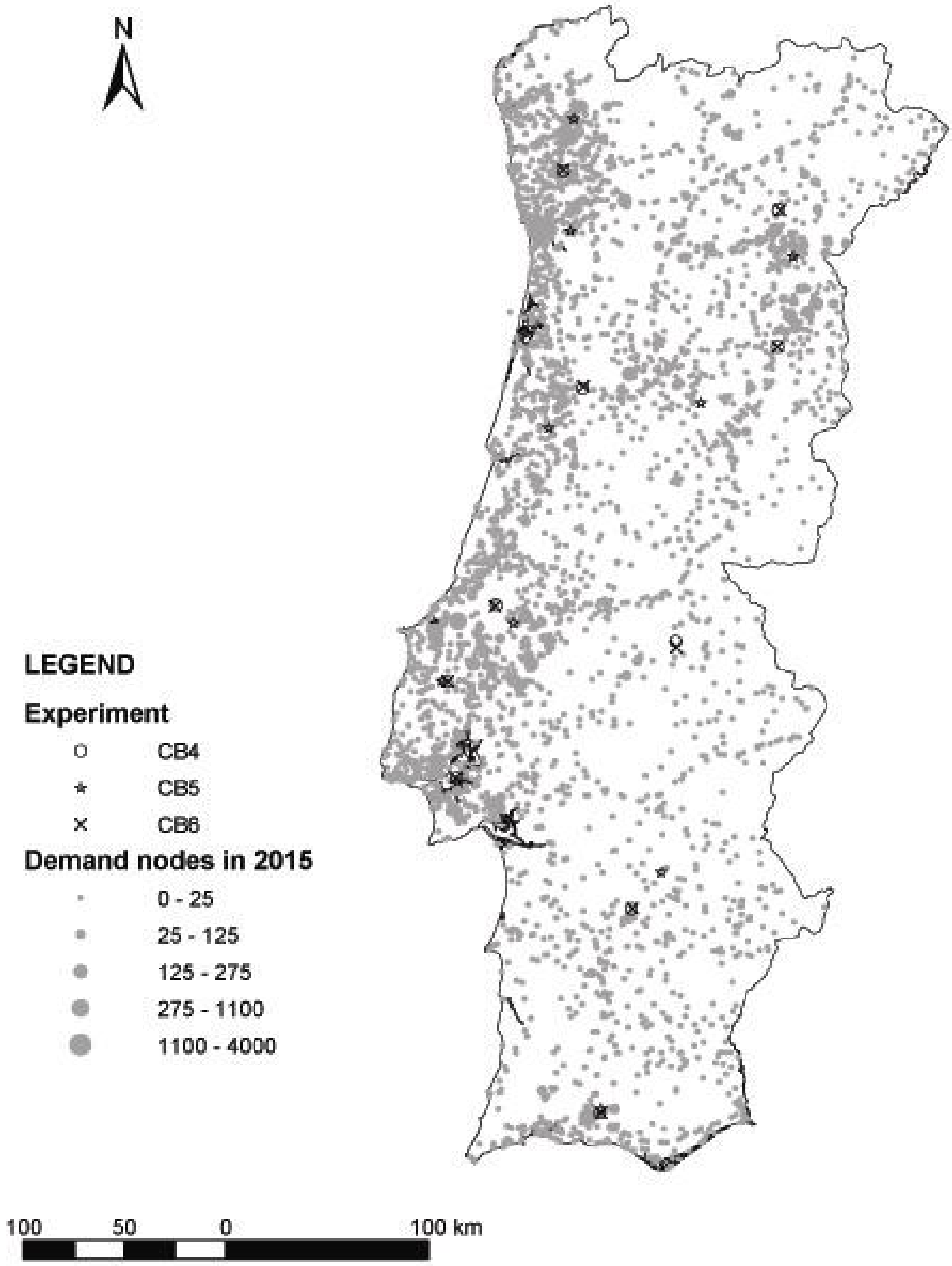

Figure III. 5 – Results from Test 1, for 2015 and considering 10 remediation units.

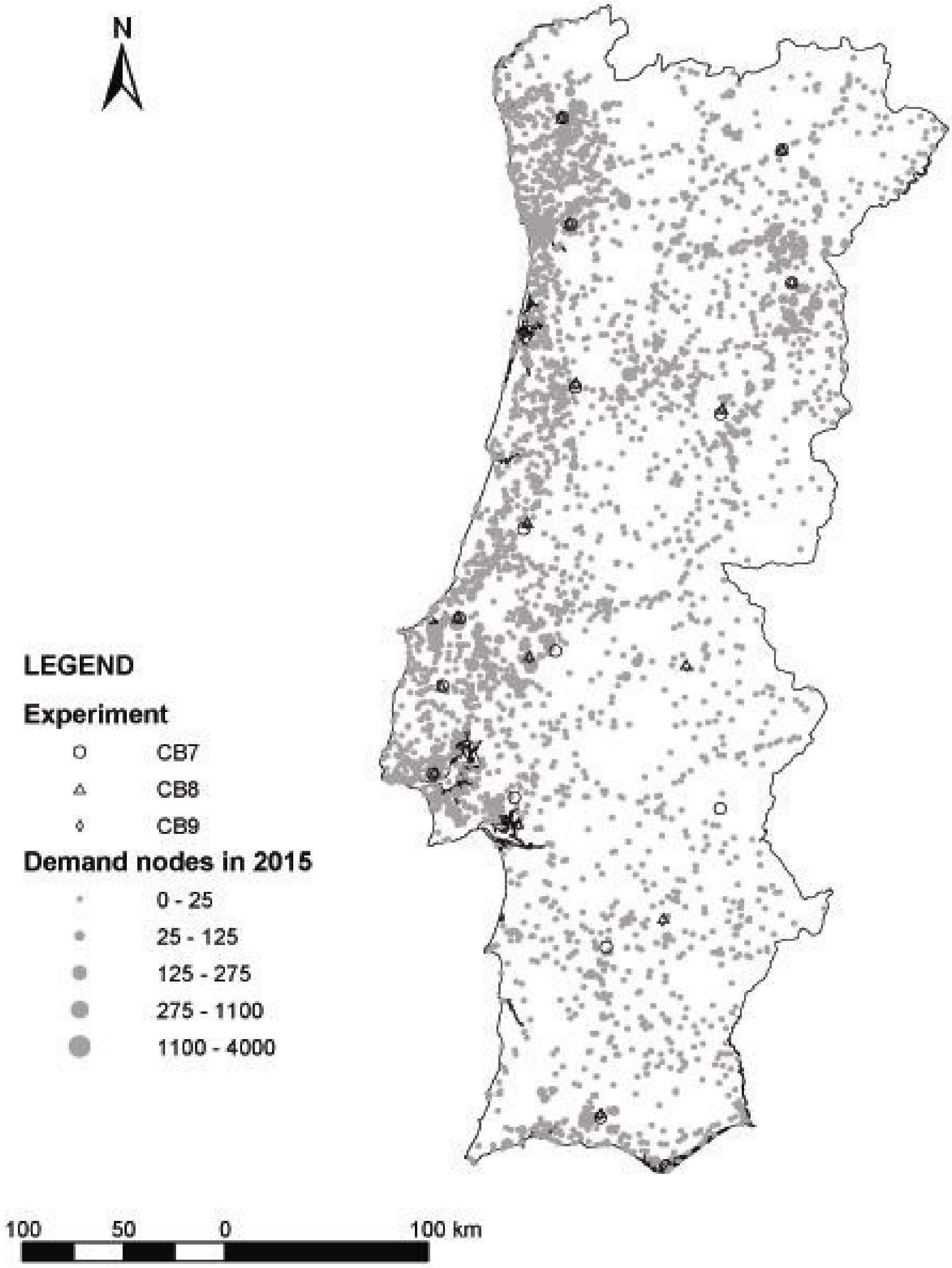

Figure III. 6 – Results from Test 1, for 2015 and considering 15 remediation units.

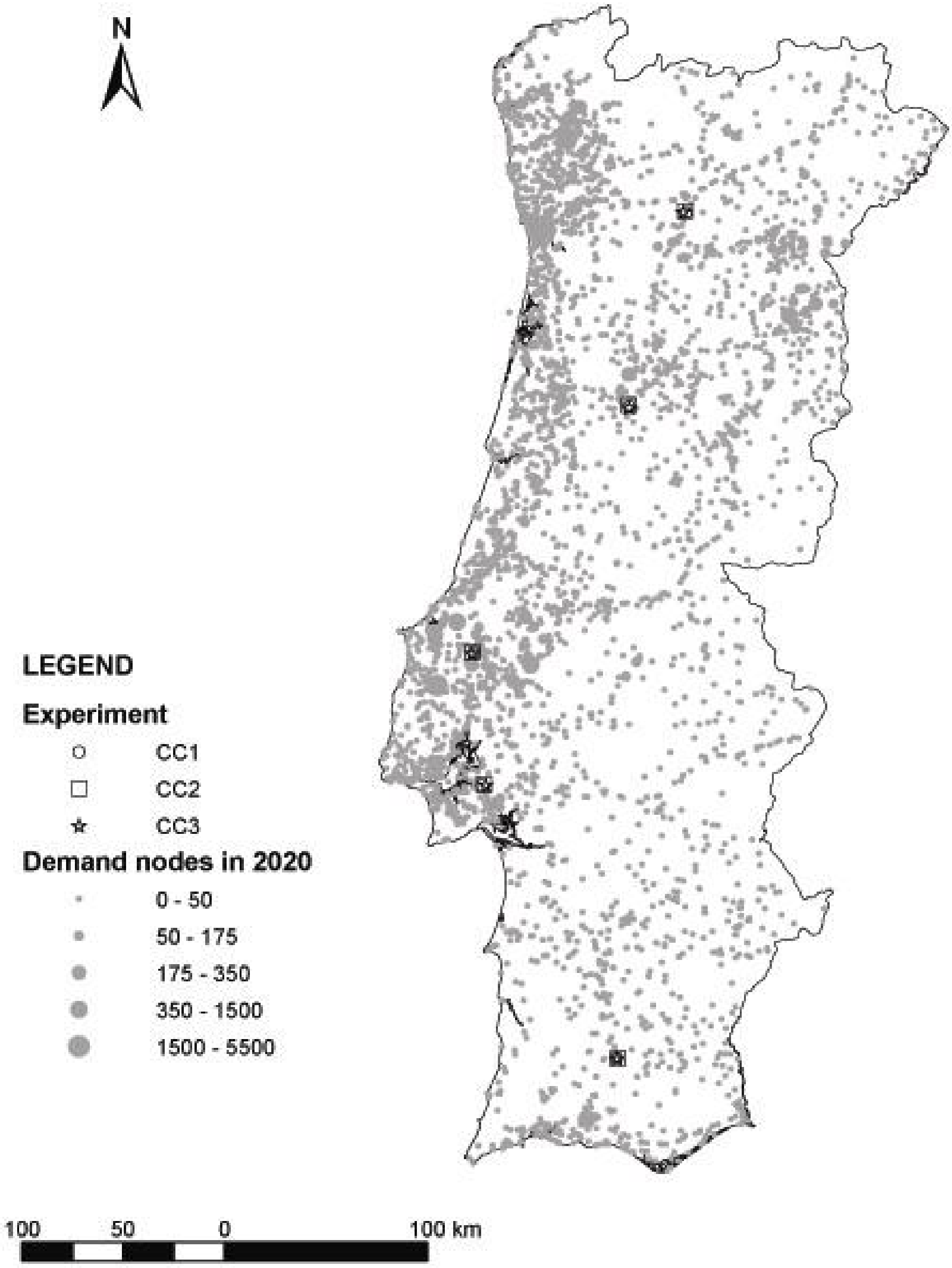

Figure III. 7 – Results from Test 1, for 2020 and considering 5 remediation units.

