INVESTIGAÇÃO OPERACIONAL, vol. 12 (Junho 1992) 43-50

LINEAR AND LAGRANGEAN PENALTIES FOR VARIABLE BOUNDING IN ILP: AN APPLICATION TO A COVERING PROBLEM

Margarida Vaz Pato
Departamento de Matemática
Instituto Superior de Economia e Gestão
Rua Miguel Lupi, 20
1200 Lisboa - Portugal

Abstract

Lagrangean and linear penalties can be used for variable bounding in ILP. Such penalties, embedded in a branch-and-bound algorithm, yield remarkable reductions in the search procedure effort for large scale problems.

In this paper, four different ways of exlporing this idea for a covering problem with integer variables are presented. Computing results taken from test problems have revealed the efficiency of the technique in reducing the amplitude of variable intervals, and even in fixing them at feasible values.

Resumo

Num problema de PLI com variáveis limitadas os intervales de variação das variáveis podem ser "apertados" tomando em consideração penalidades lagrangeanas ou lineares, as quais vêm assim facilitar a enumeração implicita em PLI, especialmente se os problemas forem de grande dimensão.

Para um problema de cobertura com variáveis inteiras limitadas apresentam-se quatro alternativas de aplicação desta ideia, bem como os resultados computacionais obtidos num conjunto de problemas-teste. A experiência evidenciou a eficácia da técnica na fixação de variáveis e na redução, percentualmente elevada, da amplitude dos intervalos.

Keywords

integer linear programming, covering problems, lagrangean relaxation, penalties.

1 – Introduction

The purpose of this paper is to show how lagrangean and linear relaxations can be used for variable bounding in integer linear problems with bounded variables (ILPs). Two general results, stated in Section 2, validate the updated variable bounds for all the feasible solutions better than a stipulated one. Such features have been called lagrangean or linear penalties though they do not lead to penalties in the strict sense of the word. This nomenclature arises from the penalization incurred when the solution goes outside the updated variable interval.

In Section 3, we shall point to different ways of defining lagrangean penalties for a covering problem with integer variables, referred to as the generalized set covering problem (Paixão and Pato (1988)). Finally, some computational results will be given to ilustrate that penalties introduce remarkable reductions in the solution space, at least in the covering problems tested so far.

Therefore, as this technique is very inexpensive computationalwise, it is most suitable for accelerating the bounding procedure in large scale problems.

Mention should be made of the following works, where lagrangean or linear penalties were applied to several problems: Sweeney and Murphy (1981) – the multi-item scheduling problem; Balas and Ho (1980), Hey (1980), Paixão (1983), Beasley (1987) – the set covering problem; Chan, Bean and Yano (1987) – the set partitioning problem; Hall and Hochbaum (1985) – the multi-covering problem. It may be worth noting that, in these studies, the updating of the variable bounds does no more than fix the variables at 0 or 1, because the problems are binary. In the case of the ILP, Nemhauser and Wolsey (1988) present a linear penalty result which is weaker than the one stated here. These authors apply the penalties to a binary problem – the travelling salesman problem. In fact, the possibilities for bounding are much more relevant in binary problems.

As is known similar penalties have already been derived from the linear relaxation by Beale and Small (1965), Driebeek (1966) and Tomlin (1971). However such post-linear penalties are extremely time-consuming as they require the optimal linear tableau.

2 – Linear and Lagrangean Penalties for ILP

Let us take the following integer linear problem with bounded variables:

$$(\text{ILP}) \qquad \begin{array}{ll} \text{minimize} & \mathbf{c}\, x \\ \text{subject to} & \mathbf{A}\, x \geq \mathbf{b} \\ & \mathbf{0} \leq x \leq \mathbf{h} \\ & x \text{ integer} \end{array}$$

where $\mathbf{c}$ is a real vector, $\mathbf{b}, \mathbf{h}$ are integer vectors, $\mathbf{A}$ is an integer matrix, and all have appropriate dimensions.

As we know, the linear relaxation of the integer linear problem is the ILP without the integrality constraints, and is denoted by LinILP.

Now let $\bar{z}$ be an upper bound on the optimal value of ILP, z_{ILP}, and $[\mathbf{u}\ \mathbf{v}]$ a feasible solution for the dual linear problem. As usual, $\mathbf{r}$ represents the reduced costs associated with $[\mathbf{u}\ \mathbf{v}]$ and $\underline{z}$ the dual objective value corresponding to $[\mathbf{u}\ \mathbf{v}]$.

