UNIVERSIDADE NOVA DE LISBOA

Faculdade de Economia

BOUND IMPROVING SEQUENCES: A TOOL FOR DISCRETE PROGRAMMING*

Paulo Bárcia

Working Paper nº 18

*Trabalho elaborado em colaboração com o projecto I&D nº 452.82.41 da Junta Nacional de Investigação Científica e Tecnológica, em curso nesta Faculdade.

UNIVERSIDADE NOVA DE LISBOA
Faculdade de Economia
Campo Grande, 185
1 700 LISBOA

Abril, 1984

Table of contents

1. HEADINGS

1.1. TITLE

Bound improving sequences:A tool for discrete programming

1.2. AUTHOR

Paulo Barcia

1.3. AFILLIATION

Electricidade de Portugal,Network Planning Department and Universidade Nova de Lisboa,Computer Science Department

1.4. ABSTRACT.

The purpose of this note is to report a new tool for discrete programming:Bound improving sequences.

It consists on the construction of a sequence of bounds that, under appropriate conditions,converges in a finite number of steps to the optimal value of the objective function of the problem studied.As a byproduct an optimal solution for that problem is produced.

For the case of 0-1 LP's such a sequence can be efficiently computed.Examples,geometric interpretations and computational experience reports for this case are given.

1.5. KEYWORDS

Lower Bounding,Lagrangean Relaxation,Knapsack Problem

1.6. ACKNOWLEGEMENT

I am indebted to Prof.J.D.Coelho of Universidade Nova de Lisboa Faculty of Economics for providing the Martello and Toth software and constructive criticism.

any lower bound on v

$$
\begin{aligned}
v = \min\ & f(x) \\
& f(x) \geq l(k) \\
& g(x) \geq b \\
& x \in X
\end{aligned}
\qquad (2)
$$

(2) is obviously equivalent to (1), in the sense that both problems yield the same optimal values. Take the following Lagrangean dual of problem (2)

$$
l(k+1) = \max_{d \geq 0} \ \min_{\substack{f(x) \geq l(k) \\ x \in X}} [f(x) + d(b - g(x))] \qquad k = 0, 1, \text{etc} \qquad (3)
$$

where $d \in R^m$ is a vector of multipliers.

In this section we shall prove that, under suitable assumptions, formula (3) defines a bound improving sequence (BIS), i.e. $\lim l(k) = v$, and produces for $l(k) = v$, an optimal solution for problem (1) thus becoming a bound improving sequence algorithm (BISA).

Those results can be easely established if we assume some hypothesis on f( ), g( ) and X, as follows:

X will be supposed a discrete set of points of finite cardinality and we shall further assume that f( ) and g( ) are bounded on X and continuous.

However a less trivial hypothesis is needed.

Consider the level sets of f( ), i.e., $L(p) = \{x \in X,\ f(x) = p\}$, let $W(p) = \{q = b - g(x),\ x \in L(p)\}$ and call $Co[W(p)]$ the convex hull of W(p).

We shall assume that, for any $p < v$, the following holds

$$
\exists\ y \leq 0,\ y \in Co[W(p)] \Rightarrow \exists\ q \leq 0,\ q \in W(p) \qquad (4)
$$

Note that for m=1 or if $\forall p < v$ W(p), if not empty, is a singleton the hypothesis is fullfiled.For practical purposes it is allways possible to force every non empty W(P) into a singleton by forcing L(P) into one introducting a small perturbation in f( ), so the fullfilment of (4) must not wory us if we are application oriented although it may be a serious limitation from the theoretical standpoint.

Before stating and proving the main result of this section we shall introduce some new notation rewriting formula (3) as follows

$$l(k+1)=F[l(k)]=\max_{d\geq 0} G[d,l(k)] \qquad (5)$$

where

$$G[d,l(k)]= \min_{\substack{f(x)\geq l(k)\\ x\in X}} [f(x)+d(b-g(x))] \qquad (6)$$

We also note that X being a finite set of points is such that f(X) is also a finite set of points in R.We will define $f(\overline{X})$, an finite ordered set of real numbers ,as f(X) extended whith $-\infty$ and $+\infty$

$$f(\overline{X})=f(X)\cup\{-\infty,+\infty\} = \{p(0),p(1),\cdots,p(K)\}$$

with $p(0)=-\infty$, $p(K)=+\infty$ and $p(k-1)<p(k)$ for $1\leq k\leq K$.