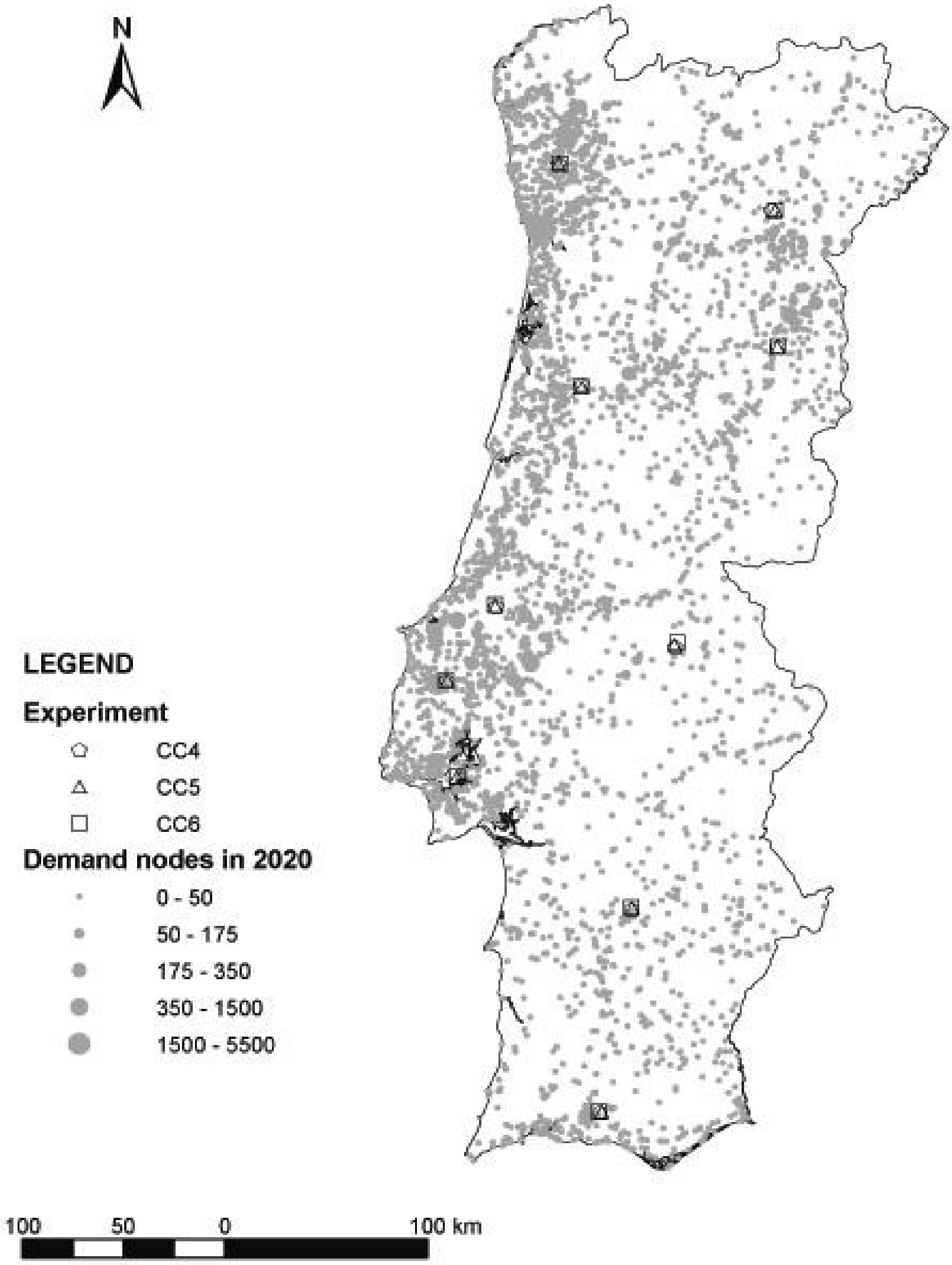

Figure III. 8 – Results from Test 1, for 2020 and considering 10 remediation units.

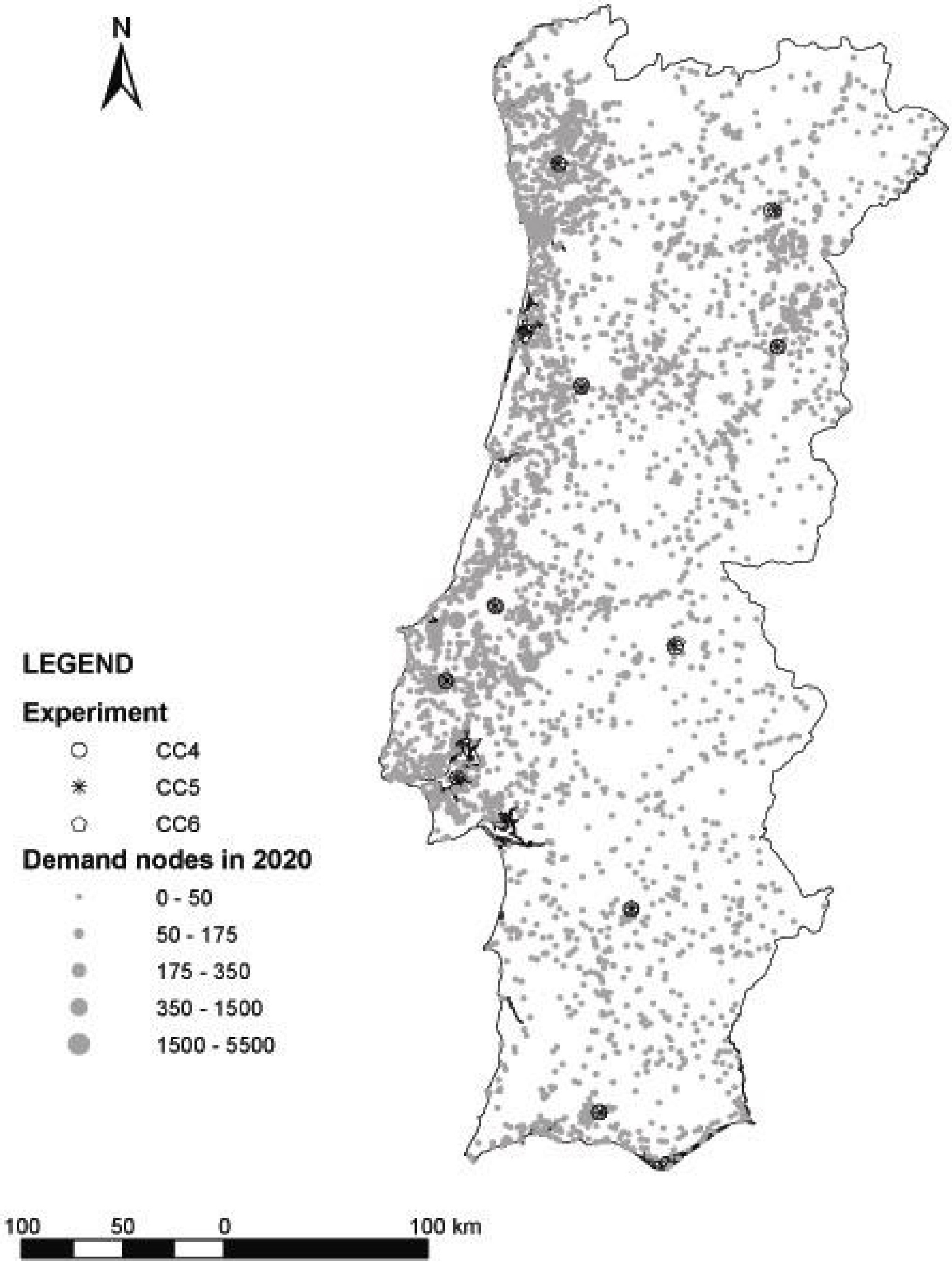

Figure III. 9 – Results from Test 1, for 2020 and considering 15 remediation units.

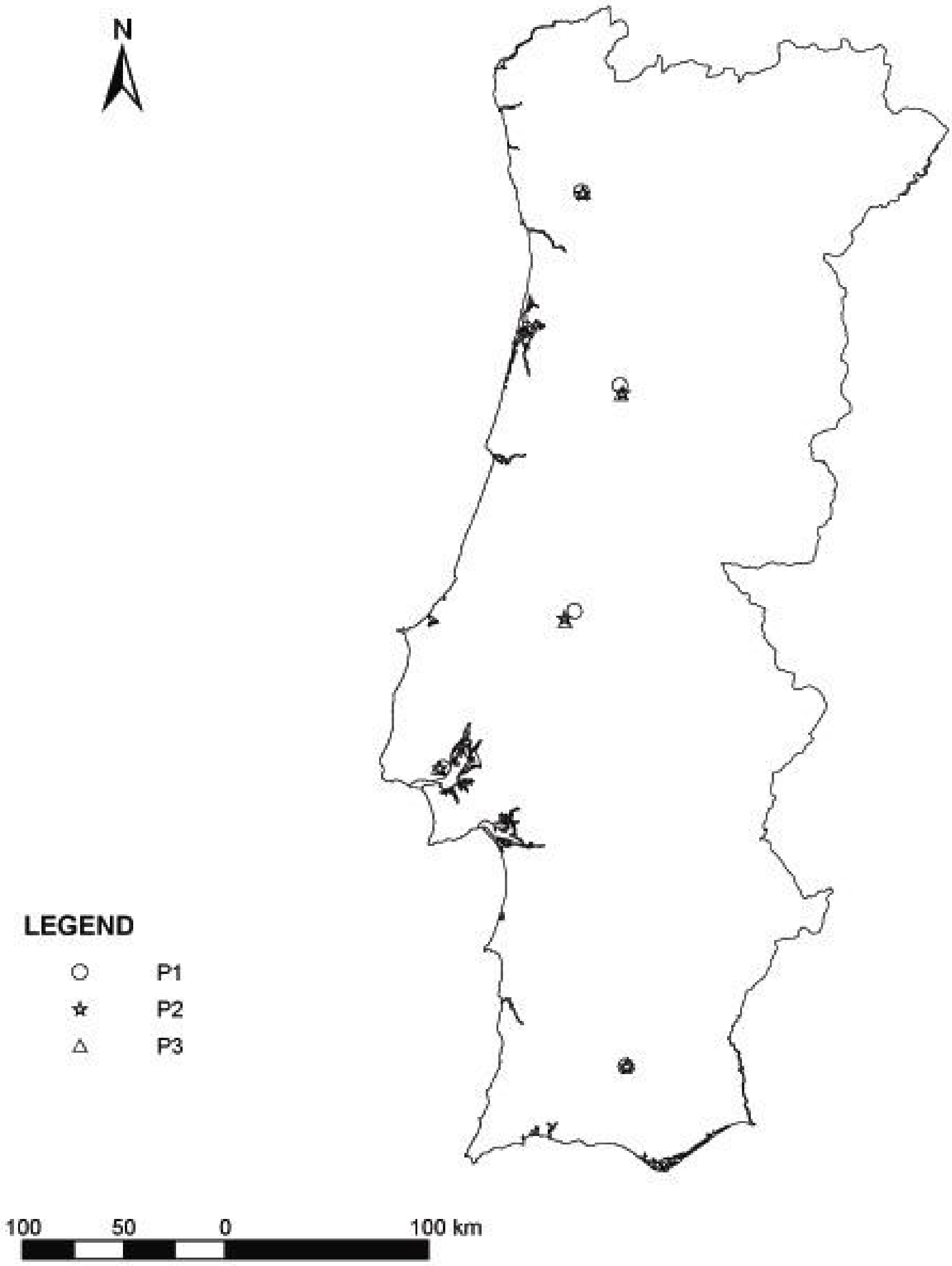

Figure III. 10 – Results from Test 1, for 2010, considering 5 remediation units and population data.

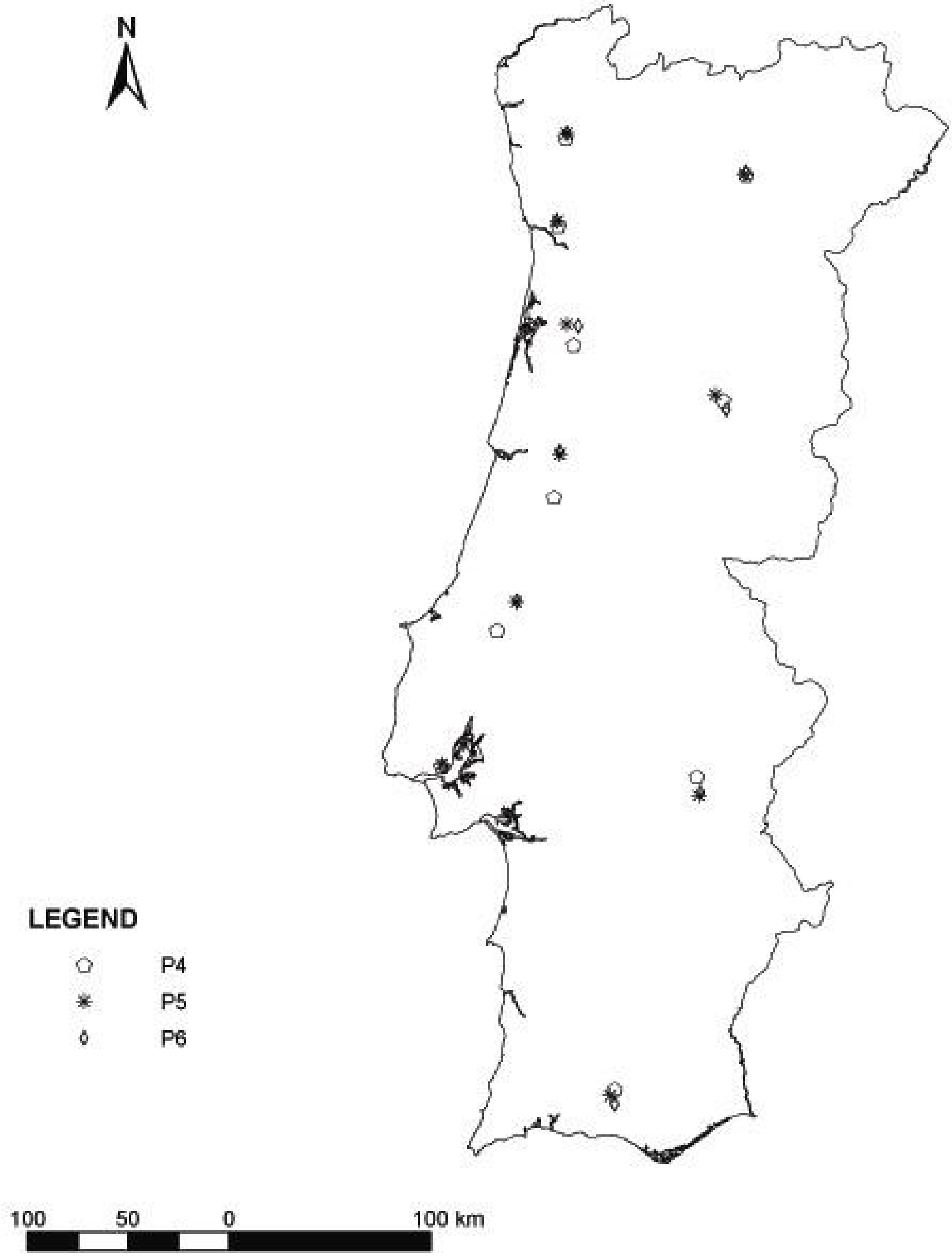

Figure III. 11 – Results from Test 1, for 2010, considering 10 remediation units and population data.