Following these definitions, consider the general result for linear penalization which can be directly taken from duality:

Proposition 1

For a particular k,

- if $r_k > 0$ then $x_k \leq \rfloor (\bar{z} - \underline{z})/r_k \lfloor$

  in any feasible solution better than $\bar{z}$.

  ($\rfloor a \lfloor$ – the largest integer strictly less than a)

Now, rather than rejecting the integrality constraints, some of the main constraints are rejected by embedding them in the objective function associated with non-negative multipliers, $\lambda \geq \mathbf{0}$. Thus, this lagrangean relaxation of ILP may be presented as follows:

$$(\text{Lag}_\lambda\text{ILP}) \qquad \begin{array}{ll} \text{minimize} & \mathbf{c}\,x - \lambda\,(\mathbf{A}^2 x - \mathbf{b}^2) \\ \text{subject to} & \mathbf{A}^1 x \geq \mathbf{b}^1 \\ & \mathbf{0} \leq x \leq \mathbf{h} \\ & x \text{ integer} \end{array}$$

where the vectors and matrices have appropriate dimensions.

The decomposition of the constraint set into two blocks, $\mathbf{A}^1 x \geq \mathbf{b}^1$ and $\mathbf{A}^2 x \geq \mathbf{b}^2$, is performed in such a way that $\text{Lag}_\lambda\text{ILP}$ possesses the integrality property (Geoffrion (1974)). In an extreme case, constraints $\mathbf{A}^1 x \geq \mathbf{b}^1$ may not exist. Thus, all constraints, with the exception of the bounding ones, are relaxed.

Once more, let $\overline{z}$ stand for an upper bound on z_{ILP}. Let $\text{Lag}_\lambda\text{ILP}$ represent the optimal value for $\text{Lag}_\lambda\text{ILP}$. Here vector $\hat{r}$ stands for the reduced lagrangean costs defined through the expression

$$\hat{r}_j = c_j - \lambda\,\mathbf{A}^{2j} - \mathbf{u}^*\,\mathbf{A}^{1j}$$

where λ is the vector of lagrangean multipliers and $\mathbf{u}^*$ is the vector of optimal dual linear variables corresponding to the constraints $\mathbf{A}^1 x \geq \mathbf{b}^1$ in $\text{Lag}_\lambda\text{ILP}$.

One may, at this point, derive the second result for lagrangean penalization:

Proposition 2

For a particular k,

- if $\hat{r}_k > 0$ then $x_k \leq \rfloor (\overline{z} - z_{\text{Lag}_\lambda\text{ILP}})/\hat{r}_k \lfloor$ (case 1)
- if $\hat{r}_k < 0$ then $x_k \geq \rbrack\, h_k + (\overline{z} - z_{\text{Lag}_\lambda\text{ILP}})/\hat{r}_k \lbrack$ (case 2)

in any feasible solution better than $\overline{z}$.

($\rbrack a \lbrack$ – the smallest integer strictly greater than a)

Proof

The main ideas involved are as follows.

In the first case, where $\hat{r}_k > 0$, we include the following additional constraint in ILP

$$x_k \geq (\overline{z} - z_{\text{Lag}_\lambda\text{ILP}})/\hat{r}_k$$

which produces the enlarged problem, ILP^E.

Firstly, the constraints $\mathbf{A}^2 x \geq \mathbf{b}^2$ are relaxed and associated with any vector of non-negative multipliers $\lambda \geq \mathbf{0}$. Then we find problem $\text{Lag}_\lambda\text{ILP}^E$.