We can now state the main result of this section as follows

THEOREM 1:Under the assumptions stated above and if $l(0)\leq v$ then $\lim l(k)=v$.Furthermore one of the solutions of the problem defining G[0,v] is an optimal solution for problem (1).

The proof of the theorem relies on some intermediate results concernning the properties of F[ ] that we state and prove now.

LEMA 1: $\forall p \in R$ $F[p] \geq p$

PROOF: It is clear from (6) that $G[0,p] \geq p$ and this implies

$$F[p] = \max_{d \geq 0} G[d,p] \geq p \qquad //$$

LEMA 2: $\forall p \in R$ $F[\ ]$ is lower semi continuous at p.

PROOF: As $f(X)$ is a finite ordered set of real numbers there must exist $p(k-1)$ and $p(k)$ in $f(\overline{X})$ such that $p(k-1) < p \leq p(k)$, $k \leq K$.

This means that $D[p] = \{x \in X\ ,\ f(x) \geq p\}$, the set of feasible solutions for problems defining $G[d,p]$, remains constant for $p \in (p(k-1), p(k)]$ and so the same happens to $G[d,p]$. In fact we have $\forall p \in (p(k-1), p(k)]$ $G[d,p] = G[d,p(k)]$.

This implies that $\forall \tilde{p}$ $(p(k-1), p]$ $F[\tilde{p}] = F[p] = F[p(k)]$.

The fact that $F[\ ]$ remains constantly equal to $F[p]$ on a non empty interval $(p(k-1), p]$ implies the lower semi continuity of $F[\ ]$ at p//.

We are now able to prove theorem 1.

PROOF OF THEOREM 1:

It is clear that lema 1 and the fact that $l(k)$ is bounded above by v show that $l(k)$ converges to a limit l such that $\lim l(k) = l \leq v$.

We note now that lema 2 and the fact that $l(k)$ is a nondecreasing sequence imply $l = \lim F[l(k)] = F[\lim l(k)] = F[l]$.

In other words, lema 2 implies that the limit l is a fixed point of $F[\ ]$.

Now if we prove that v is the only fixed point of $F[\ ]$ on $(-\infty, v]$ the first part of the theorem holds. We prove first that v is a fixed point of $F[\ ]$.

Lema 1 shows that $F[v] \geq v$ but for any $p \in (-\infty, v]$ $F[p]$, being the optimal value of a Lagrangean dual of (2), is a lower bound on v, so we have $\forall p \in (-\infty, v]$ $F[p] \leq v$, and this implies that v is a fixed point of F[ ].

Now suppose that $\exists p<v$ such that $F[p]=p$.

This would imply that $\forall d \geq 0$ $G[d,p] \leq p$, in particular it shows that $G[0,p]=p$. This means that the maximum of $G[d,p]$ over $d \geq 0$ must be at $d=0$ thus there must not exist ascending feasible directions for $G[d,p]$ at $d=0$. The feasible directions at 0 are directions $d \geq 0$. The ascending directions at 0 are directions d such that $0<\min dy, y \in Co[W(p)]$, Bazaraa[2] pp. 187-196. Then if we have no ascending feasible directions we must have

$$\max_{d \geq 0} \min_{y \in Co[W(p)]} dy \leq 0 \qquad (7)$$

Using Lagragean duality it is trivial to show, from (7), that the set

$\{y \leq 0,\ y \in Co[W(p)]\}$ is not empty and then (4) implies that $\exists q \leq 0, q \in W(p)$. Call $x[0,p]$ the point of $L(p)$ that yields $q=b-g(x[0,p]) \leq 0$.

We have thus found a point $x[0,p]$ feasible for Problem (1) such that $f(x[0,p])=p<v$, a contradiction. Then v is the only fixed point of F[ ] and the first part of the theorem holds.

