

Division of Research
Graduate School of Business Administration
The University of Michigan

March 1985

GENERATING OPTIMAL DUAL SOLUTIONS
WITH SIMPLE PRIMAL HEURISTICS

Working Paper No. 422

Hans M. Schneeberger

FOR DISCUSSION PURPOSES ONLY

None of this material is to be quoted or
reproduced without the expressed permission
of the Division of Research.

Abstract

This paper develops results to solve the Lagrangean dual problem with simple primal heuristics. The analysis is motivated by the fact that current solution procedures for the dual problem can become computationally burdensome. The results are used on the "nearest unvisited city heuristic" to solve the travelling salesman dual problem and on the "bang for a buck ratio" to solve the knapsack dual problem.

1. Introduction

This paper investigates procedures to generate optimal dual solutions with simple primal heuristics. Research to date indicates that it is unlikely that NP-hard optimization problems can be solved without explicitly or implicitly enumerating all possible feasible solutions [2]. At the same time any implicit enumeration such as branch and bound or dynamic programming with branch and bound requires the use of upper and lower bounds to the optimal solution. The performance of any algorithm using upper and lower bounds depends almost exclusively on the bounding procedures used. The important criteria for these bounding procedures is the computational effort used to generate the bounds and the tightness of the bounds generated.

In order to generate upper bounds one can apply a heuristic to the original optimization problem to get a feasible solution. The objective function value of such a solution can then serve as an upper bound. Successful upper bounding strategies have been such that upper bounds are computed only a very few times in the course of an optimization. Hence the computational effort to generate an upper bound has not been critical to the performance of implicit enumeration algorithms.

Lower bounds, on the other hand, are often computed at every node in a branch and bound tree or at every state in a dynamic program. The computational efficiency of lower bounding procedures therefore becomes a major factor for the performance of implicit enumeration algorithms. The fine tuning of such enumeration algorithms often involves tradeoffs between the time spent to compute lower bounds and work with a smaller problem and the time spent to work with larger problems in the main routine [4].

Lower bounds which have been used successfully are the objective function value of an LP-relaxation of the original problem and the objective function value of the Lagrangean dual to the original problem. It is well known that the solution to the dual problem provides a lower bound which is at least as tight as the objective function value of the LP relaxation [3]. However, solving the dual problem can involve considerably more computational effort than solving the LP-relaxed problem. This is often the case if one needs to resort to a subgradient optimization to obtain the dual solution, as noted by Fisher [1]. Fisher [1] concludes that the development of multiplier adjustment methods to circumvent the subgradient optimization holds great promise for future research.

In order to avoid the computational burden of solving the dual problem without abandoning the tight bounds provided by a dual solution, recent work has been focused on procedures which approximate the optimal solution to the dual problem [5, 6]. In general such approximation procedures exploit well known properties of a given solution by adjusting the Lagrangean multipliers such that the solution under consideration is optimal for the computed multipliers. It should also be noted that such multiplier adjustment methods provide an update of the upper bound each time a lower bound is computed.

Potts and van Wassenhoven [5] reported on a successful application of such a multiplier adjustment method in a branch and bound algorithm developed to solve the single machine scheduling problem to minimize weighted completion time. Schneeberger [6] showed that the multiplier adjustment method used does not just approximate the dual solution but actually provides the optimal solution to the dual problem.

This paper is an extension of [6] and develops results and procedures to solve the dual of NP-hard problems, using multiplier adjustment methods. We illustrate the use of these results on two well-known problems, the travelling salesman problem and the knapsack problem.

In the next section we develop results, characterizing optimal solutions to the integer programming dual problem. In section three we illustrate the use of the results established in the previous section, by developing an optimal multiplier adjustment method for the travelling salesman problem. In section four we illustrate the use of the same results, by developing an optimal multiplier adjustment method for the knapsack salesman problem. The paper concludes in section five.

2. Sufficient saddlepoint conditions

In this section we develop conditions sufficient to verify a saddlepoint, that is, sufficient conditions to verify an optimal dual solution. Consider the following optimization problem where x is a vector of integers.

$$\begin{aligned} \min_x & f(x) \\ \text{s.t. } & \underline{g(x)} < 0. \end{aligned}$$

The Lagrangean dual to this problem can be stated as:

$$\max_u \min_x \{f(x) + ug(x)\}.$$

Let $h(u)$ be the objective function value of the Lagrangean dual evaluated at u , that is:

$$h(u) = \min_x \{f(x) + ug(x)\}.$$

Note that x which solves $h(u)$ may not be unique, let $X(u)$ be the set of all X which solve $h(u)$.

Theorem 1: Let $h(\bar{u}) = \max h(u)$. Then there exists a $\bar{x} \in X(\bar{u})$ such that $g(\bar{x}) \leq 0$.

