Proceedings of the 9th WSEAS International Conference on Applied Mathematics, Istanbul, Turkey, May 27-29, 2006 (pp296-300)

A Dual Heuristic for Set Partitioning

D. MAGOS
Department of Informatics
Technological Educational Institute of Athens
Ag. Spyridonos, Egaleo 12210
GREECE

Abstract: - Dual heuristics have been successfully used to generate bounds in the context of branch and bound algorithms. In the current work, we present a new dual heuristic for the set partitioning problem. Our approach provides a lower bound at each node of the branch and bound tree by calculating approximate solutions to the *dual* of the linear relaxation of the set partitioning problem. The calculation of the lower bound is carried out incrementally as we optimise over reduced costs instead of the original ones. This feature increases efficiency and reduces memory requirements.

Key-Words: - Integer programming, Set partitioning, Branch and Bound, Dual heuristic

1 Introduction

Several combinatorial optimisation problems are special cases of the Set Partitioning problem (SP)

$$\begin{aligned} &\min z(x) = \sum_{j\in J} c_j x_j, \\ &\sum_{j\in J_i} x_j = 1, \forall\, i \in I, \qquad (1) \\ &x_j \in \{0,1\}, \forall j \in J, \end{aligned}$$

where $J=\{1,...,n\}$, $I=\{1,...,m\}$, and $J_i \subset J$ denotes the set of indices of variables appearing at constraint $i \in I$. Applications of the SP arise in transportation including scheduling of trucks, trains, ships, airline crews, etc. Other applications include stock cutting, political districting [1, 5, 10] etc.

Most of the methods developed for solving SP implement a tree search strategy called *Branch and Bound* (BB) (see [7, 8]). In general terms, the problem is recursively reduced to subproblems each of which is defined by restricting variables to specific values. Conceptually, the initial problem can be considered as the root of a tree while the subproblems constitute the remaining nodes. At each node a lower bound on the value of $z(x)$ is computed. If this value is equal (or greater) to the best solution found so far the current node is labelled *terminal* and the search *backtracks* to another subproblem (node). On the opposite side, if the node is not terminal then additional variables are fixed and new subproblems are generated. The current node is called the *predecessor* of the new nodes and the new nodes are the *successors* of the current node. The search is complete when all the nodes are examined.

The performance of a BB algorithm depends highly on the lower bound procedure used. A dual heuristic, in the role of a lower bound procedure, has been proven most effective in dealing with SP [5]. Other problems to which dual heuristics have been successfully applied are *location* problems [2, 3, 4], *Steiner* problems [9], the *vehicle routing* problem [10], etc.

In the current work, we develop a dual heuristic for SP. The procedure embeds a strategy for evaluating the lower bound at each node based on the lower bound and the reduced costs of the predecessor node. A specialised version of the procedure presented here is used in [6] for the *planar three-index assignment* problem.

2 The dual heuristic

The linear relaxation of SP is obtained by substituting the integrality constraint $x_j \in \{0,1\}$ with the non-negativity constraint $x_j \geq 0$, for all $j \in J$. The dual of the linear relaxation of SP, namely (DSP), is

$$\max \sum_{i \in I} u_i,$$
$$\sum_{i \in I_j} u_i \le c_j, \forall\, j \in J, \quad (2)$$
$$u_i \in \Re, \forall\, i \in I,$$

where the set $I_j \subset I$ indexes the constraints in which the variable x_j appears. The constraints of DSP are called *dual* constraints whereas these of SP *primal*. Under the same convention, x_j, u_i are the primal and dual variables, respectively. The primal variable x_j is associated with the dual constraint j (i.e., the dual constraint having c_j as its right-hand side). A constraint is called *binding* if it is satisfied as equality, i.e. the sum of the values of the variables of the left-hand side is equal to the right-hand side.

Every feasible solution of DSP is a lower bound for SP. The best such bound is the optimum solution of DSP. However, in the context of a BB algorithm, it is not efficient to solve the DSP to optimality at every node of the enumeration tree. Instead, it is preferable to use a heuristic to generate solutions of good quality with minimal computational effort. Such a heuristic is called *dual.* Furthermore, because the solution of the DSP, at a node, is very similar (in terms of the values of the dual variables) to the solution obtained at the predecessor node, it seems appropriate to update the values of the dual variables instead of calculating them from scratch. These observations are exploited in the dual heuristic to be described next.