To prove the second part of the theorem one should repeate the proof of part one using v instead of p. That would lead us to find a point $x[0,v]$ such that $f(x[0,v])=v$ and $g(x[0,v]) \geq b$ thus concluding the proof of the theorem//.

We have proved now that $l=v$ and that one of the optimal solutions for the problem defining $G[0,v]$ is an optimal solution for Problem (1). Note incidentaly that although v is the only fixed point of F[ ] on $(-\infty,v]$ F[ ] is not neceserely a contraction map so the Banach theorem on contraction maps does not apply to this situation inspite we have $\lim l(k)=v$. We shall prove now a final result concernning the number of iterations of formula (3) needed to attain v.

THEOREM 2: The limit $l=v$ will be attained using a finite number of iterations of formula (3).

PROOF: Proving lema 2 we established that $\forall p \in R$ $F[\ ]$ remains constantly equal to $F[p]$ in a non empty interval $(p(k-1),p]$, for some $1 \leq k < K$. This means that there exists $1 \leq k < K$ such that for p in $(p(k-1),v]$ we have $F[p]=v$. As $l(\tilde{k})$ is nondecreasing and converging to v this certainly implies that $F[l(\tilde{k})]=v$ $\forall \tilde{k} \geq \tilde{K}$, for some finite $\tilde{K}$, and this concludes the proof of the theorem//.

Theorem 1 and theorem 2 define the BISA as follows: Start with any $l(0)<v$, say $l(0)=-\infty$. Use formula (3) until $l(k)$ converges, in a finite number of steps, to v. One of the optimal solutions for the problem defining $G[0,v]$ is then optimal for problem (1).

BISA has two main drawbacks. The first, is the obvious fact that formula (6) will be, for the general case, very hard to compute. However for a particular, yet important, instance of problem (1), 0-1 LP'S, efficient computation can be achieved as we shall see in next section.

The second drawback is related to the fact that we may not be able to compute v exactly, either because a large number of iterations of formula (3) is needed or because the algorithm used to compute formula (5) is not acurate enought. However $l(k)$ is a sequence of lower bounds on v converging to v so at any moment a switch to a branch and bound procedure to solve problem (1) is possible with an improved lower bound. Finaly we note that if (4) does not hold we may have $l<v$. Indeed if (4) is not fullfiled the possible values for l are v toghether with the values $p<v$ such that $W(p)$ does not satisfy (4).

2.3. 0-1 LP PROBLEMS

Consider the following 0-1 LP problem and, with no loss of generality, suppose it has positive cost coefficients

$$\min cx$$
$$Ax \geq b \qquad (8)$$
$$x \in X$$

where $x \in R^n$, $c \in R^n$, $c \geq 0$, $b \in R^m$, $X = \{0,1\}^n$ and A is a m*n real matrix.

This sort of problem is generaly solved by branch and bound procedures where sharp lower bounds are badly needed. For this particular case formula (3) becomes

$$l(k+1) = \max_{d \geq 0} \min_{\substack{cx \geq l(k) \\ x \in X}} [cx + d(b - Ax)] \qquad k=0,1, \text{etc} \qquad (9)$$

Note now that the inner minimization problem is a single 0-1 knapsack problem wich can be efficiently solved using Martello[3] and Toth[3][4] techniques .

A computer code using theese procedures for the inner minimization problem and sub-gradient optimization for the outer maximization was developed for the computation of formula (9).

In the remainder of this section we shall report the computational behaviour of this code on several types of problems

2.3.1. Pure Random Problems

Problems were randomly generated as follows: A and c coefficients were uniformely random generated in [0,1]. The b coefficients were obtained by summing up the corresponding row of A and then multiplying the result by an uniformely random number in [0,P].The P values used were the following:

(1)Problems type 'T'(P=1.0).These problems are likely to be tightly constrained in the sense that a large number of x_i will be equal to 1 in the optimal solution.

(2)Problems type 'MT'(P=.5).Medium tight problems.Around half of the variables will be 1 in the optimal solution

(3)Problems type 'ML'(P=.1).Medium loose problems.About 1/10 of the variables will be 1 in the optimal solution

(4)Problems type 'L'(P=.025).Loose problems.Only a few variables will be 1 in the optimal solution.

