MANAGEMENT SCIENCE
Vol. 27, No. 1, January 1981
Printed in U.S.A.

THE LAGRANGIAN RELAXATION METHOD FOR SOLVING INTEGER PROGRAMMING PROBLEMS*

MARSHALL L. FISHER†

One of the most computationally useful ideas of the 1970s is the observation that many hard integer programming problems can be viewed as easy problems complicated by a relatively small set of side constraints. Dualizing the side constraints produces a Lagrangian problem that is easy to solve and whose optimal value is a lower bound (for minimization problems) on the optimal value of the original problem. The Lagrangian problem can thus be used in place of a linear programming relaxation to provide bounds in a branch and bound algorithm. This approach has led to dramatically improved algorithms for a number of important problems in the areas of routing, location, scheduling, assignment and set covering. This paper is a review of Lagrangian relaxation based on what has been learned in the last decade.
(PROGRAMMING—INTEGER ALGORITHMS; PROGRAMMING—INTEGER ALGORITHM BRANCH AND BOUND; PROGRAMMING—INTEGER ALGORITHMS, HEURISTIC)

1. Introduction

It is well known that combinatorial optimization problems come in two varieties. There is a small number of "easy" problems which can be solved in time bounded by a polynomial in the input length and an all-too-large class of "hard" problems for which all known algorithms require exponential time in the worst-case. Among the hard problems, there are "easier hard" problems, like the knapsack problem, that have pseudo-polynomial algorithms that run in polynomial time if certain numbers in the problem data are bounded.

One of the most computationally useful ideas of the 1970s is the observation that many hard problems can be viewed as easy problems complicated by a relatively small set of side constraints. Dualizing the side constraints produces a Lagrangian problem that is easy to solve and whose optimal value is a lower bound (for minimization problems) on the optimal value of the original problem. The Lagrangian problem can thus be used in place of a linear programming relaxation to provide bounds in a branch and bound algorithm. As we shall see, the Lagrangian approach offers a number of important advantages over linear programming.

There were a number of forays prior to 1970 into the use of Lagrangian methods in discrete optimization, including the Lorie-Savage [31] approach to capital budgeting, Everett's proposal for "generalizing" Lagrange multipliers [14] and the philosophically-related device of generating columns by solving an easy combinatorial optimization problem when pricing out in the simplex method [24]. However, the "birth" of the Lagrangian approach as it exists today occurred in 1970 when Held and Karp [27], [28] used a Lagrangian problem based on minimum spanning trees to devise a dramatically successful algorithm for the traveling salesman problem. Motivated by Held and Karp's success Lagrangian methods were applied in the early 1970s to

*Accepted by Donald Erlenkotter, Special Editor; received June 13, 1979. This paper has been with the author 5 months for 1 revision.
†University of Pennsylvania.

0025-1909/81/2701/0001$01.25

scheduling problems (Fisher [15]) and the general integer programming problem (Shapiro [42], Fisher and Shapiro [16]). Lagrangian methods had gained considerable currency by 1974 when Geoffrion [22] coined the perfect name for this approach—"Lagrangian relaxation." Since then the list of applications of Lagrangian relaxation has grown to include over a dozen of the most infamous combinatorial optimization problems. For most of these problems, Lagrangian relaxation has provided the best existing algorithm for the problem and has enabled the solution of problems of practical size.

This paper is a review of Lagrangian relaxation based on what has been learned in the last decade. The reader is referred to Shapiro [43] for another recent survey of Lagrangian relaxation from a somewhat different perspective. The recent book by Shapiro [44] marks the first appearance of the term Lagrangian relaxation in a textbook.

2. Basic Constructions

We begin with a combinatorial optimization problem formulated as the integer program

$$\begin{aligned} Z = \min\ & cx \\ \text{s.t. } & Ax = b, \\ & Dx \leqslant e \\ & x \geqslant 0 \text{ and integral,} \end{aligned} \tag{P}$$

where x is $n \times 1$, b is $m \times 1$, e is $k \times 1$ and all other matrices have conformable dimensions. Let (LP) denote problem (P) with the integrality constraint on x relaxed, and let Z_{LP} denote the optimal value of (LP).