Proof: Consider the theorem is not true, that is $h(\bar{u}) = \max h(u)$ and there does not exist a $\bar{x} \in X(\bar{u})$ such that $g(\bar{x}) \leq 0$. This implies that for all $x \in X(\bar{u})$ there exists an Δ such that $\bar{u}g(x) < (\bar{u} + \Delta)g(x)$ which implies $h(\bar{u}) < \max h(u)$. This contradicts the assumption and hence completes the proof.

Theorem 1 leads immediately to the following Lemma:

Lemma: The following two conditions are sufficient to verify a saddle-point at x^0, u^0 .

i) $g(x^0) \leq 0$

ii) for any $\Delta \geq 0$, $\bar{x} \in X(u^0 - \Delta) \rightarrow g(\bar{x}) \not\leq 0$.

Proof: Since $g(x^0) \leq 0$, $h(u^0 + \Delta) \leq h(u^0)$. In addition, from Theorem 1 we know that the second condition implies $h(u^0 - \Delta) \leq h(u^0)$, which completes the proof.

The preceding observations facilitate the development of optimal multiplier adjustment methods as illustrated in the next two sections. We will generate a feasible solution $x^0 \in X(u^0)$, that is, a solution which belongs to the set of solutions solving $h(u^0)$. Once we have found such a solution we compute $f(x^0)$, $g(x^0)$ and find u^0 such that $f(x^0) + u^0 g(x^0) = \max \{f(x^0) + u g(x^0)\}$.

3. Travelling salesman problem

In this section we show that the nearest unvisited city heuristic can be used to compute the optimal solution to the travelling salesman dual problem. Let $i=1,2,\dots,N$ be the cities to be visited and let C be the $N \times N$ distance

matrix with element c_{ij} denoting the distance between city i and city j . The travelling salesman problem can then be formulated as follows:

$$\min \sum_{i=1}^N \sum_{j=1}^N c_{ij} x_{ij} \quad (1)$$

$$\text{s.t.} \quad \sum_{i=1}^N x_{ij} = 1 \quad (2)$$

$$\sum_{j=1}^N x_{ij} = 1 \quad (3)$$

Note that x_{ij} is a zero/one vector, indicating which links are to be selected in a tour. For the remaining part of our discussion we relax constraint (3) such that the sum over all j 's in x_{ij} is less than or equal to one, rather than equal to one. This relaxation does not effect the optimal solution or any of the solution procedures. However, it does simplify our presentation since the dual variables of this constraint can now only take on values greater than or equal to zero.

The Lagrangean dual problem to [(1), (2), (3)] can then be stated as follows:

$$\begin{aligned} \max_u \{ \min_x \sum_{i=1}^N [\sum_{j=1}^N (c_{ij} + u_i) x_{ij} - u_i] \\ \text{s.t. (2)} \end{aligned} \quad (4)$$

For any given vector of multipliers u problem (4) can be stated as:

$$\begin{aligned} h(u) = \min_x \sum_{i=1}^N [\sum_{j=1}^N (c_{ij} + u_i) x_{ij} - u_i] \\ \text{s.t. (2)} \end{aligned} \quad (5)$$

The objective function value $h(u)$ can easily be obtained by scanning all the columns $j=1,2,\dots,N$, and setting the x_{ij} 's which correspond to the lowest $(c_{ij} + u_i)$ -value in a given column equal to one and all other x_{ij} 's in the same column equal to zero, and then computing

$$\sum_{i=1}^N [\sum_{j=1}^N (c_{ij} + u_i) x_{ij} - u_i].$$

Consider the following multiplier adjustment scheme. Let k_j be the difference between the lowest and the second lowest c_{ij} value in column j . Start the nearest unvisited city heuristic with the column which corresponds to the largest k_j -value. Next consider the column with the next largest k_j -value, and so on. That is, set x_{ij} which corresponds to the lowest c_{ij} value in the column with the largest k_j -value equal to one and all other x_{ij} 's in the same column equal to zero. Next set x_{ij} which corresponds to the lowest c_{ij} value in the column with the second largest k_j -value equal to one and all other x_{ij} 's in the same columns equal to zero, and so on.

If at any stage the nearest unvisited city does not correspond to the nearest city (cities) we increase the necessary u_i -values until $(c_{ij} + u_i)$, corresponding to the nearest unvisited city, is equal to $(c_{ij} + u_i)$, corresponding to the nearest city (cities). We then recompute k_j for the whole column, using the $(c_{ij} + u_i)$ -values rather than the c_{ij} -values, and start the nearest unvisited city heuristic again with the column corresponding to the largest k_j -value. Note that the particular sequence in which the columns are chosen for evaluating the nearest city guarantees starting at most N times. Let u denote the multipliers obtained by this scheme.

Theorem 2: \bar{u} solves $\max h(u)$, that is, \bar{u} solves the dual to the travelling salesman problem [(1), (2), (3)].