For these 4 problem types TABLE 1 shows BISA performence on problems with dimensions ranging from 20*40 up to 50*100.

The l(0) value used was allways l(0)=ZLP(LP bound).The algorithm was stoped as soon as a 1/10000 relative precision on the estimated value of l was attained.

Looking at TABLE 1 it is clear that BISA is likely to work well on almost every type of problem.

It is also clear that performances tend to degradate as m increases.This is most likely due to the fact that,in the present implementation,the search for the optimum multiplyers in each iteration is done using a crude projected subgradient procedure wich is not very acurate.This implies that more iterations are needed and false convergences are likely to be detected.One can expect that the code will become more performant if a better non-differentiable optimization algorithm is used for this task.

2.3.2. Petersen Test Problems

BISA was used on the 7 test problems described in [1].The results are shown in table 2.In problems 1 to 5 an exact solution was found and no perturbation scheme was needed. In problems 6 and 7 the perturbation scheme was used however premature convergence was detected soon after ZLP. This misbehaviour is most likely due to the following: In solving the generated knapsack problems using Martello and Toth algorithm data must be converted to integer. This conversion must be done in such a way that the optimal value of the converted problem is less or equal the optimal value of the true problem so that we can be sure that we are always generating lower bounds, i.e. that we never have l(k)>v because of data conversion. This procedure is likely to introduce some lack of accuracy causing premature termination in some problems.

2.3.3. Problems With Data Dependencies.

We have also examined BISA behaviour on problems where constraint data has some sort of dependency upon the cost row.

We have thus generated problems as follows:

The c coefficients were randomly generated in [0,1]. The A rows were generated by adding to the corresponding c value a random number in [-.1,.1].

$$A(i,j)=c(j)+r(i,j) \qquad r(i,j) \in [-.1,.1]$$

So there is a strong dependency between c and the rows of A as they will be 'near' parallel. The b coefficients were generated as in pure random problems and so we will also have problems type 'T' ,'MT', 'ML' and 'L'.

Table 3 reports computational experience with those problems. BISA performances are not as good as in pure random problems . One should however note the following:

In 1/3 of the cases studied the exact solution was directly found.

In 1/2 of the cases studied althought an exact solution was not found directly by BISA due to premature convergence a significant lower bound improvement was made enabling a branch and bound exact solution that was not possible using ZLP.

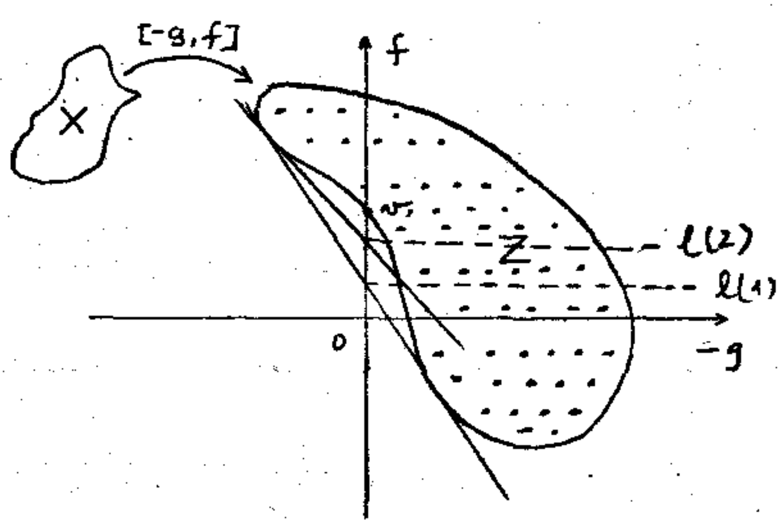
In every case lower bound improvements were made enabling even on the worst cases a more accurate judgement on the distance to optimality of the best solution found by branch and bound.

2.4. GEOMETRIC INTERPRETATIONS

We deal first with a simple geometric interpretation of BISA that will give us a deeper insight on the way formula (3) operates.