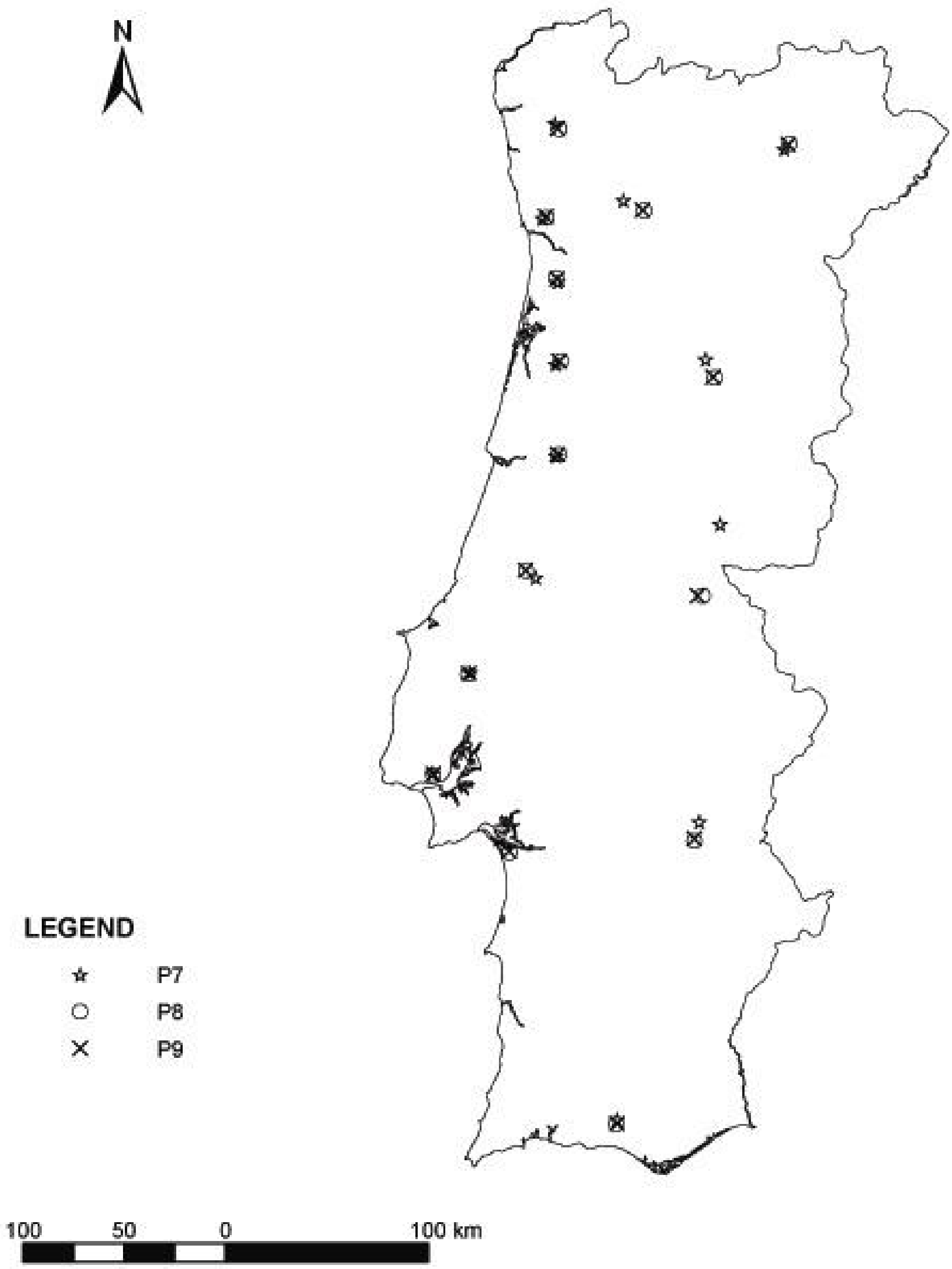

Figure III. 12 – Results from Test 1, for 2010, considering 15 remediation units and population data.

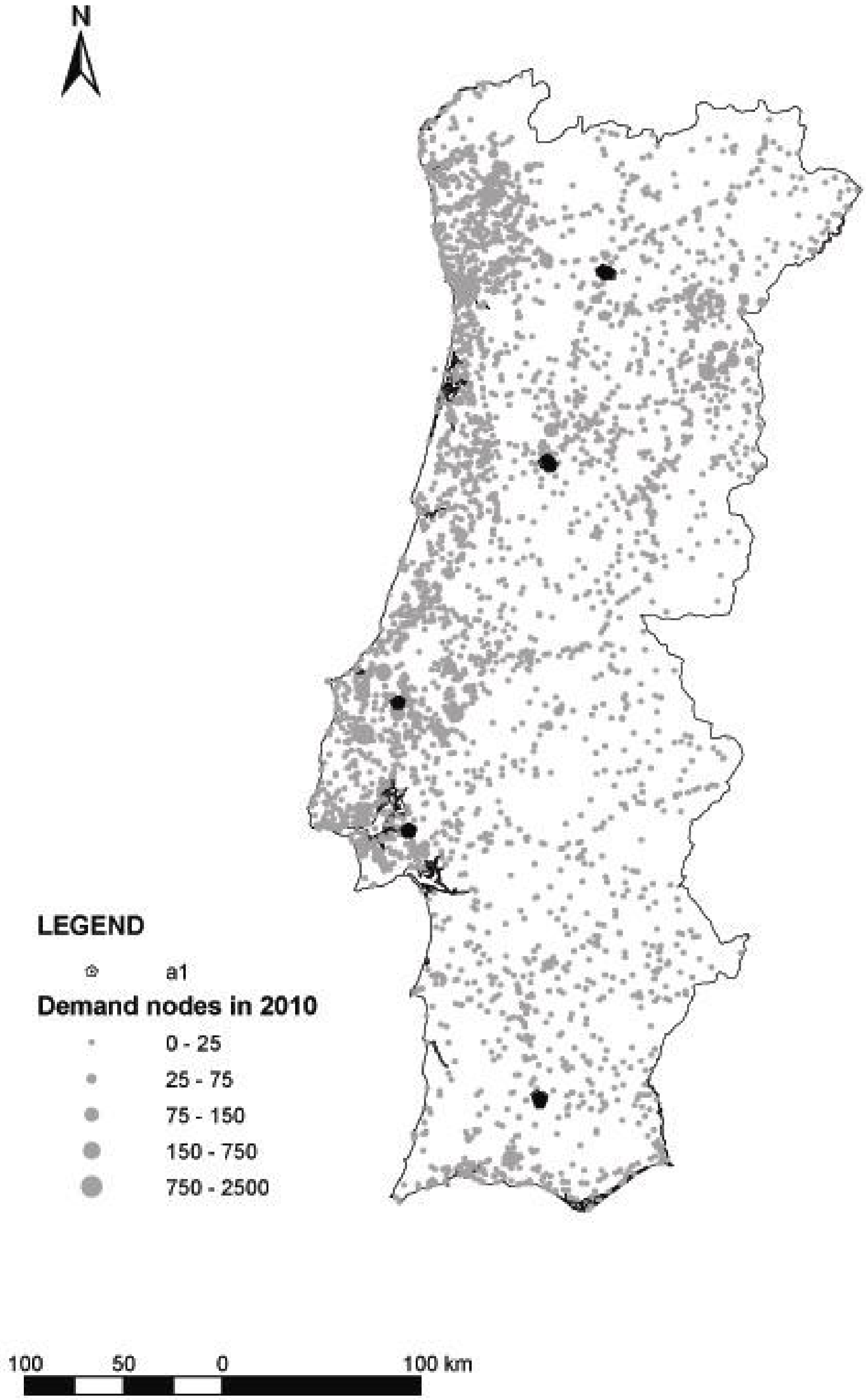

Figure III. 13 – Results from Test 2, for 2010 and considering 5 remediation units.

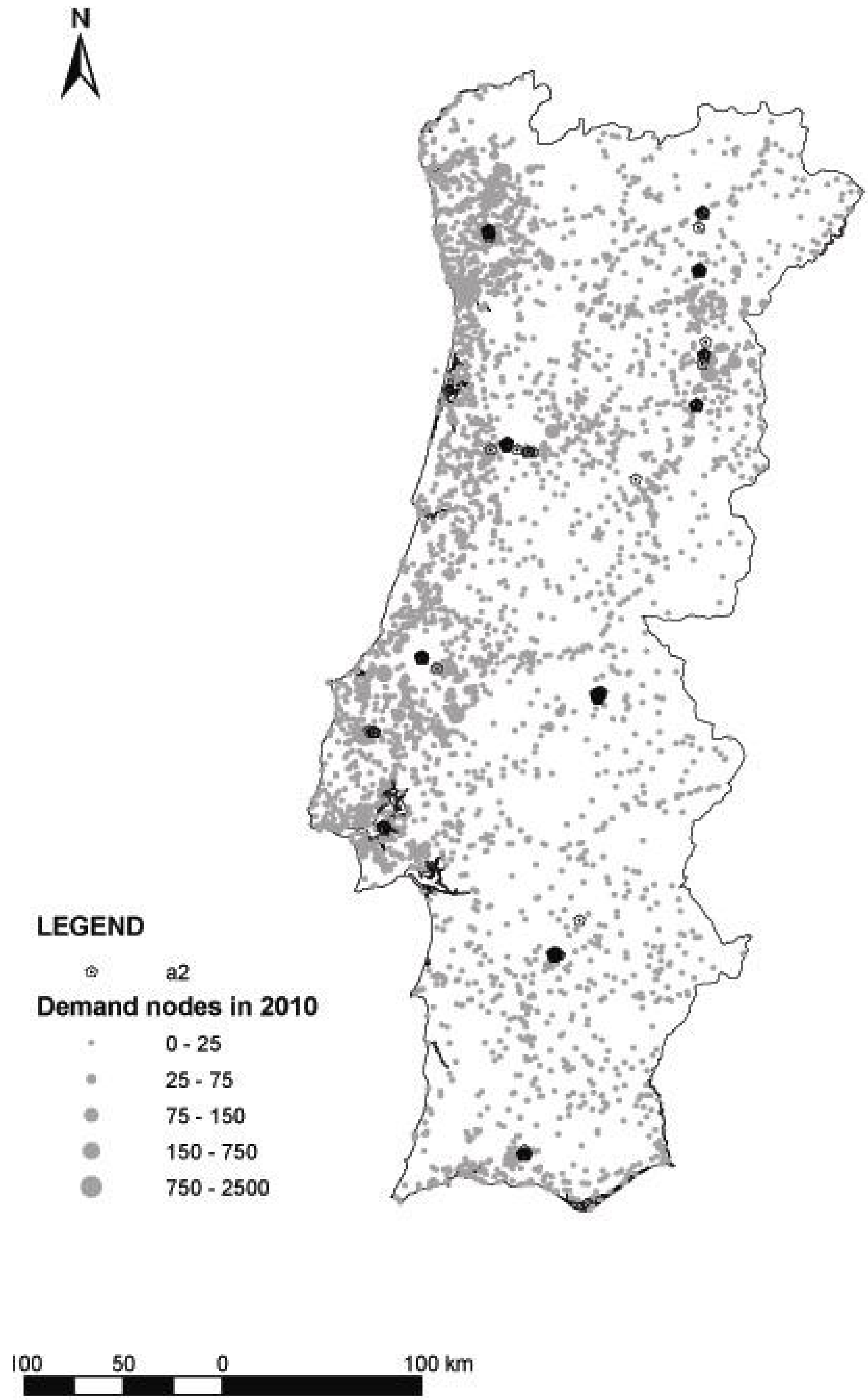

Figure III. 14 – Results from Test 2, for 2010 and considering 10 remediation units.