Let the current node be the successor of node *τ*, and assume that variable x_{j_0} free at node *τ* is now set to zero. The DSP at the current node can be derived from the DSP of node *τ* by increasing to a large value the right-hand side of the dual constraint associated to x_{j_0}. For the constraint j_0 of the DSP, we calculate

$$\bar{c}_{j_0} = c_{j_0} - \sum_{i \in I_{j_0}} u_i. \quad (3)$$

Observe that $\bar{c}_{j_0}$ is the *reduced cost* of the primal variable x_{j_0}. If $\bar{c}_{j_0} > 0$, then an increase in the right-hand side of the dual constraint j_0 (c_{j_0}) will have no effect on the objective function value of DSP. If $\bar{c}_{j_0} = 0$, then the *resource* j_0 has been exhausted and the increase in the value of c_{j_0}, at the current node, can result in an increase in the value of the lower bound. This is achieved only if there is an index $i \in I_{j_0}$ such that u_i belongs solely to non-binding dual constraints.

Therefore, for every variable x_{j_0} free at the predecessor node and restricted to zero at the current node, if $\bar{c}_{j_0} > 0$ then nothing can be done to improve the bound. If $\bar{c}_{j_0} = 0$, we determine for $i \in I_{j_0}$ the maximal increase in the value of the variable u_i. In particular, for all the dual constraints u_i belongs to, we record the difference between the left- and right-hand side. The minimum such difference, namely du, is the amount that u_i can be increased without violating feasibility. Apparently, if u_i belongs to non-binding constraints exclusively, then $du > 0$ and the value of the lower bound is increased by du. We also subtract du from the right-hand sides of all dual constraints u_i belongs to. The procedure is repeated for every $i \in I_{j_0}$.

Finally, we note that we do not have to consider the case $x_{j_0} = 1$ as this is equivalent to setting to zero the remaining variables of any of the primal constraints x_{j0} belongs to.

An initial feasible dual **u** vector at the root of the search tree can be obtained through the following procedure

Step 1

Set $I(u) = I.$
Set $u_i = 0, \forall i \in I.$

Step 2

Let

$$\Delta(u) = \min_j \frac{c_j - \sum \{u_j : j \in I_j\}}{\left| I_j \cap I(u) \right|}, \quad (4)$$

$$j^* = \arg\min_j \frac{c_j - \sum \{u_j : j \in I_j\}}{\left| I_j \cap I(u) \right|} \quad (5)$$

For every $i \in I_{j^*} \cap I(u)$, set

$$u_i = \Delta(u), \quad (6)$$
$$I(u) = I(u) - \{i\}. \quad (7)$$

Step 3

If $I(u) = \emptyset$ stop else go to Step 2.

At each node of the search tree, we record the values of the reduced costs instead of the dual variables. This strategy is based on the observation that the lower bound of each successor of τ is equal to the sum of the lower bound τ and the objective function value of a modified DSP, namely RCDSP, where the right hand side coefficients c_j are replaced by $\bar{c}_j$. The validity of this approach is shown in the following Lemma.

Lemma 1: Let $\mathbf{u}^r$ be a feasible solution of DSP with objective function value $z(\mathbf{u}^r)$. Then, the sum of $z(\mathbf{u}^r)$ and the value of the objective function of the modified DSP obtained by replacing c_j with $\bar{c}_j$, is a lower bound for SP.

Proof: SP can be written in general format as

$$\{\min \mathbf{cx} : \mathbf{Ax} = \mathbf{e}, \mathbf{x} \in \{0,1\}^n\},$$

where $\mathbf{A}$ is a matrix of zeros and ones and $\mathbf{e}$ is a vector of ones. The DSP is

$$\{\max \mathbf{ue} : \mathbf{uA} \le \bar{\mathbf{c}}, \mathbf{u} \in \Re^m\}.$$

Hence, $z(\mathbf{u}^r) = \mathbf{u}^r\mathbf{e} = \sum_{i \in I} u_i^r$. Let $\bar{\mathbf{c}}$ denote the reduced costs vector. Then, $\bar{\mathbf{c}} = \mathbf{c} - \mathbf{u}^r\mathbf{A}$. We define the *reduced-cost* set partitioning problem **(RCSP):**

$$\begin{aligned}
&\{\min \bar{\mathbf{c}}\mathbf{x} : \mathbf{Ax} = \mathbf{e}, \mathbf{x} \in \{0,1\}^n, \bar{\mathbf{c}} = \mathbf{c} - \mathbf{u}^r\mathbf{A}\} \\
&= z(\mathbf{u}^r) + \{\min \bar{\mathbf{c}}\mathbf{x} : \mathbf{Ax} = \mathbf{e}, \mathbf{x} \in \{0,1\}^n\}. \\
&= \{\min \mathbf{cx} - \mathbf{u}^r\mathbf{e} : \mathbf{Ax} = \mathbf{e}, \mathbf{x} \in \{0,1\}^n\} \\
&= -z(\mathbf{u}^r) + \{\min \mathbf{cx} : \mathbf{Ax} = \mathbf{e}, \mathbf{x} \in \{0,1\}^n\}. \quad (8)
\end{aligned}$$