The other constraints, $\mathbf{A}^1 x \geq \mathbf{b}^1$, are now relaxed by taking $\mathbf{u}^*$ as multipliers, which leads to:

$$(\text{Lag}_{u^*}\text{Lag}_\lambda\text{ILP}^E) \qquad \begin{array}{lll} \text{minimize} & \mathbf{c}\,\mathbf{x} - \lambda\,(\mathbf{A}^2 x - \mathbf{b}^2) - \mathbf{u}^*\,(\mathbf{A}^1 x - \mathbf{b}^1) & \\ \text{subject to} & 0 \leq x_j \leq h_j \text{ and integer} & (j \neq k) \\ & (\overline{z} - z_{\text{Lag}_\lambda\text{ILP}})/\hat{r}_k \leq x_k \leq h_k. & \end{array}$$

The optimal value for the enlarged doubly-relaxed problem verifies:

$$z_{Lag_u*Lag_\lambda ILPE} = z_{Lag_u*Lag_\lambda ILP} + \hat{r}_k\,(\bar{z} - z_{Lag_\lambda ILP})/\hat{r}_k =$$

and thus, from the integrality of the involved problems, $Lag_u Lag_\lambda ILP$ and $Lag_\lambda ILP$, and the specific selection of multipliers for the last relaxation, $\mathbf{u}^*$, the following equality arises:

$$= z_{Lag_\lambda ILP} - \bar{z} - z_{Lag_\lambda ILP} = \bar{z}.$$

Therefore, by applying the definition of lagrangean relaxation we have

$$z_{ILPE} \geq \bar{z}$$

that is, the additional constraint must be violated in a solution (to the ILP) whose value is greater than $\bar{z}$. And from the integrality of this variable, x_k, one may immediately conclude the validity of the updated variable bound presented in this proposition (case 1).

For (case 2) the proof is similar. ♦

Returning to the two propositions stated above, we can see that penalties provide some kind of valid inequalities for the feasible solutions better than $\bar{z}$.

As has already been seen, these results simply require an upper bound on the optimal value of the ILP, together with the by-products of a lagrangean relaxation (Proposition 2), or a feasible dual linear solution (Proposition 1). They may therefore be easily computed.

For the above reasons such penalties may be used for variable bounding purposes in the case of large scale ILPs.

3 – Linear and Lagrangean Penalties for the Generalized Set Covering Problem

Let us now take a particular ILP which is a covering problem, the so-called generalized set covering problem, in short GSCP. It is expressed in the same way as the ILP, where matrix **A** is a binary one.

Several relaxations were developed and tested on the GSCP (Paixão and Pato (1988)).

One of them is the linear relaxation. Another corresponds to the relaxation of all covering constraints in a lagrangean fashion. The corresponding relaxed problem is trivially solved by inspection.

Two different lagrangean relaxations were also specially designed for the scheduling applications of the GSCP, taking into account the structure of such instances. One of them, the structural relaxation, includes a reformulation of the GSCP with redundant constraints. It then relaxes some of the original constraints. The other one, the decomposition relaxation, comes from a standard use of variable splitting. Both relaxed problems can be efficiently solved by a minimum cost network flow code.

Appropriate penalties were deduced from each of the four relaxations mentioned. These penalties can be easily calculated through a heuristic-lagrangean procedure such as the following:

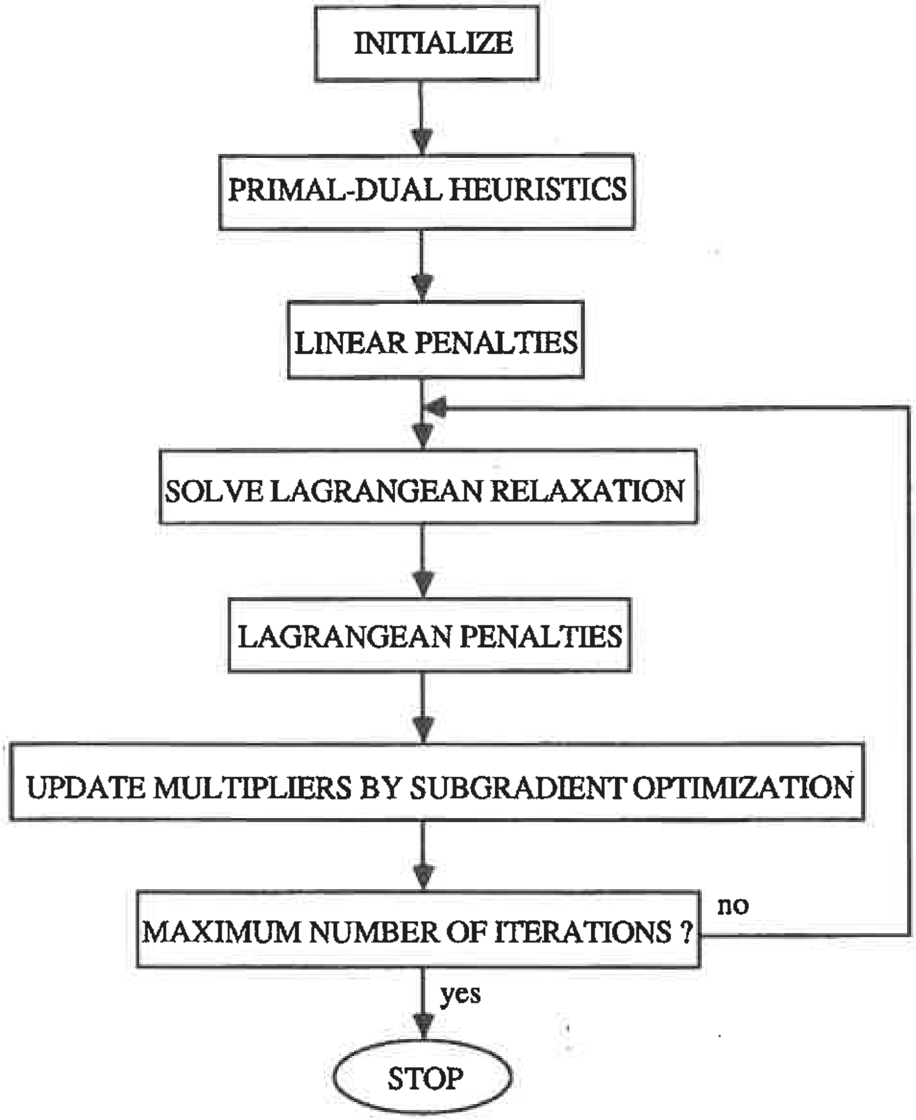

Figure 1 – Heuristic-Lagrangean Procedure

This heuristic-lagrangean algorithm with penalties (Figure 1) was devised to bound the optimal value of the GSCP. In our case, it was used not only in the root node of a branch-and-bound, but also – again for bounding purposes – in every node of the branching procedure. It should be noted that in this algorithm the calculation was truncated when a maximum number of generated nodes was reached. Such a combined procedure was tested in a set of medium-sized GSCP instances. Some of them were taken from literature whereas others were not. They were either derived through random generation, according to different patterns, or from scheduling real applications.

Structural relaxation was the most favourable of the three lagrangean relaxations in this experiment.

Computing results for the above truncated branch-and-bound using structural relaxation are summarized in the following table.

Table 1 – Summary of Typical Cases (Combined Procedure)

probl. (1)	dimensions density (2)	type references (3)	effect a penalties node zero fixed (4)	reduction (5)	relative behaviour (6)
RN18	36 × 865 56%	scheduling applications Pato (1989)	587	86%	BETTER UPPER BETTER CPU
RN21	36 × 865 56%	idem	731	51%	BETTER UPPER BETTER CPU OPTIMAL SOLUTION
RN7	46 × 1500 45%	idem	103	80%	BETTER UPPER BETTER LOWER BETTER CPU
RNPA	41 × 378 52%	idem	157	69%	BETTER LOWER BETTER CPU OPTIMAL SOLUTION
CRF1	47 × 333 39%	idem	18	83%	BETTER LOWER BETTER CPU
RN12	36 × 865 56%	idem	580	94%	BETTER CPU
HAH2	50 × 200 5%	random generation Hall and Hochbaum (1985)	144	10%	WORSE UPPER BETTER CPU
MOR1	168 × 168 24%	scheduling applications Morris and Showalter (1983)	0	0%	—

This table sets out to demonstrate the effects of linear plus structural lagrangean penalties by showing only one of each typical case found throughout the experiment. This small set of problems (see column (1)) thus ilustrates all the relevant situations found so far.

Column (2) indicates the dimensions and density of the problems, whereas column (3) indicates the type and source of each.

Column (4) represents the number of variables for which the upper and lower bounds were fixed at equal values during the process. And column (5) gives the percentual reduction on interval width over non-fixed variables. All information given in columns (4) and (5) was obtained at the end of node zero.