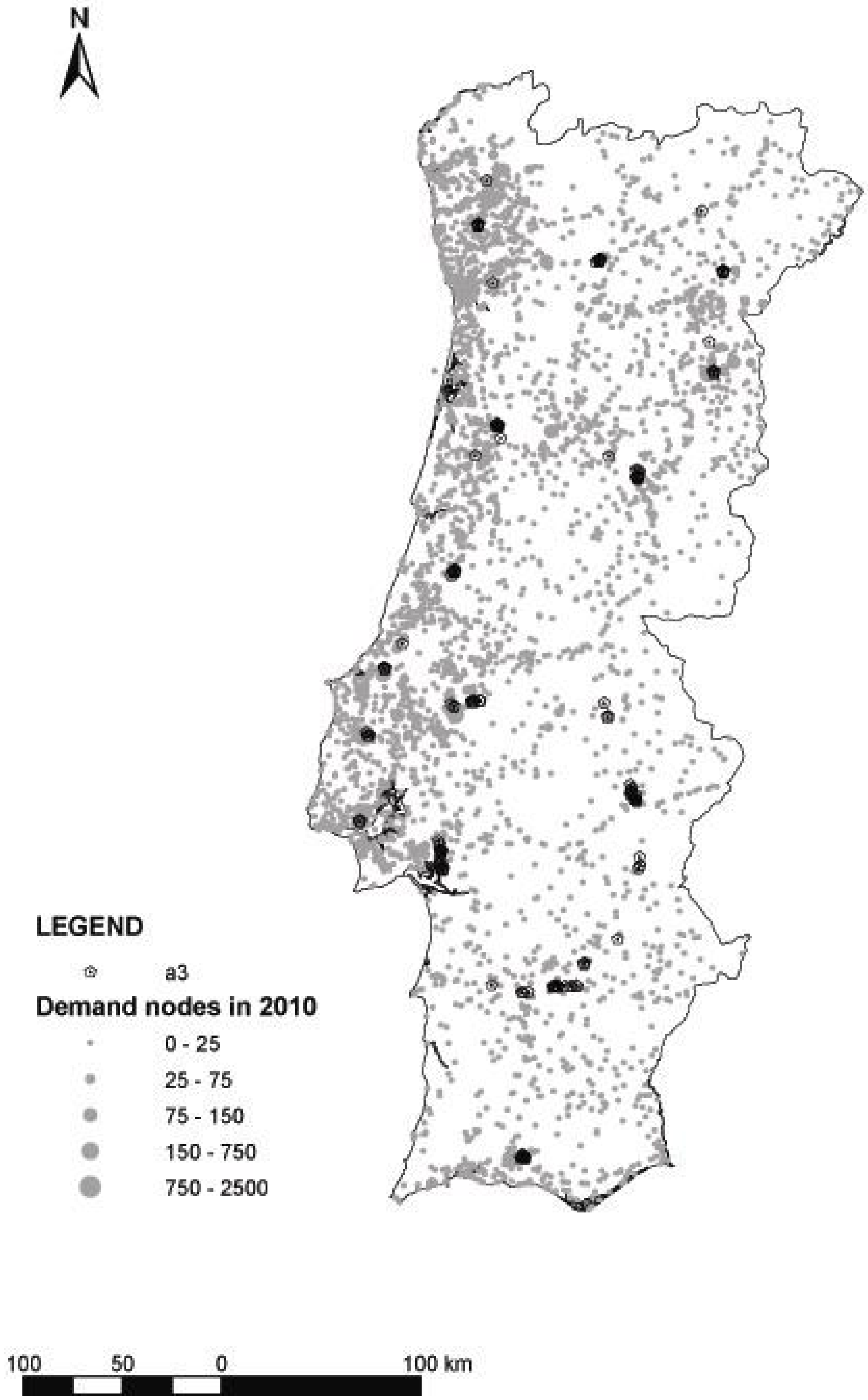

Figure III. 15 – Results from Test 2, for 2010 and considering 15 remediation units.

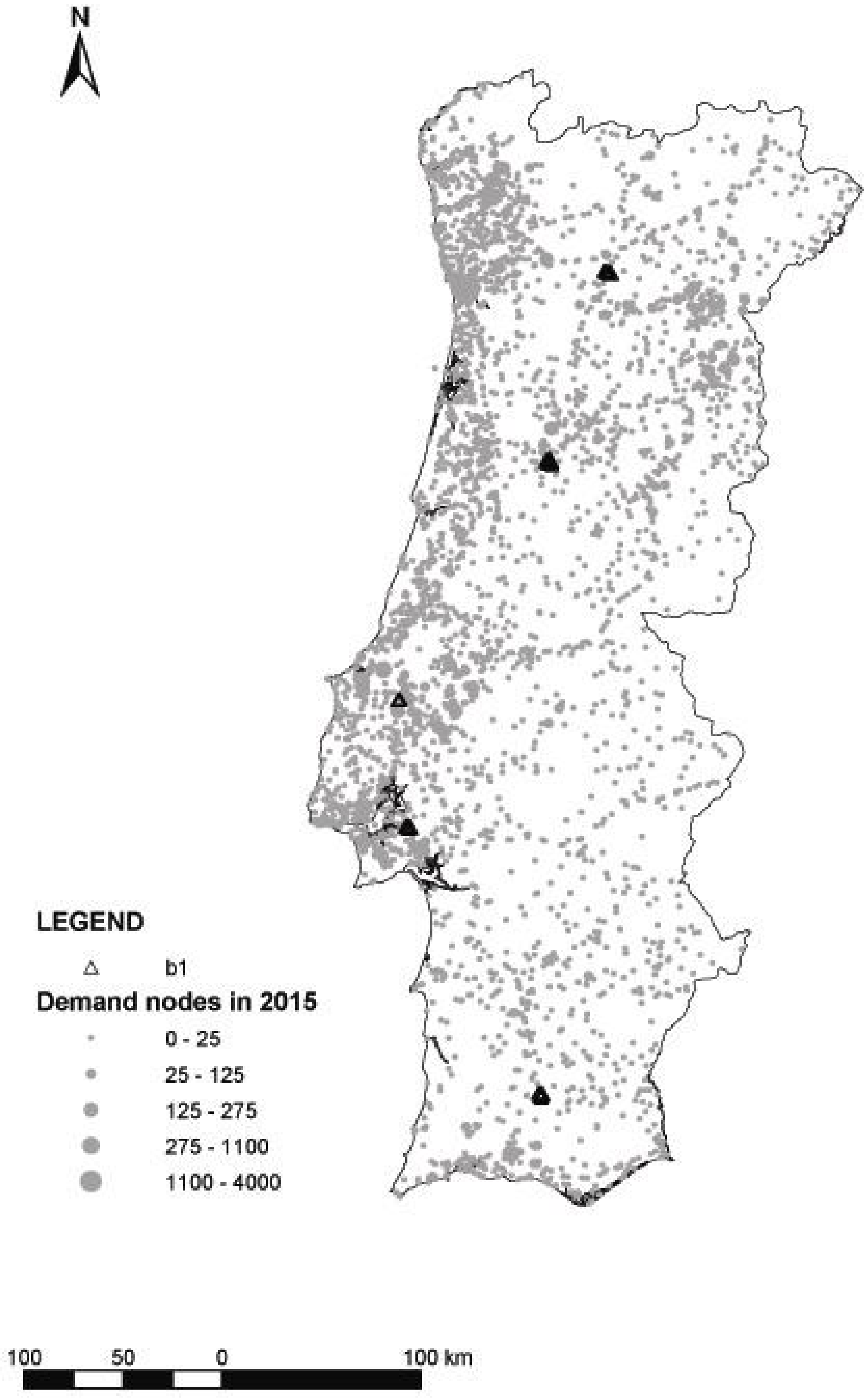

Figure III. 16 – Results from Test 2, for 2015 and considering 5 remediation units.

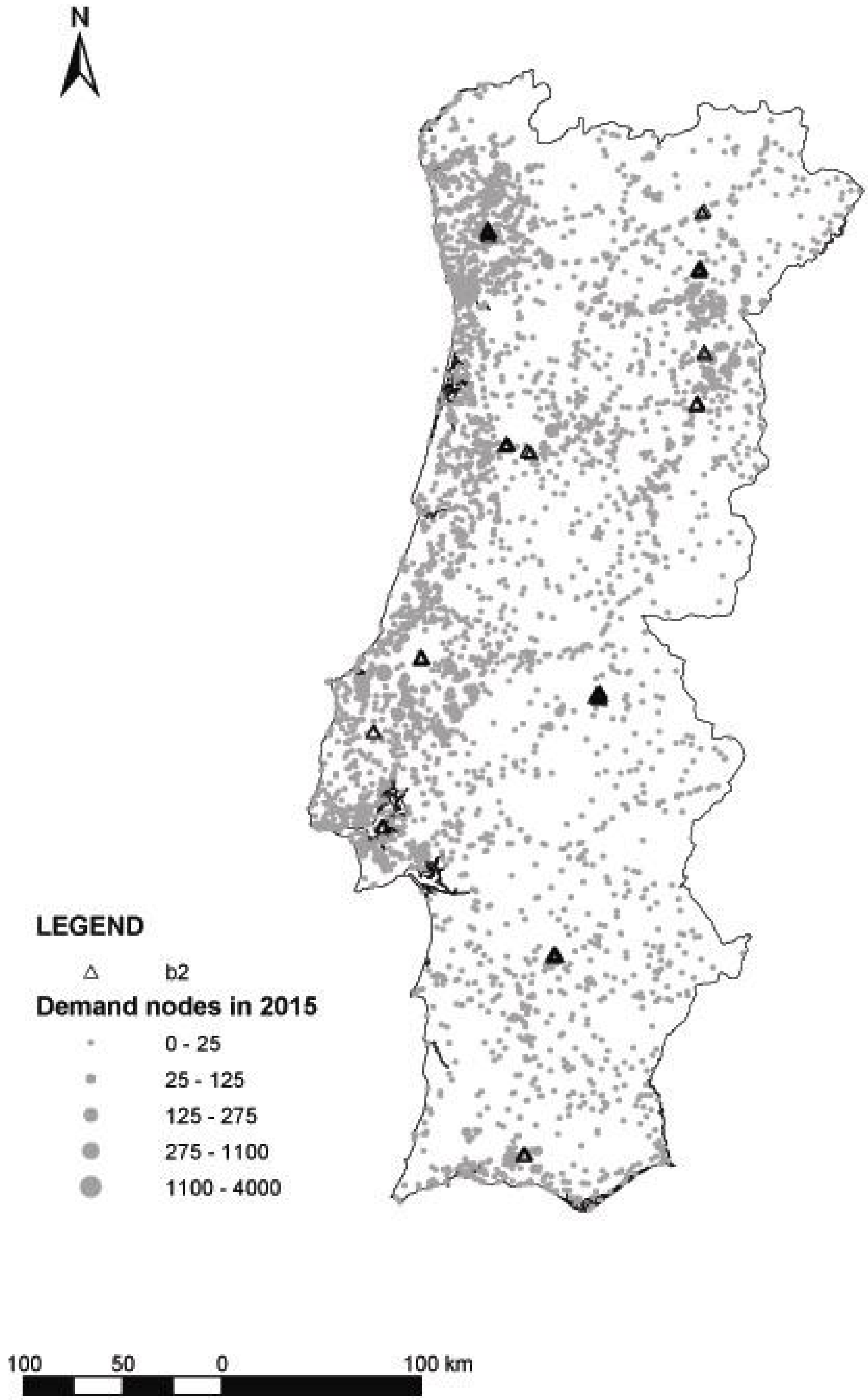

Figure III. 17 – Results from Test 2, for 2015 and considering 5 remediation units.

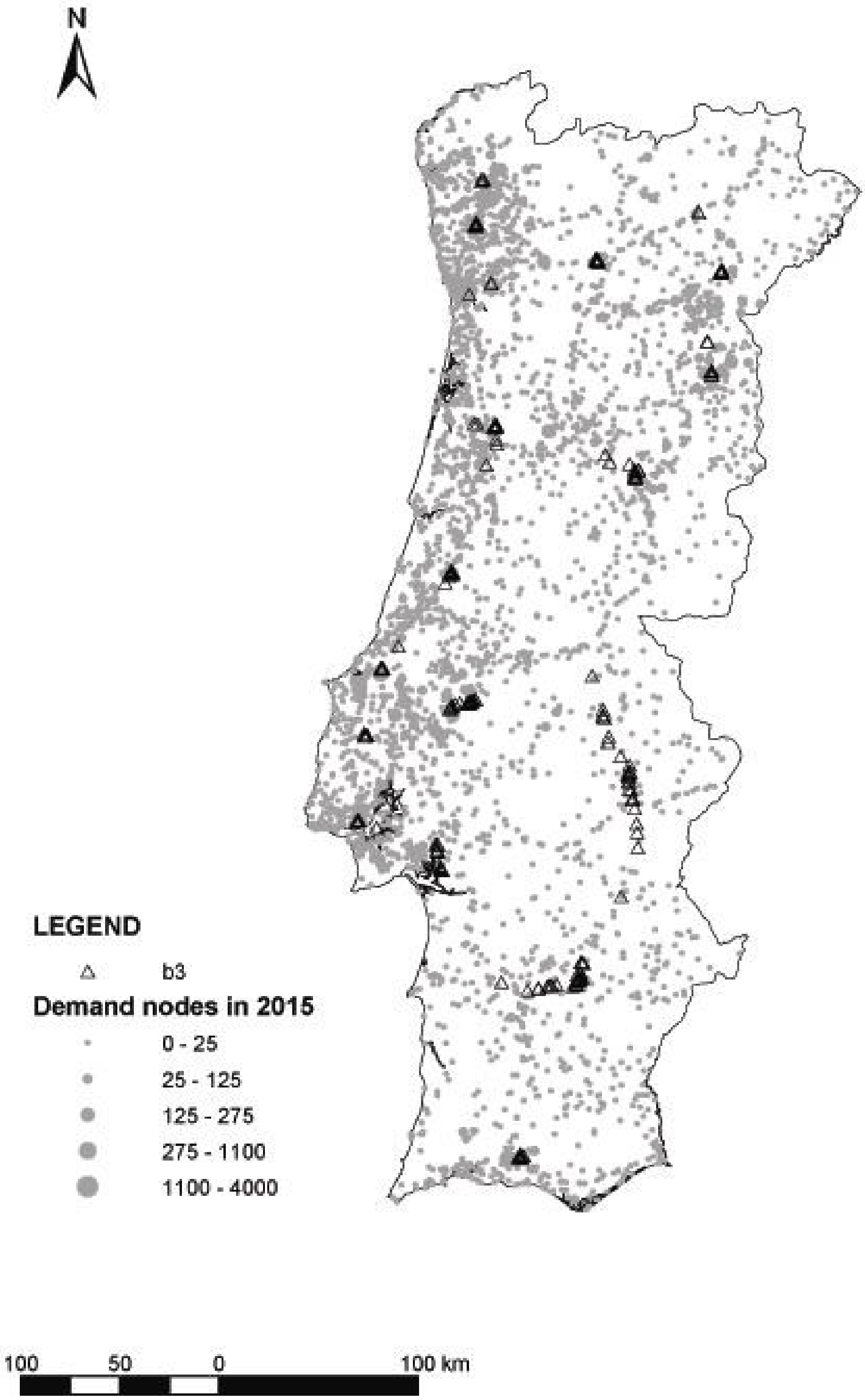

Figure III. 18 – Results from Test 2, for 2015 and considering 15 remediation units.