For the sake of simplicity we shall assume that m=1 and b=0. The doted zone on figure 1 ,Z,represents the image of X by the [-g,f] map.

Figure 1.



If we start using formula (3) with a sufficiently low l(0), say l(0)=-∞ ,computing l(1) will be equivalent to the usual

Lagrangean dual of problem (1).But as we impose the extra condition $f(x) \geq l(1)$ we are cutting a part of Z thus enabling better relaxations that yield $l(2)$.

This procedure will go on until v is attained . Then the relaxation with zero slope ,among others,will yield $x[0,v]$ the optimal solution for problem (1).

We shall deal now with a more farfetched geometric interpretation for the case of 0-1 LP's.We shall do it by looking at the following example: Min [$x+y$, $x+y \leq 2$, $x \geq 1/2$, $y \geq 1/2$, $x,y \in \{0,1\}$].

To ilustrate the behaviour of BISA on the example one must change a bit the 'looks' of formula (9).

Note that the inner minimization problem can be written

$$\begin{aligned} &\min \; cx + d(b-Ax) \\ &x \in \mathrm{Co}[\, cx \geq l(k),\; x \in X] \end{aligned} \qquad (10)$$

where Co stands for 'convex hull'.This being so we have

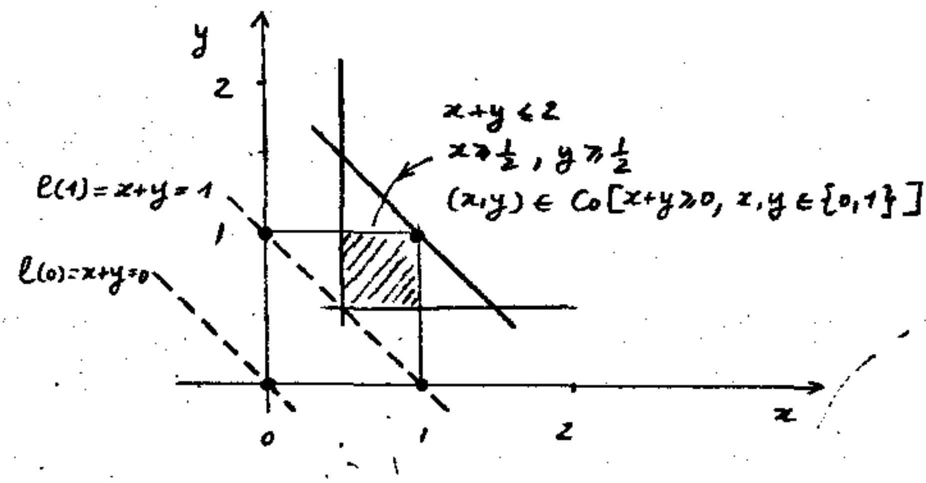
$$\begin{aligned} l(k+1) = \min \; & cx \\ & Ax \geq b \\ & x \in \mathrm{Co}[\, cx \geq l(k),\; x \in X]. \end{aligned} \qquad (11)$$

This 'form' of the BISA definition has nothing new and just translates the known fact that for 0-1 LP's Lagrangean relaxation is equivalent to the convexification of the non-relaxed constraints.However it will enable us to see, geometricaly,how BISA behaves for the above example.For that case we shall have

$$\begin{aligned} l(k+1) = \min \; & x+y \\ & x+y \leq 2 \\ & x \geq 1/2 \;,\; y \geq 1/2 \\ & x,y \in \mathrm{Co}[\, x+y \geq l(k),\; x,y \in \{0,1\}]. \end{aligned}$$

Suppose we start with a low l(0), say l(0)=0 . Finding l(1) will be equivalent to solve the problem shown in fig 2 where the shaded zone is the feasible region.

Figure 2.