As may be seen, the penalties were efficient in reducing the size of the solution space. This resulted from the fixing of variables or the updating of their bounds. In fact, in some problems the number of fixed variables was high. Moreover, in all cases with the exception of the last two, the percentual reduction was significant. These exceptions were due to the poor quality of the upper bound solutions produced by the heuristics.

The last column (6) shows the results at the end of the procedure. Not all the computational results obtained through the tests are to be found in the table. We preferred to emphasize the relative behaviour of this algorithm with penalties as compared to a version without penalties.

Let us now stress the results for the algorithm (with penalties):

- the final gap for the optimum was slightly better in most cases;
- the CPU time was lower in almost all problems and, in some cases, very much lower;
- on a number of occasions the algorithm optimally solved the problem before reaching the stipulated maximum number of nodes – the version without penalties did not;
- in rare situations one found that the upper bound was worse.

One should add that, in many other problems, quality and time proved to be the same for both versions.

In short, due to the high number of fixed variables and the significant reduction in variable intervals, the performance of the combined procedure was improved by penalties – not to the extent that one would have expected – but at no computer expense. One may add that its behaviour proved to be better in the practical cases (the algorithm was, in fact, developed with them in mind).

4 – Conclusions

Penalties from linear and lagrangean relaxations of ILPs are easily calculated and computationally cheap. In addition, they can be effective in reducing the size of the problem by fixing variables at particular values and shortening the width of variable intervals.

Literature presents the favourable application of these penalties for binary linear problems and, as has been seen, the above experiment applies such penalties to medium-scale covering problems with integer variables. Therefore, it is felt that similar results would be obtained from other IPLs with bounded variables, whenever the resolution option includes relaxation procedures and the problem dimensions are considerable.

5 – Referencies

[1] Balas, E. and Ho, A. (1980), "Set Covering Algorithms Using Cutting Planes, Heuristics and Subgradient Optimization: A Computational Study", Mathematical Programming 12, 37-60.
[2] Beale, E.M.L. and Small, R.E. (1965), "Mixed Integer Programming by a Branch and Bound Technique", Proceedings of the Third IFIP Congress 1965, vol.2, 450-451.
[3] Beasley, J.E (1987), "An Algorithm for Set Covering Problem", European Journal of Operational Research 31, 85-93.
[4] Chan, T.J, Bean, J.C. and Yano, C.A. (1987), "A Multiplier-Adjustment-Based Branch-and-Bound Algorithm for the Set Partitioning Problem", Technical Report 87-OR-O8, Department of Operational Research and Engineering, Management School of Engineering and Applied Science, Dallas.
[5] Driebeek, N.J. (1966), "An Algorithm for the Solution of Mixed Integer Programming Problems", Management Science 12, 576-587.
[6] Geoffrion, A.M. (1974), "Lagrangean Relaxation for Integer Programming", Mathematical Programming Study 2, 82-114.
[7] Hall, N.G. and Hochbaum, D.S. (1985), "The Multicovering Problem: The Use of Heuristics, Cutting Planes and Subgradient Optimization for a Class of Integer Programs", Working Paper WPS 85-73, College of Administrative Science, The Ohio State University, Columbus.
[8] Hey, A. (1980), "Algorithms for the Set Covering Problem", Ph.D. Thesis, Imperial College, London.
[9] Morris, J.G. and Showalter, M.J. (1983), "Simple Approaches to Shift Days-Off and Tour Scheduling Problems", Management Science 29, 924-950.
[10] Nemhauser, G.L. and Wolsey, L.A. (1988), "Integer and Combinatorial Optimization", John Wiley & Sons, New York.
[11] Paixão, J.P. (1983), "Algorithms for Large Scale Set Covering Problems", Ph.D.Thesis, Imperial College, London.
[12] Paixão, J.P. and Pato, M.V. (1988), "A Structural Lagrangean Relaxation for the Two-Duty Period Bus Driver Scheduling Problem", European Journal of Operational Research 39, 213-222.
[13] Pato, M.V. (1989), "Algoritmos para o Problema de Cobertura Generalizado", Ph.D.Thesis, Faculdade de Ciências, Lisboa.
[14] Sweeney, J.D. and Murphy, R.A.(1981), "Branch and Bound Methods for Multi-Item Scheduling", Operations Research 29, 853-864.
[15] Tomlin, J.A. (1971), "An Improved Branch-and-Bound Method for Integer Programming", Operations Research 19, 1070-1075.