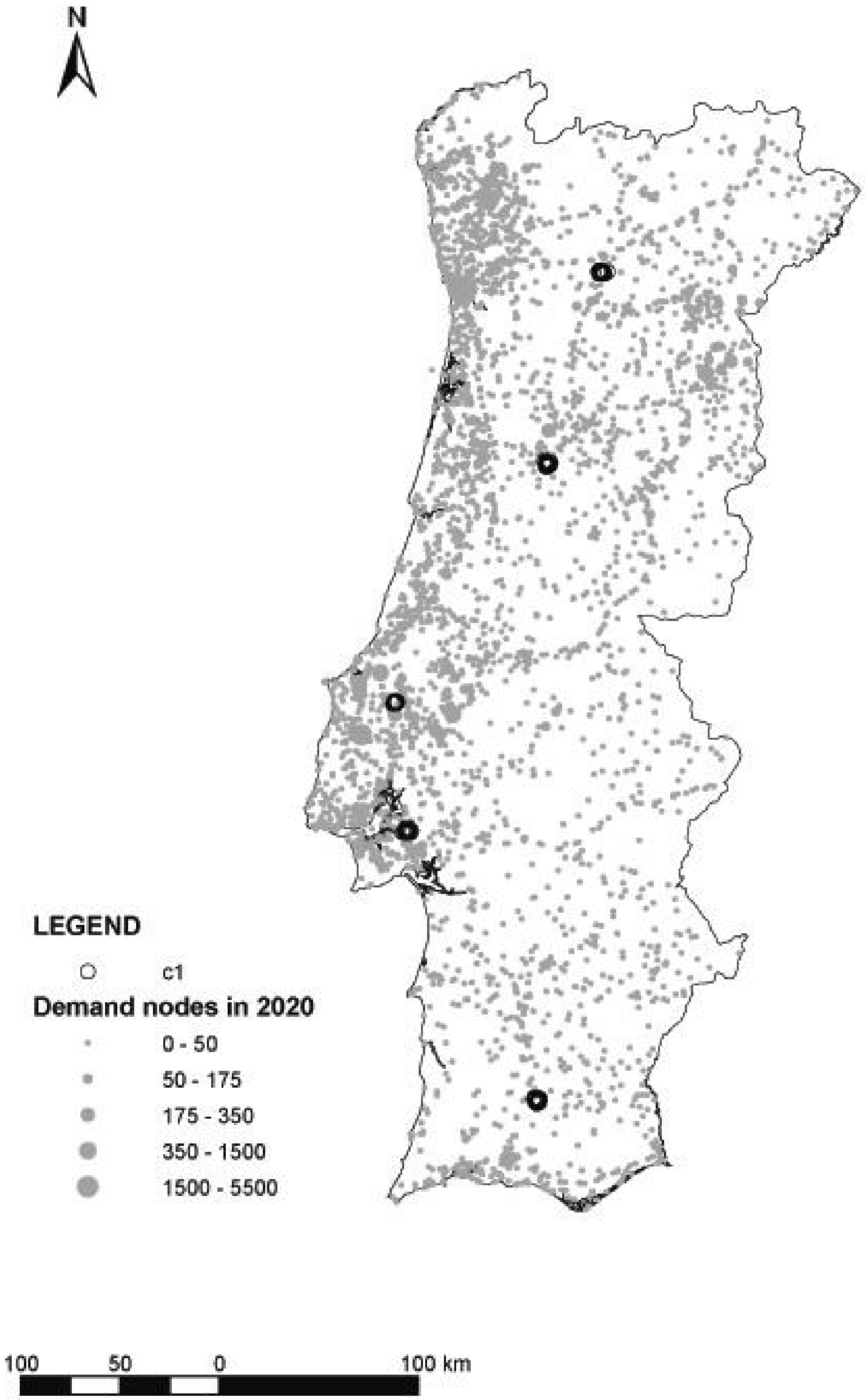

Figure III. 19 – Results from Test 2, for 2020 and considering 5 remediation units.

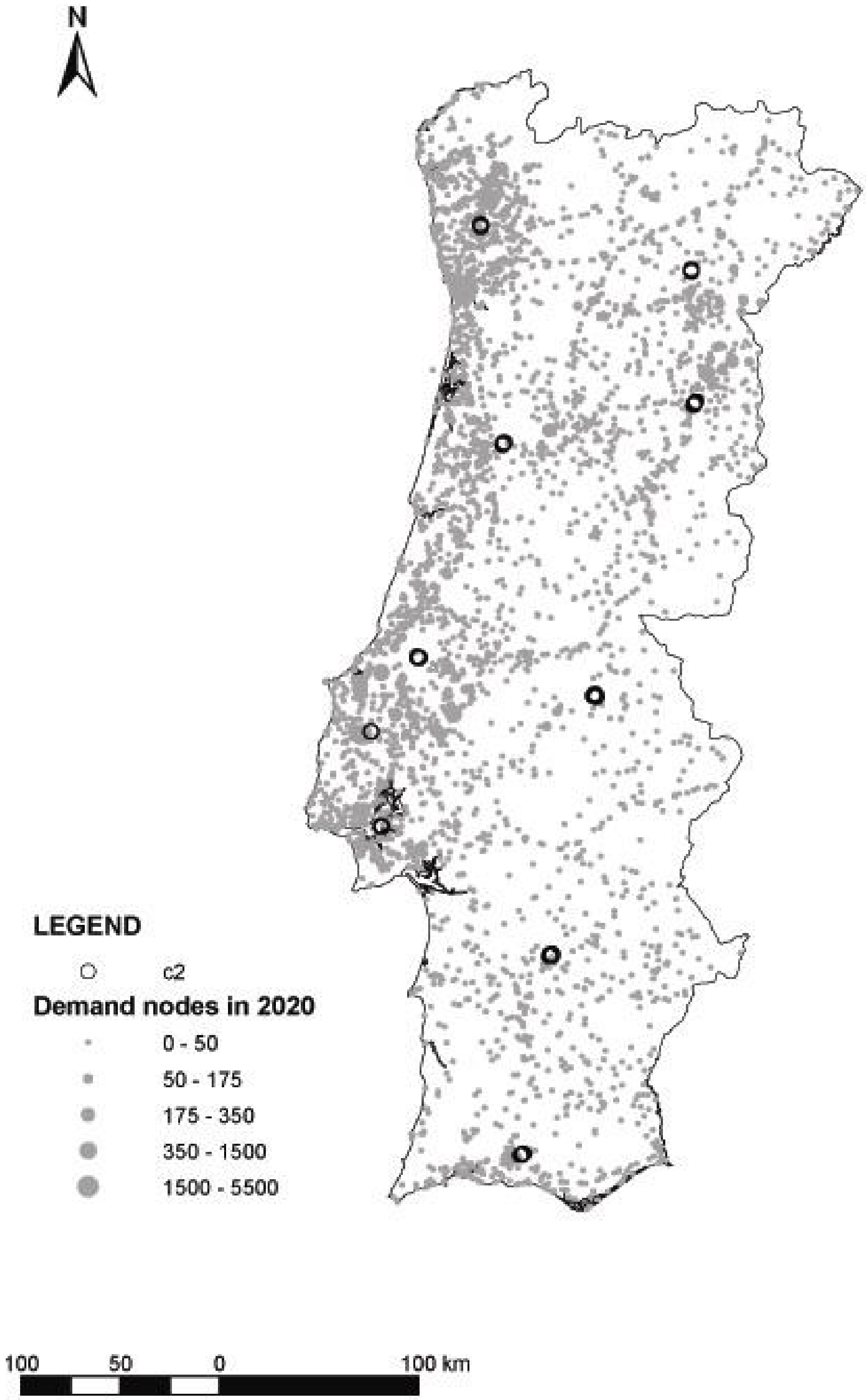

Figure III. 20 – Results from Test 2, for 2020 and considering 10 remediation units.

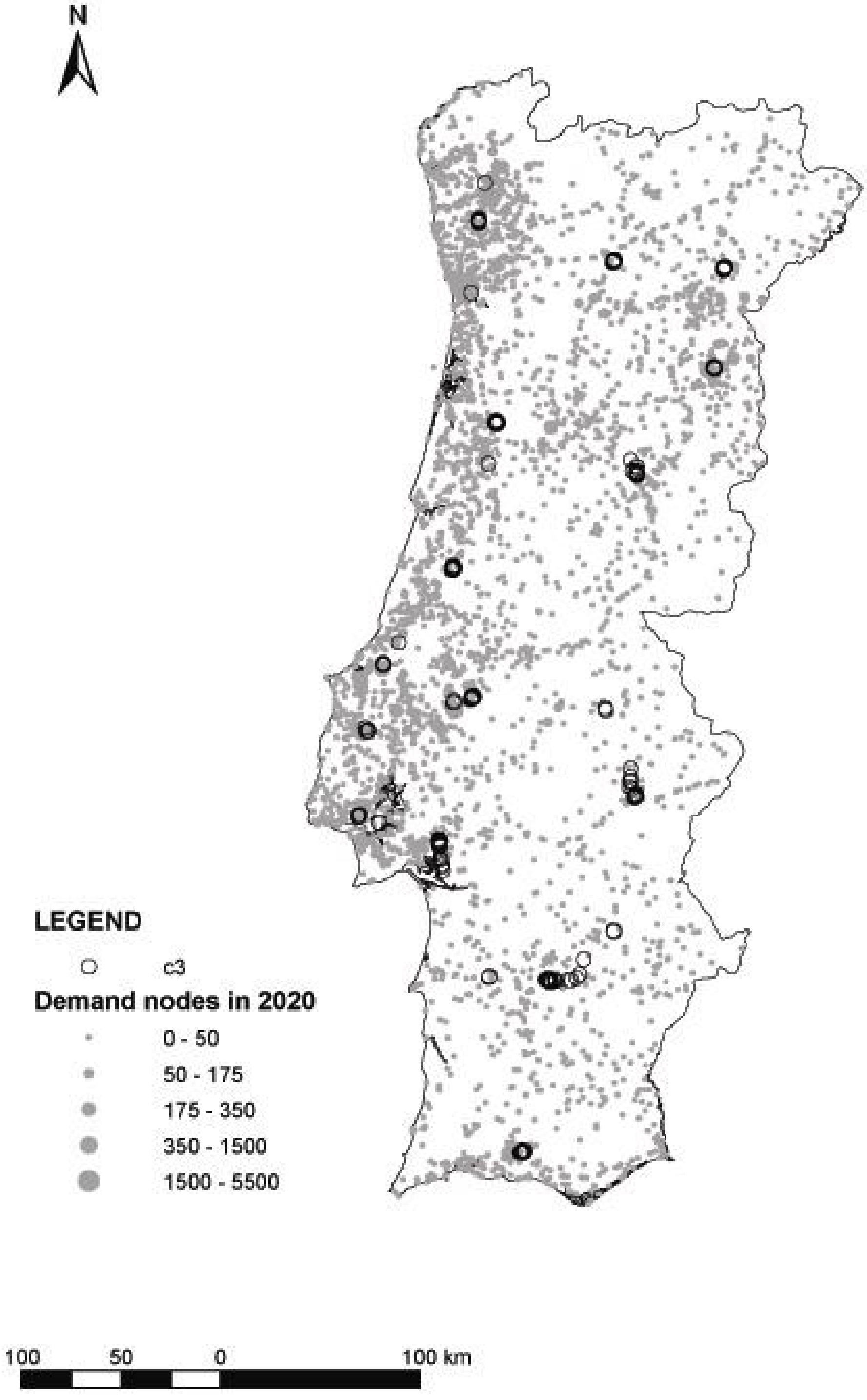

Figure III. 21 – Results from Test 2, for 2020 and considering 15 remediation units.