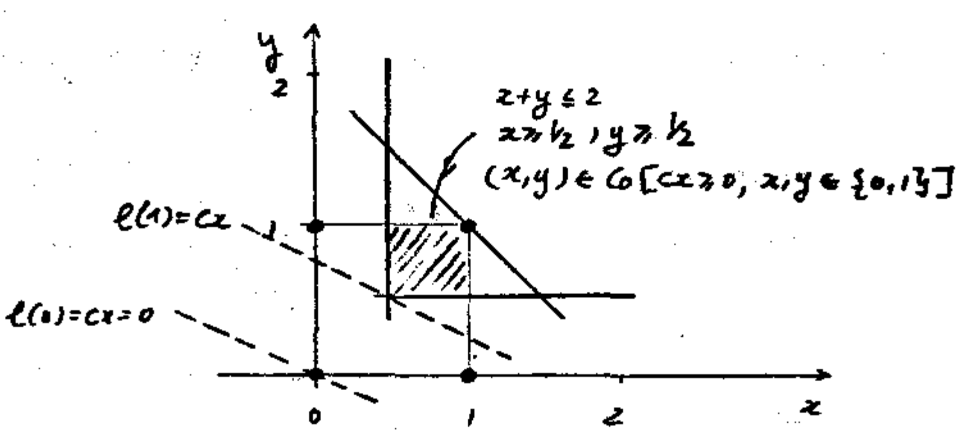


We will obtain, of course, the solution x=y=1/2 and l(1)=1. If we try to find l(2) we will be stopped because the objective function x+y violates the hypothesis upon wich the convergence proof relies.

But this is the only case for all the objective functions possible for the example.

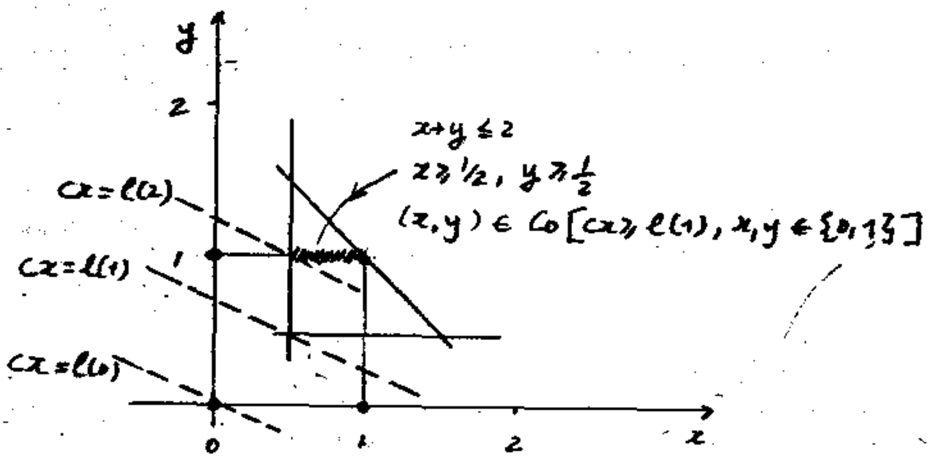
To see this consider fig 3 where the computation of l(1) is represented with an arbitrary objective function different from x+y, noted cx on the figures.

Figure 3.



Now when computing l(2) the points (0,0) and (1,0) will be removed and this computation is shown in fig 4 where the dark line stands for the feasible region for the problem defining l(2).

Figure . 4.



It is clear now that ,when computing l(3) the point (0,1) will be also removed and the optimal solution will be attained.

2.5. CONCLUSIONS

Bound improving sequences appear likely to be an usefull computational tool to solve some classes of discrete programming problems, namely 0-1 LP's.

The efficiency of of the algorithm will hopefully be improved if a twofold action is taken as future research direction:

First, computational experience should be carried on with other knapsack and nondiferencial optimization routines.An improvement in those routines should have dramatic effects upon the efficiency of the present implementation of the code.

Last, but not least, further research effort should be made in the following sense: Note that formula(9) can be viewed as a sort of cutting plane approach where cutting planes $cx \geq l(k)$ are inbeded in a dual formulation. This being so one should expect that the use of deeper dual cuts should greatly improve the algorithm's efficiency. This is likely to be a very promissing research area on this matter.