Proof: First note that $\sum_{j=1}^N x_{ij} \leq 1$ for all $i=1,2,\dots,N$, and hence condition 1 of theorem 1 is satisfied. Further note that any decrease in any of the multipliers will cause $\sum_{j=1}^N x_{ij} > 1$ and hence condition two of Theorem one is satisfied as well, which completes the proof.

4. Knapsack Problem

In this section we show that a simple heuristic to the knapsack problem can be used to compute the optimal solution to the knapsack dual problem. Let $i=1,2,\dots,N$ be the elements to be considered for loading into the knapsack and let a_i and p_i denote the return and the size, respectively, for element i . In addition let the elements be labeled according to decreasing a/p -ratios and let C be the capacity of the knapsack. The problem can then be stated as follows:

$$\max_S \sum_{i \in S} a_i \quad (6)$$

$$\text{s.t. } \sum_{i \in S} p_i \leq C \quad (7)$$

S is the subset of all N elements which are loaded into the knapsack. The Lagrangean dual problem to [(6), (7)] can be stated as follows:

$$\min_u \left\{ \max_S \sum_{i \in S} (a_i - up_i) + uC \right\} \quad (8)$$

For any given multiplier, problem (8) can be stated as follows:

$$h(u) = \max \left[\sum_{i \in S} (a_i - up_i) + uC \right] \quad (9)$$

The objective function value $h(u)$ can be obtained simply by defining

$$S = \{i \mid (a_i - up_i) > 0\}, \text{ and computing } \sum_{i \in S} (a_i - up_i).$$

Let job K be such that

$$\begin{aligned} \text{i)} \quad & \sum_{i=1}^{k-1} p_i \leq C \\ \text{ii)} \quad & \sum_{i=1}^k p_i > C \end{aligned}$$

Theorem 3: $u = a_k/p_k$ solves $\min h(u)$, that is, u solves the dual to the knapsack problem [(6),(7)].

Proof: The proof can easily be established using Theorem 1.

It can now easily be established that for any knapsack problem the bound obtained using a Lagrangean relaxation is always equal to the bound obtained using a LP-relaxation.

Lemma: The objective function value of the knapsack dual problem is always equal to the objective function value of an LP-relaxation to the knapsack problem.

Proof: Consider the dual problem $h(\bar{u})$ at \bar{u} . The objective function value at this point equals:

$$\sum_{i=1}^{k-1} a_i + \bar{u} \left(C - \sum_{i=1}^{k-1} p_i \right) \quad (10)$$

Let $(C - \sum_{i=1}^{k-1} p_i) = \Delta p$. Note that Δp indicates the amount of "space" left in the knapsack. Further $\bar{u} = a_k/p_k$. Hence equation (10) can be rewritten as

$$\sum_{i=1}^{k-1} a_i + a_k (\Delta p/p_k).$$

By definition of k , $0 \leq \Delta p/p_k < 1$. It is easy to see that applying an LP-relaxation to the knapsack problem will yield a solution such that all the elements in the knapsack have a/p -ratios higher than the elements not loaded into the knapsack. In addition the element which is only partially loaded has an a/p ratio which is lower than the ratio of any other element in the knapsack, but higher than the a/p -ratio of any elements not loaded into the knapsack.

5. Conclusion

In this paper we developed results to solve the dual of NP-hard optimization problems with relatively simple procedures. The research was motivated by the fact that current methods to solve the dual problems, e.g. subgradient optimization, can become computationally burdensome. The results are applied to two problems, the travelling salesman problem and the knapsack problem.

A further example can be found in [6]. It appears that the ideas developed in this paper can be used to solve any dual problem, which would imply that the computationally burdensome subgradient methods can be avoided for the most part.

REFERENCES

1. Fisher, M.L., "The Lagrangean Relaxation Methods for Solving Integer Programming Problems," Management Science, Vol. 27, No. 1 (1981), pp. 1-18.
2. Garey, M.R. and D.S. Johnson, "Computers and Intractability," San Francisco, CA: W.H. Freeman and Company, 1979.
3. Lasdon, L.S., "Optimization Theory for Large Systems," New York, NY: MacMillan Publishing Co., 1970.
4. Morin, T.L. and R.E. Marsten, "Branch and Bound Strategies for Dynamic Programming," Operations Research, Vol. 24, No. 4 (1976), pp. 611-27.
5. Potts, C.N. and L.N. van Wassenhoven, "An Algorithm for Single Machine Sequencing with Deadlines to Minimize Total Weighted Completion Time," European Journal of Operations Research, No. 12 (1983), pp. 379-87.
6. Schneeberger, H.M., "On Potts' and van Wassenhoven's Algorithm for Single Machine Sequencing with Deadlines to Minimize Total Weighted Completion Time," Working Paper, The University of Michigan, Graduate School of Business Administration, 1985.