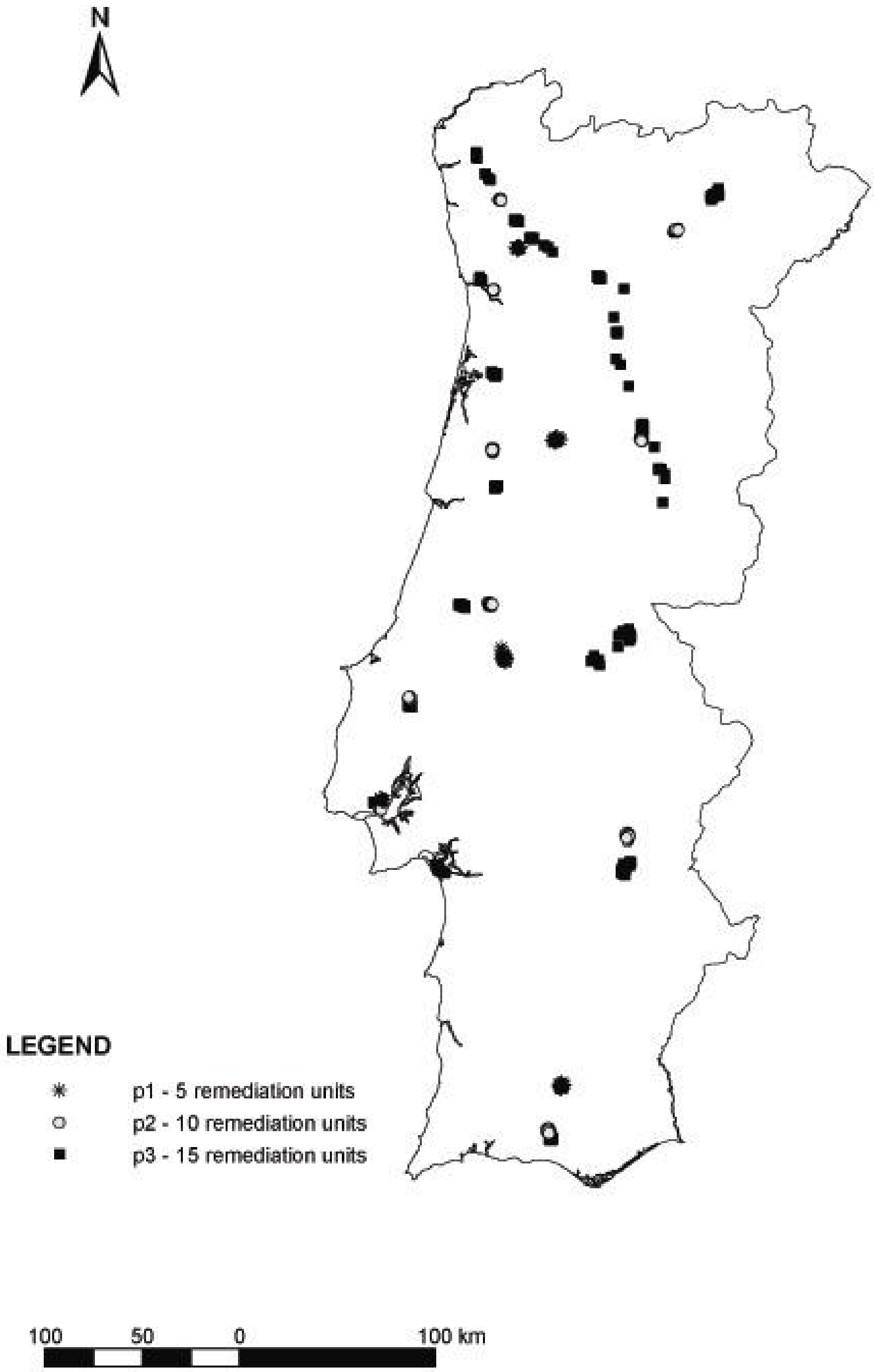

Figure III. 22 – Results from Test 2 using population data.

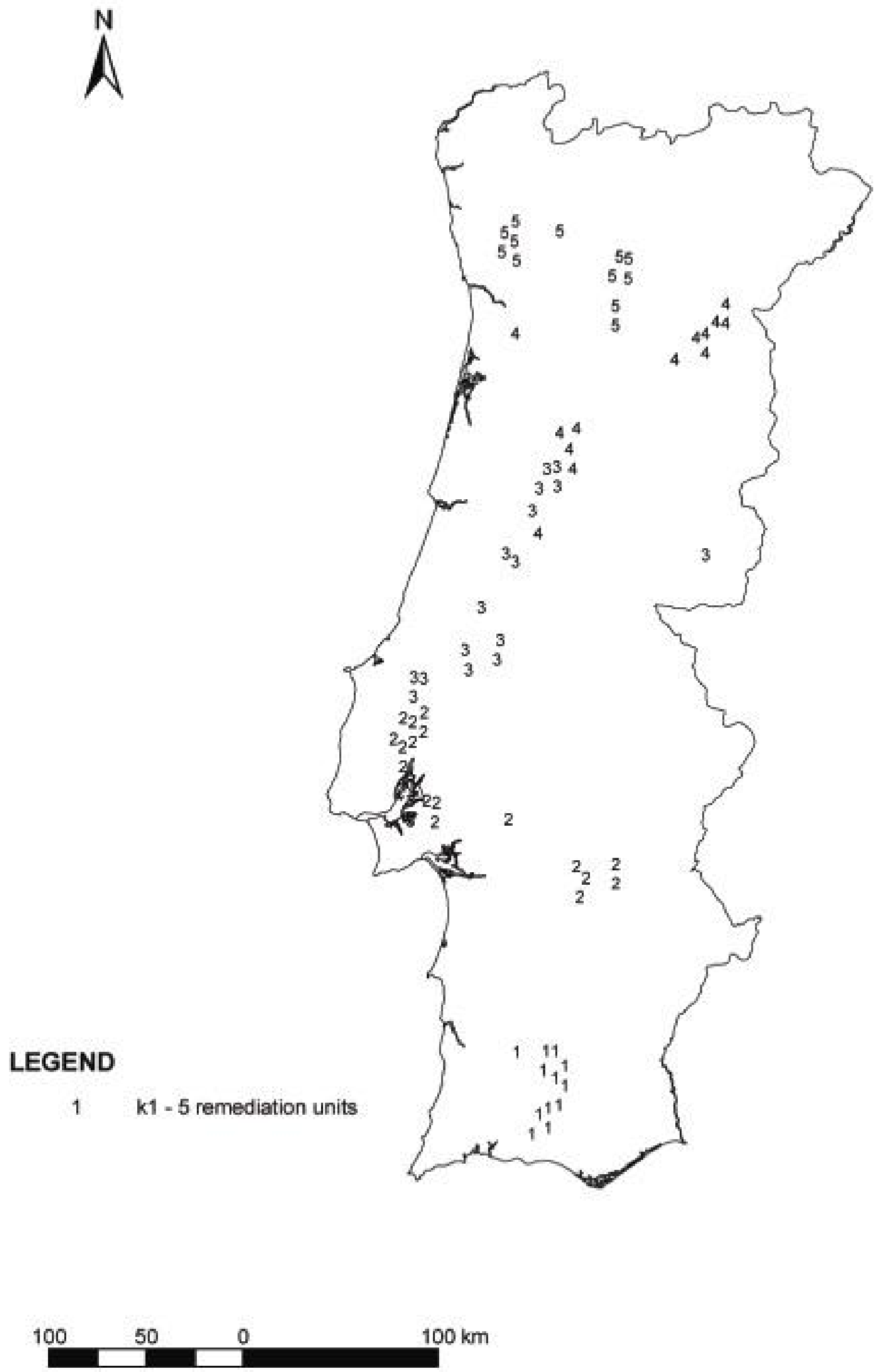

Figure III. 23 – Results from Test 2, for 2010, using k-means and considering 5 remediation units.

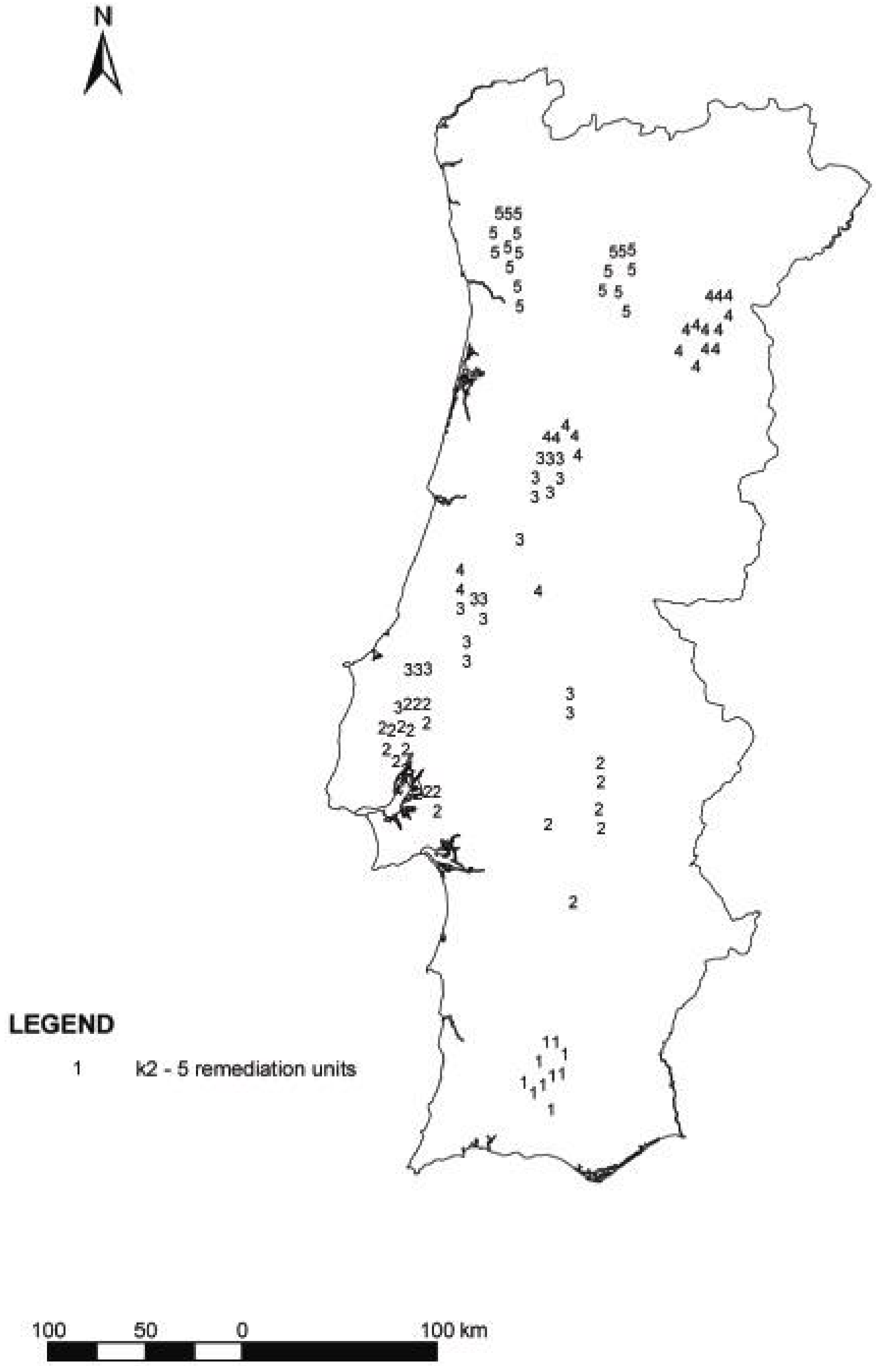

Figure III. 24 – Results from Test 2, for 2015, using k-means and considering 5 remediation units.