2.6. REFERENCES

[1] Petersen, C.C. 'Computational experience with variants of the Balas algorithm applied to the selection of R&D projects', Management Science, Vol 13, no 9, May 1967

[2] Bazaraa, M. Shetty, C 'Nonlinear Programming', Willey 1979

[3] Martello, S. Toth, P 'An upper bound for the 0-1 knapsack problem and a branch and bound algorithm', European Journal of Operations Research, Vol 1, 169-175, 1977

[4] Toth, P 'A new reduction algorithm for the 0-1 knapsack problem' ORSA/TIMS joint national meeting, MIAMI 1976.

3. APPENDIX (Tables)

Table 1. PURE RANDOM PROBLEMS

P num	m	n	type	zlP	l	v	l/v	nit	t
1	20	40	t	18.2282	18.5299	18.5299	1.000	10	181
2	20	40	Mt	4.7792	4.8909	4.9317	0.992	9	300
3	20	40	Ml	.2488	.2866	.2866	1.000	4	222
4	20	40	L	.0644	.1706	.1706	1.000	9	282
5	20	75	T	29.7852	29.9747	29.9747	1.000	11	563
6	20	75	Mt	9.0787	9.1278	9.6179	0.949	7	253
7	20	75	Ml	.3400	.3971	.3971	1.000	3	133
8	20	75	L	.0568	.0681	.0681	1.000	3	266
9	20	100	T	40.6423	40.7094	40.7094	1.000	5	559
10	20	100	Mt	9.7750	9.9395	11.1719*	0.889*	5	231
11	20	100	Ml	.6394	.6818	.6933	0.983	5	337
12	20	100	L	.0857	.1181	.1181	1.000	5	558
13	35	100	T	40.5431	40.8557	40.8557	1.000	8	713
14	35	100	Mt	9.8796	9.9142	11.2350*	0.882*	4	754
15	35	100	Ml	.4790	.4888	.5396*	0.906*	2	583
16	35	100	L	.0804	.1178	.1178	1.000	3	739
17	50	100	T	42.6104	43.1177	43.1177	1.000	42	3723
18	50	100	Mt	11.0487	11.0798	12.1413*	0.912*	5	619
19	50	100	Ml	.4894	.5625	.5625	1.000	3	643
20	50	100	L	.0941	.1237	.1237	1.000	3	481

Table 2. PETERSEN TEST PROBLEMS

P num	m	n	type	zlp	l	v	v/l	nit	t
1	10	6	-	4133.1	3800.0	3800.0	1.000	3	3
2	10	10	-	9294.8	8706.1	8706.1	1.000	13	11
3	10	15	-	4125.0	4015.0	4015.0	1.000	9	16
4	10	20	-	6152.1	6120.0	6120.0	1.000	3	51
5	10	28	-	12459.2	12400.0	12400.0	1.000	3	9
6	5	39	-	10672.3	10670.6	10618.0	0.995	2	131
7	5	50	-	16612.8	16608.2	16537.0	0.995	1	53

Table 3. PROBLEMS WITH DATA DEPENDENCIES

P num	m	n	type	zlp	l	v	l/v	nit	t
1	20	40	t	18.9694	19.0045	19.0045	1.000	4	14
2	20	75	T	34.5634	34.6196	34.6196	1.000	5	30
3	20	100	T	50.2563	50.2849	50.2849	1.000	8	351
4	20	40	Mt	4.1249	4.4410	4.5876	0.968	11	80
5	20	75	Mt	8.2314	8.4302	8.8432*	0.954*	10	533
6	20	100	Mt	10.9088	11.0054	11.0054	1.000	7	549
7	20	40	Ml	1.5547	1.6577	1.6687	0.993	3	163
8	20	75	Ml	2.3391	2.4246	2.5019*	0.970*	10	260
9	20	100	Ml	3.5177	3.5759	4.0631*	0.880*	11	491
10	20	40	L	.3375	.4016	.4673	0.859	7	73
11	20	75	L	.5735	.6307	.6440	0.979	5	59
12	20	100	L	.8884	1.1484	1.1599	0.990	3	163

Nit- number of BISA iterations

t - computer time, in seconds, for a SPERRY 1100/80 machine

* - best value found after 15 minutes of branch and bound