Hence,

$$\begin{aligned}
&\{\min \mathbf{cx} : \mathbf{Ax} = \mathbf{e}, \mathbf{x} \in \{0,1\}^n\} \\
&= z(\mathbf{u}^r) + \{\min \bar{\mathbf{c}}\mathbf{x} : \mathbf{Ax} = \mathbf{e}, \mathbf{x} \in \{0,1\}^n\}. \quad (9)
\end{aligned}$$

Observe that

$$\begin{aligned}
&\{\min \bar{\mathbf{c}}\mathbf{x} : \mathbf{Ax} = \mathbf{e}, \mathbf{x} \in \{0,1\}^n\} \\
&\ge \{\max \mathbf{ue} : \mathbf{uA} \le \bar{\mathbf{c}}, \mathbf{u} \in \Re^m\} \quad (10)
\end{aligned}$$

From (9), (10) follows that

$$\begin{aligned}
&\{\min \mathbf{cx} : \mathbf{Ax} = \mathbf{e}, \mathbf{x} \in \{0,1\}^n\} \\
&\ge z(\mathbf{u}^r) + \{\max \mathbf{ue} : \mathbf{uA} \le \bar{\mathbf{c}}, \mathbf{u} \in \Re^m\}.
\end{aligned}$$

■

The value of the objective function of RCDSP can be obtained using the dual heuristic described previously. Furthermore, there is no need to record all the reduced costs at the successor of τ but only those which are modified. In this way the reduced costs can be recovered during *backtracking*.

3 Numerical example

Consider the set partitioning problem

$$\begin{aligned}
\min\ & 3x_1 + 7x_2 + 5x_3 + 8x_4 \\
& + 10x_5 + 4x_6 + 6x_7 + 9x_8
\end{aligned}$$

$$\begin{aligned}
x_1 + x_2 + x_3 + x_8 &= 1, \\
x_3 + x_4 + x_5 + x_6 &= 1, \\
x_2 + x_5 + x_6 + x_7 + x_8 &= 1, \\
x_3 + x_7 + x_8 &= 1, \\
x_2 + x_4 + x_6 + x_8 &= 1, \\
x_1, \ldots, x_8 &\in \{0,1\}.
\end{aligned}$$

The corresponding DSP is

$$\max z(u) = u_1 + u_2 + u_3 + u_4 + u_5$$

$$\begin{aligned}
u_1 &\le 3, \\
u_1 + u_3 + u_5 &\le 7, \\
u_1 + u_2 + u_4 &\le 5, \\
u_2 + u_5 &\le 8, \\
u_2 + u_3 &\le 10, \\
u_2 + u_3 + u_5 &\le 4, \\
u_3 + u_5 &\le 6, \\
u_1 + u_3 + u_4 + u_5 &\le 9, \\
u_1, \ldots, u_5 &\in \Re.
\end{aligned}$$

At the root of the enumeration tree, the dual solution obtained by the execution of Steps 1-3 and the corresponding reduced costs are

$$z(u) = 7.66,$$
$$u_1 = u_4 = 1.833, u_2 = u_3 = u_5 = 1.333,$$
$$\bar{c}_1 = 1.167, \bar{c}_2 = 3, \bar{c}_3 = 0, \bar{c}_4 = 5.33,$$
$$\bar{c}_5 = 1.333, \bar{c}_6 = 0, \bar{c}_7 = 2.83, \bar{c}_8 = 2.667.$$