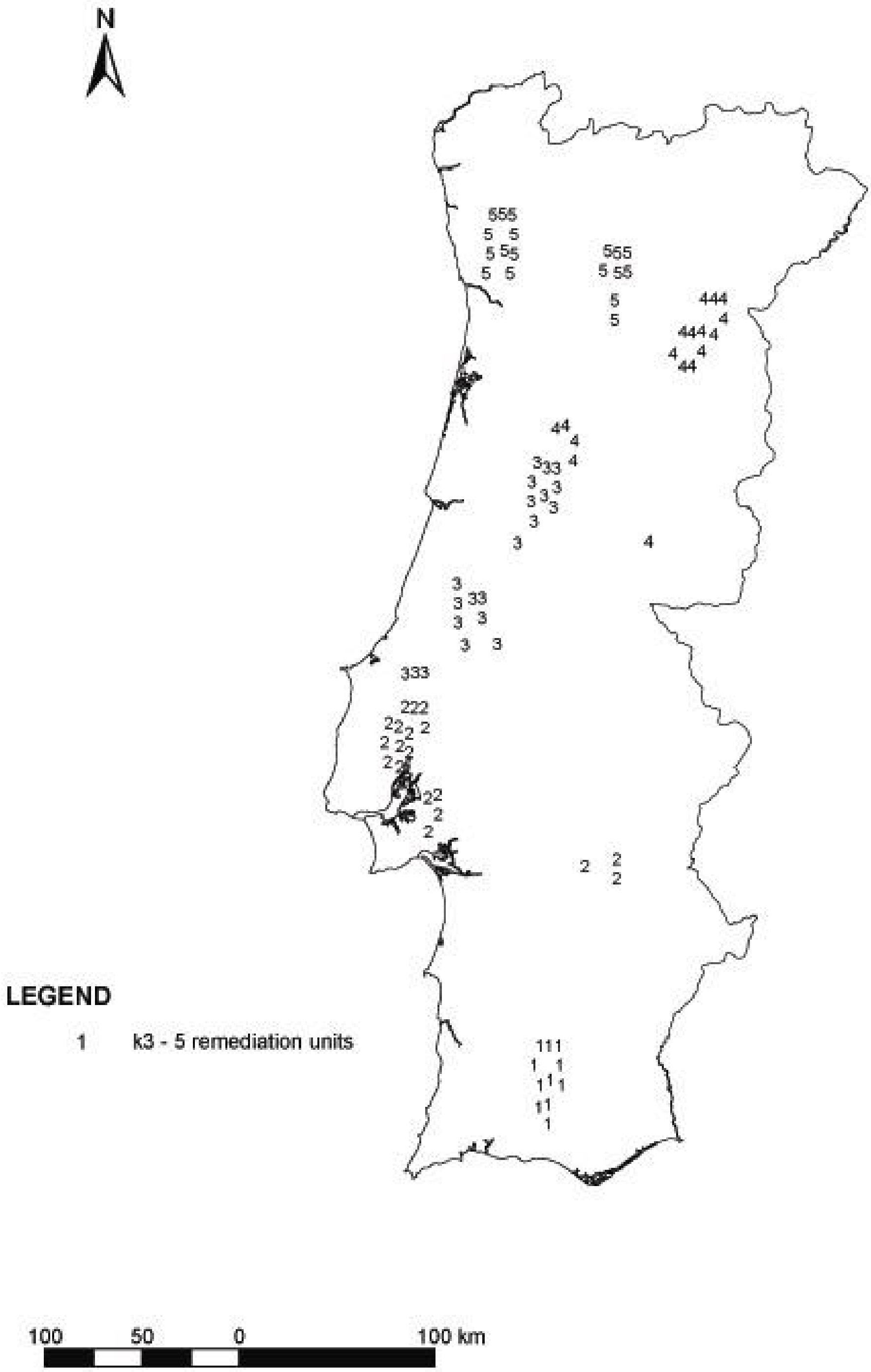

Figure III. 25 – Results from Test 2, for 2020, using k-means and considering 5 remediation units.

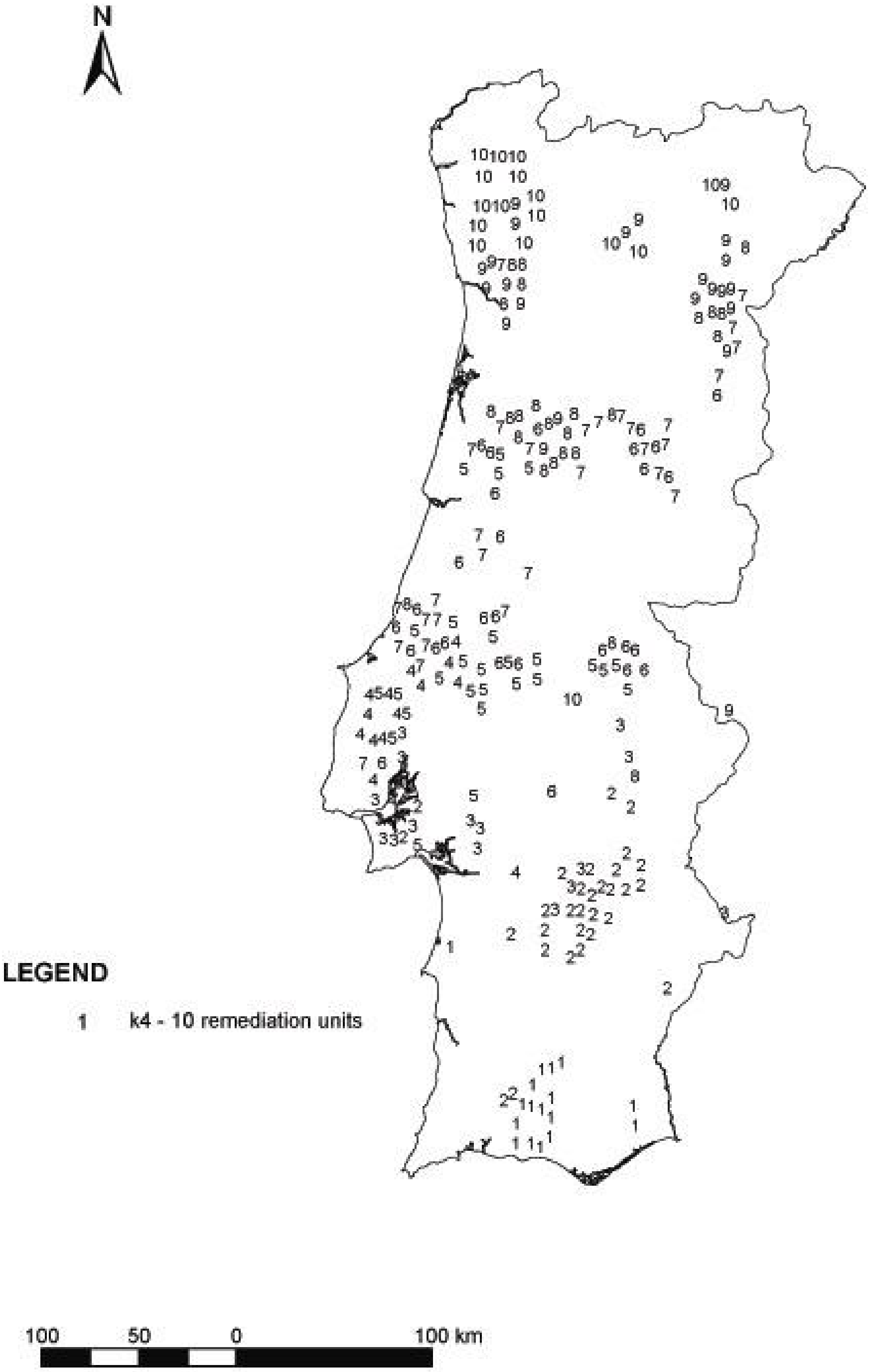

Figure III. 26 – Results from Test 2, for 2010, using k-means and considering 10 remediation units.

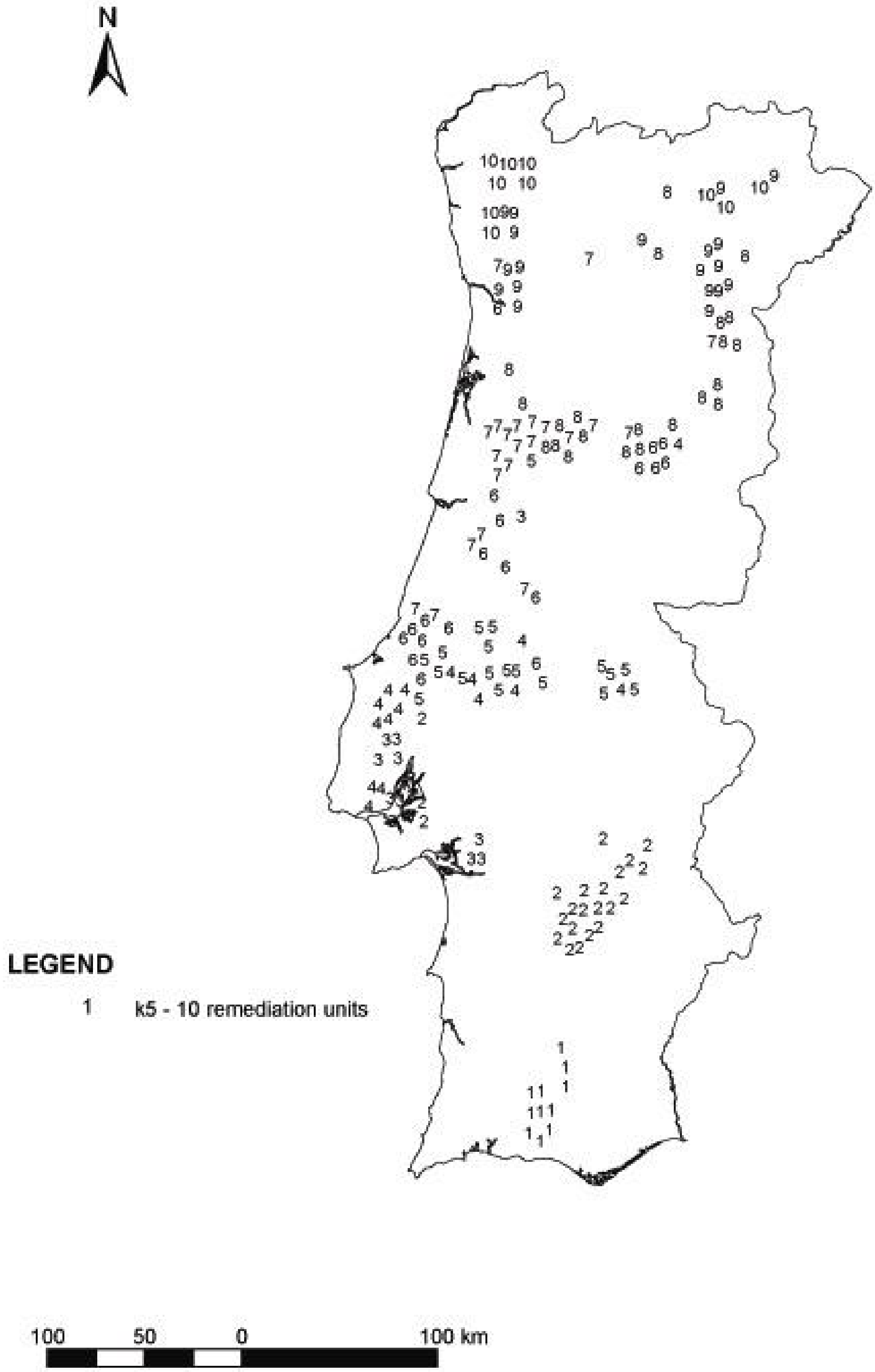

Figure III. 27 – Results from Test 2, for 2015, using k-means and considering 10 remediation units.

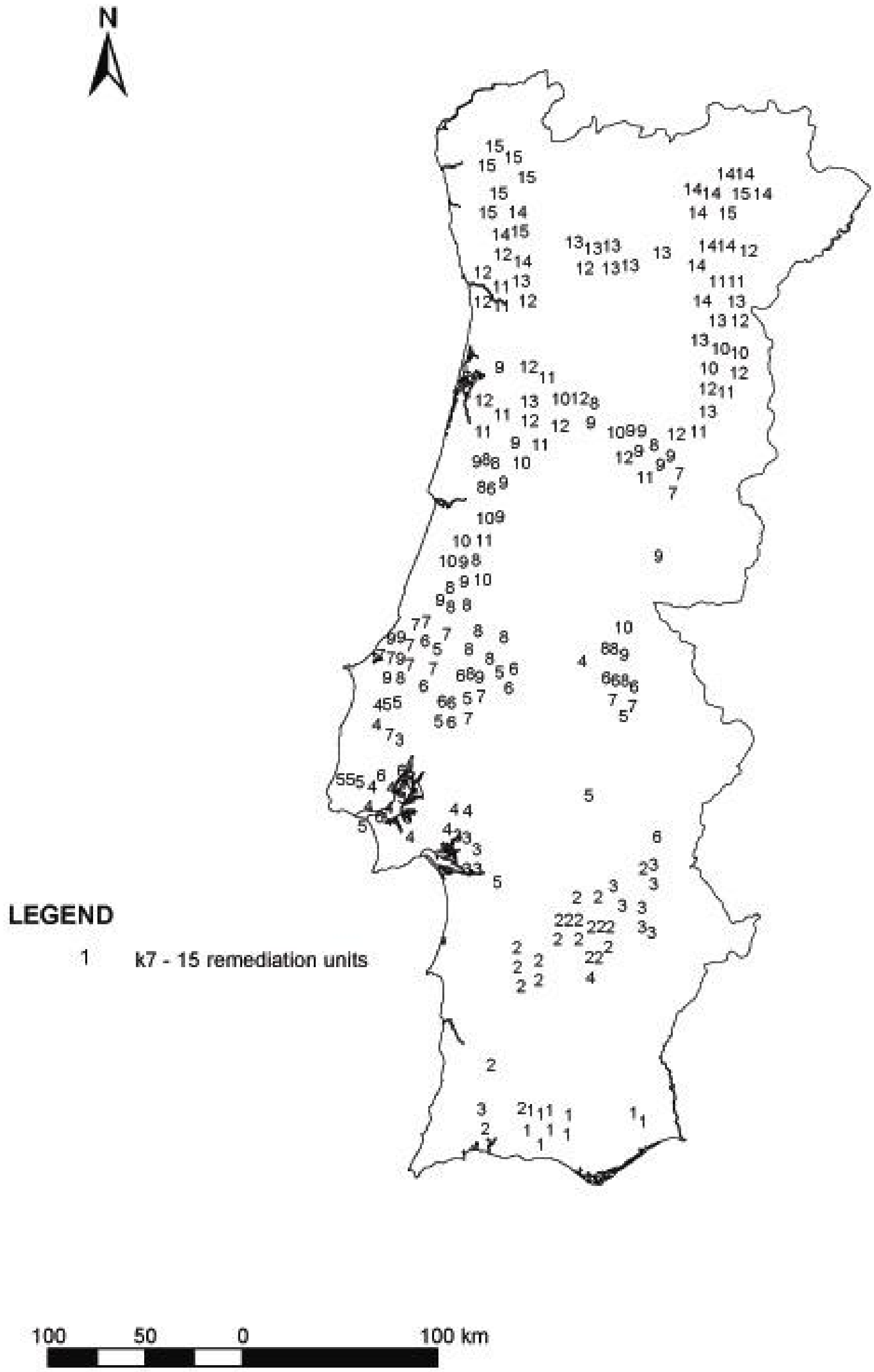

Figure III. 28 – Results from Test 2, for 2010, using k-means and considering 15 remediation units.