We assume that the constraints of (P) have been partitioned into the two sets $Ax = b$ and $Dx \leqslant e$ so as to make it easy to solve the Lagrangian problem

$$\begin{aligned} Z_D(u) = \min\ & cx + u(Ax - b), \\ & Dx \leqslant e, \\ & x \geqslant 0 \text{ and integral,} \end{aligned} \tag{LR_u}$$

where $u = (u_1, \ldots, u_m)$ is a vector of Lagrange multipliers. By "easy to solve" we of course mean easy relative to (P). For all applications of which I am aware, the Lagrangian problem has been solvable in polynomial or pseudo-polynomial time.

For convenience we assume that (P) is feasible and that the set $X = \{x \mid Dx \leqslant e, x \geqslant 0 \text{ and integral}\}$ of feasible solutions to (LR_u) is finite. Then $Z_D(u)$ is finite for all u. It is straightforward to extend the development when these assumptions are violated or when inequality constraints are included in the set to be dualized.

It is well known that $Z_D(u) \leqslant Z$. This is easy to show by assuming an optimal solution x^* to (P) and observing that

$$Z_D(u) \leqslant cx^* + u(Ax^* - b) = Z.$$

The inequality in this relation follows from the definition of $Z_D(u)$ and the equality from $Z = cx^*$ and $Ax^* - b = 0$. If $Ax = b$ is replaced by $Ax \leqslant b$ in (P), then we require $u \geqslant 0$ and the argument becomes

$$Z_D(u) \leqslant cx^* + u(Ax^* - b) \leqslant Z$$

where the second inequality follows from $Z = cx^*$, $u \geqslant 0$ and $Ax^* - b \leqslant 0$. Similarly, for $Ax \geqslant b$ we require $u \leqslant 0$ for $Z_D(u) \leqslant Z$ to hold.

We will discuss in a later section methods for determining u. In general, it is not possible to guarantee finding u for which $Z_D(u) = Z$, but this frequently happens for particular problem instances.

The fact that $Z_D(u) \leqslant Z$ allows (LR_u) to be used in place of (LP) to provide lower bounds in a branch and bound algorithm for (P). While this is the most obvious use of (LR_u), it has a number of other uses. It can be a medium for selecting branching variables and choosing the next branch to explore. Good feasible solutions to (P) can frequently be obtained by perturbing nearly feasible solutions to (LR_u). Finally, Lagrangian relaxation has been used recently [10], [20] as an analytic tool for establishing worst-case bounds on the performance of certain heuristics.

3. Example

The generalized assignment problem is an excellent example for illustrating Lagrangian relaxation because it is rich with readily apparent structure. The generalized assignment problem (GAP) is the integer program

$$Z = \min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{1}$$

$$\sum_{i=1}^{m} x_{ij} = 1, \qquad j = 1, \ldots, n, \tag{2}$$

$$\sum_{j=1}^{n} a_{ij} x_{ij} \leqslant b_i, \qquad i = 1, \ldots, m, \tag{3}$$

$$x_{ij} = 0 \text{ or } 1, \quad \text{all } i \text{ and } j. \tag{4}$$

There are two natural Lagrangian relaxations for the generalized assignment problem. The first is obtained by dualizing constraints (2).

$$Z_{D1}(u) = \min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{j=1}^{n} u_j \left(\sum_{i=1}^{m} x_{ij} - 1 \right)$$

subject to (3) and (4)

$$= \min \sum_{i=1}^{m} \sum_{j=1}^{n} (c_{ij} + u_j) x_{ij} - \sum_{j=1}^{n} u_j$$

subject to (3) and (4). ($LR1_u$)

This problem reduces to m 0-1 knapsack problems and can thus be solved in time proportional to $n \sum_{i=1}^{m} b_i$.

The second relaxation is obtained by dualizing constraints (3) with $v \geqslant 0$.

$$Z_{D2}(v) = \min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{i=1}^{m} v_i \left(\sum_{j=1}^{n} a_{ij} x_{ij} - b_i \right)$$

subject to (2) and (4)

$$= \min \sum_{j=1}^{n} \left(\sum_{i=1}^{m} (c_{ij} + v_i a_{ij}) x_{ij} \right) - \sum_{i=1}^{m} v_i b_i$$

subject to (2) and (4). ($LR2_v$)

This relaxation is defined for $v \geqslant 0$, which is a necessary condition for $Z_{D2}(v) \leqslant Z$ to hold. Since constriants (2) are generalized upper bound (GUB) constraints, we will call a problem like ($LR2_v$) a 0–1 GUB problem. Such a problem is easily solved in time proportional to nm by determining $\min_i(c_{ij} + v_i a_{ij})$ for each j and setting the associated $x_{ij} = 1$. Remaining x_{ij} are set to zero.