Let us consider the subproblem derived from the initial problem by setting $x_3 = x_5 = 0$. Because $\bar{c}_5 > 0$, no improvement on the value of the lower bound can be achieved due to $x_5 = 0$. However, because $\bar{c}_3 = 0$ a possible improvement is at hand by setting x_3 to zero. Observe that $I_3 = \{1,2,4\}$ and the corresponding RCDSP is

$$\max u_1 + u_2 + u_3 + u_4 + u_5$$

$$\begin{aligned} u_1 &\leq 1.167, \\ u_1 + u_3 + u_5 &\leq 3, \\ u_1 + u_2 + u_4 &\leq \infty, \\ u_2 + u_5 &\leq 5.33, \\ u_2 + u_3 &\leq \infty, \\ u_2 + u_3 + u_5 &\leq 0, \\ u_3 + u_5 &\leq 2.83, \\ u_1 + u_3 + u_4 + u_5 &\leq 2.667, \\ u_1, \ldots, u_5 &\in \Re. \end{aligned}$$

We have set all the right-hand sides of the dual constraints to the corresponding reduced costs except for the constraints corresponding to x_3, x_5. For these constraints the right-hand side is set to a big value, denoted ∞, implying that they can never become binding in the current subproblem or any of its successors.

In order to solve the current RCDSP, we set all dual variables to zero and attempt to increase the values of the variables indexed by I_3. Observe that the variable u_1 belongs to the first, second and sixth constraint. Thus,

$$\begin{aligned} du &= \min\{\bar{c}_1, \bar{c}_2, \bar{c}_6\} \\ &= \min\{1.167, 3, 2.667\} = 1.167. \end{aligned}$$

We subtract $du = 1.167$ from the right-hand side of the constraints u_1 belongs to, obtaining thus the following new system of inequalities.

$$\begin{aligned} u_3 + u_5 &\leq 1.833, \\ u_2 + u_4 &\leq \infty, \\ u_2 + u_5 &\leq 5.33, \\ u_2 + u_3 &\leq \infty, \\ u_2 + u_3 + u_5 &\leq 0, \\ u_3 + u_5 &\leq 2.83, \\ u_3 + u_4 + u_5 &\leq 1.5, \\ u_2, \ldots, u_5 &\in \Re. \end{aligned}$$

The value of the variable u_2 cannot be increased because it belongs to the binding constraint $u_2 + u_3 + u_5 \leq 0$. The value of the variable u_4 can be increased by

$$du = \min\{\infty, 1.5\} = 1.5.$$

The lower bound at the current node is

$$\begin{aligned} &z(u) + u_1 + u_4 \\ &= 7.66 + 1.167 + 1.5 = 10.32. \end{aligned}$$

4 Conclusions

In this paper, we have presented a new dual heuristic for SP. Within the BB framework, this heuristic can be used to compute a lower bound, on the objective function value, at every node of the search tree. The proposed procedure is applied to the reduced costs, evaluated at the predecessor of the current node, yielding thus to a fast incremental computation of the bound. Future work includes the modification of the heuristic to deal with the *set-packing* and the *set-covering* problem. These two problems are derived from SP by replacing the equality sign in (1) by '≤' and '≥', respectively.

References:

[1] E. Balas, M. Padberg, Set partitioning: a survey, *SIAM Review*, 18, 1976, pp. 710-760.

[2] R. Baldacci, E. Hadjiconstantinou, V. Maniezzo, A. Mingozzi, A new method for solving capacitated location problems based on a set partitioning approach, *Computers & Operations Research,* 29, 4, 2002, pp. 365-386.

[3] J. Dias, M. E. Captivo, J. Climaco, Efficient primal-dual heuristic for a dynamic location problem, to appear in *Computers & Operations Research.*

[4] D. Erlenkotter, A dual-based procedure for uncapacitated facility location, *Operations Research,* 26, 1, 1978, pp. 992-1009.

[5] M. L. Fischer, P. Kedia, Optimal solution of set covering/partitioning problems using dual heuristics, *Management Science,* 36, 6, 1990, pp. 674-688.

[6] D. Magos, P. Miliotis, An algorithm for the planar three-index assignment problem, *European Journal of Operational Research,* 77, 1994, pp. 141-153.

[7] R.E. Marsten, F. Shepardson, Exact solution to crew scheduling problems using the set partitioning model: recent succesful applications, *Networks* 11, 1981, pp. 165-177.

[8] G. Nemhauser, L. Wolsey, *Integer and combinatorial optimization,* John Wiley and Sons, 1988.

[9] M. Poggi de Aragao, E. Uchoa, R. Werneck, Dual heuristics on the exact solution of large Steiner problems, *Electronic Notes in Discrete Mathematics,* 7, 2004, pp. 1-4.

[10] H.D. Williams, *Model building in mathematical programming,* John Wiley and Sons, 1985.