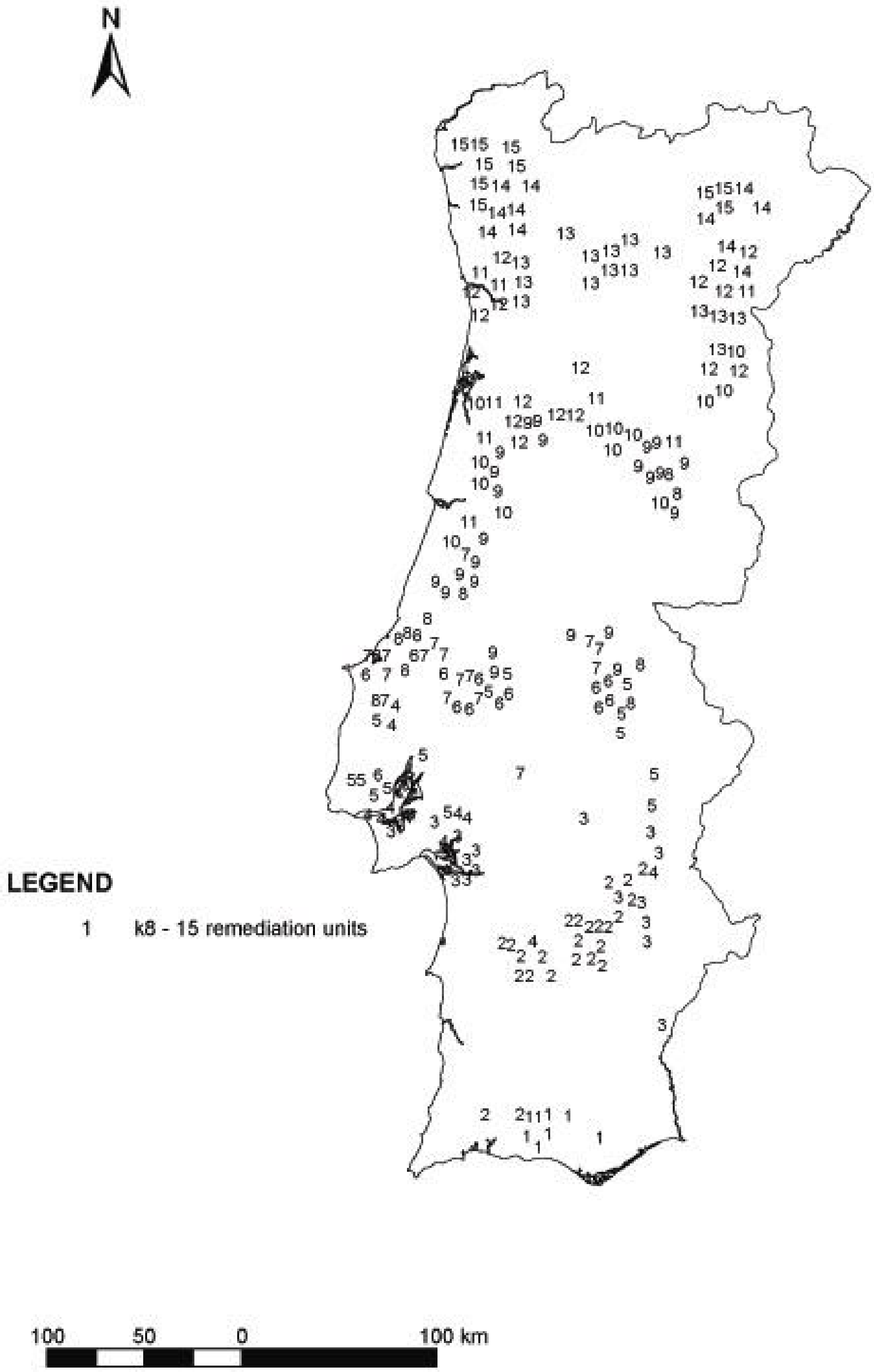

Figure III. 29 – Results from Test 2, for 2015, using k-means and considering 15 remediation units.

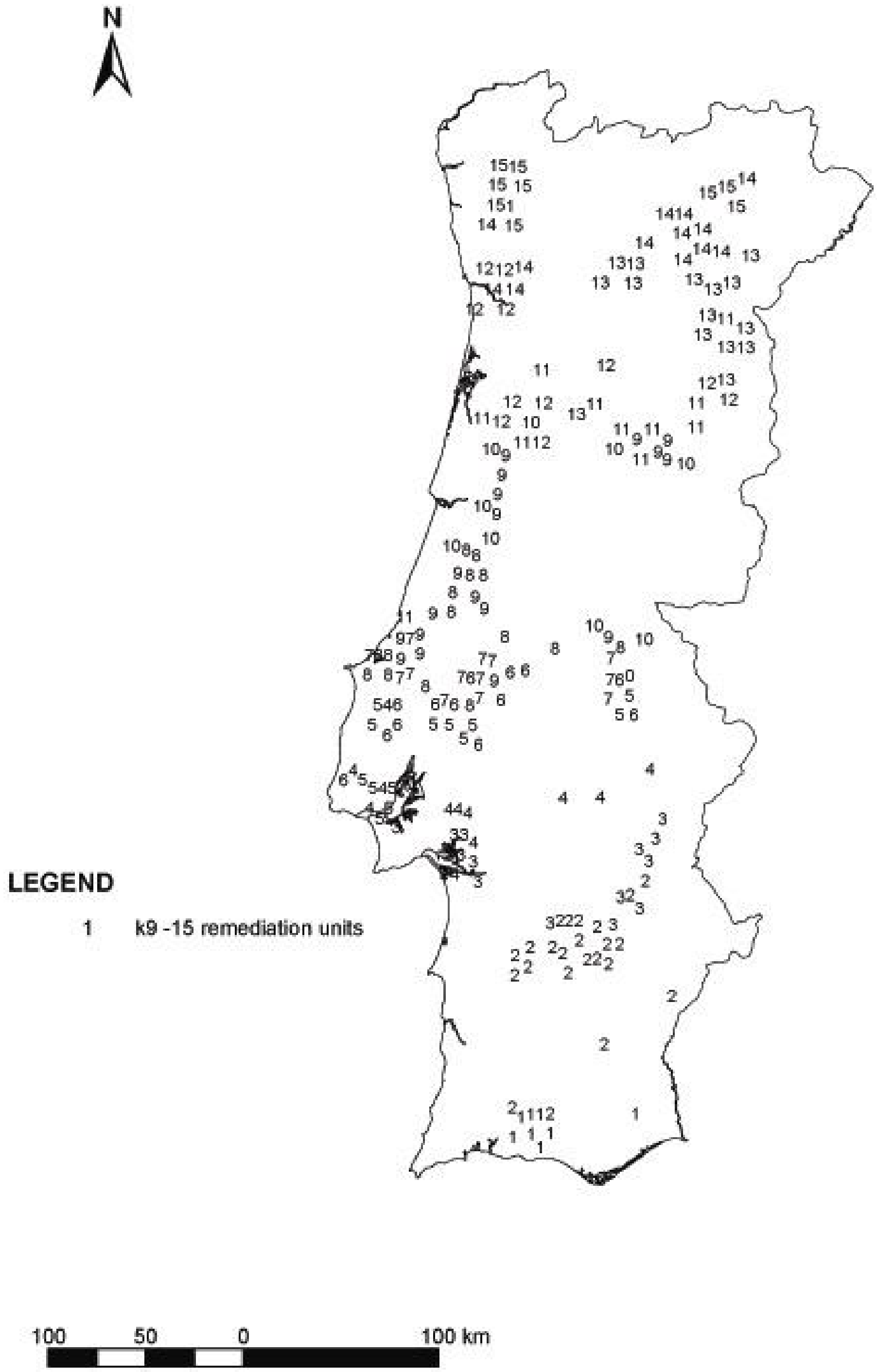

Figure III. 30 – Results from Test 2, for 2020, using k-means and considering 15 remediation units.

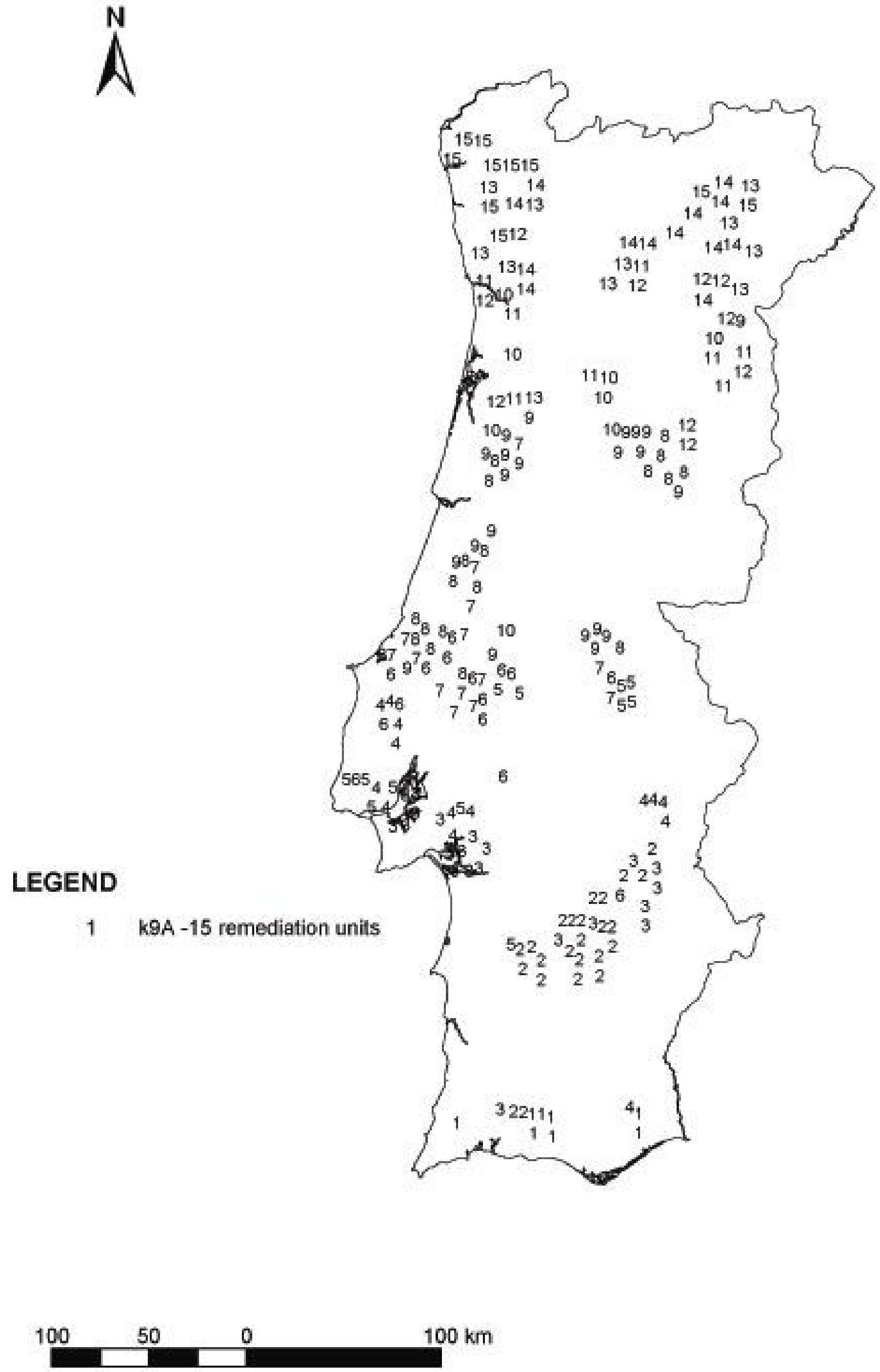

Figure III. 31 – Results from Test 2, for 2020, using k-means and considering 15 remediation units and 250 epochs.