4. Issues

A little thought about using ($LR1_u$) or ($LR2_v$) within a branch and bound algorithm for the generalized assignment problem quickly brings to mind a number of issues that need to be resolved. Foremost among these is:

(1) How will we select an appropriate value for u?

A closely related question is:

(2) Can we find a value for u for which $Z_D(u)$ is equal to or nearly equal to Z?

The generalized assignment problem also shows that different Lagrangian relaxations can be devised for the same problem. Comparing ($LR1_u$) and ($LR2_v$), we see that the first is harder to solve but might provide better bounds. There is also the question of how either of these relaxations compares with the LP relaxation. This leads us to ask

(3) How can we choose between competing relaxations, i.e., different Lagrangian relaxations and the linear programming relaxation?

Lagrangian relaxations also can be used to provide good feasible solutions. For example, a solution to ($LR2_v$) will be feasible in the generalized assignment problem unless the "weight" of items assigned to one or more of the "knapsacks" corresponding to constraints (3) exceeds the capacity b_i. If this happens, we could reassign items from overloaded knapsacks to other knapsacks, perhaps using a variant of a bin-packing heuristic, to attempt to achieve primal feasibility. In general we would like to know

(4) How can (LR_u) be used to obtain feasible solutions for (P)?

How good are these solutions likely to be?

Finally, we note that the ultimate use of Lagrangian relaxation is for fathoming in a branch and bound algorithm, which leads us to ask:

(5) How can the lower and upper bounding capabilities of the Lagrangian problem be integrated within branch and bound?

The remainder of this paper is organized around these five issues, which arise in any application of Lagrangian relaxation. A separate section is devoted to each one. In some cases (issues (1) and (3)) general theoretical results are available. But more often, the "answers" to the questions we have posed must be extrapolated from computational experience or theoretical results that have been obtained for specific applications.

5. Existing Applications

Table 1 is a compilation of the applications of Lagrangian relaxation of which I am aware. I have not attempted to include algorithms, like those given in [4] and [7], that are described without reference to Lagrangian relaxation, but can be described in terms of Lagrangian relaxation with sufficient insight. Nor have I included references describing applications of the algorithms in Table 1. For example, reference [37] describes a successful application of the Lagrangian relaxation in [10] to a specialized uncapacitated location problem involving data clustering. Finally, the breadth and

developing nature of this field makes it certain that other omissions exist. I would be happy to learn of any applications that I have overlooked.

This list speaks for itself in terms of the range of hard problems that have been addressed and the types of embedded structures that have been exploited in Lagrangian problems. Most of these structures are well-known but two require comment. The pseudo-polynomial dynamic programming problems arising in scheduling are similar to the 0–1 knapsack problem if we regard the scheduling horizon as the knapsack size and the set of jobs to be scheduled as the set of items available for packing. The notation VUB stands for "variable upper bound" [41] and denotes a problem structure in which some variables are upper bounded by other 0–1 variables. An example of this structure is given in §7.

TABLE 1

Applications of Lagrangian Relaxation

Problem	Researchers	Lagrangian Problem
TRAVELING SALESMAN		
Symmetric	Held & Karp [27], [28]	Spanning Tree
	Helbig Hansen and Krarup [26]	Spanning Tree
Asymmetric	Bazarra & Goode [3]	Spanning Tree
Symmetric	Balas & Christofides [1]	Perfect 2-Matching
Asymmetric	Balas & Christofides [1]	Assignment
SCHEDULING		
$n \mid m$ Weighted Tardiness	Fisher [15]	Pseudo-Polynomial Dynamic Programming
1 Machine Weighted Tardiness	Fisher [18]	Pseudo-Polynomial DP
Power Generation Systems	Muckstadt & Koenig [36]	Pseudo-Polynomial DP
GENERAL IP		
Unbounded Variables	Fisher & Shapiro [16]	Group Problem
Unbounded Variables	Burdet & Johnson [5]	Group Problem
0–1 Variables	Etcheberry et. al. [13]	0–1 GUB
LOCATION		
Uncapacitated	Cornuejols, Fisher, & Nemhauser [10]	0–1 VUB
	Erlenkotter [11]	0–1 VUB
Capacitated	Geoffrion & McBride [23]	0–1 VUB
Databases in Computer Networks	Fisher & Hochbaum [19]	0–1 VUB
GENERALIZED ASSIGNMENT		
	Ross & Soland [40]	Knapsack
	Chalmet & Gelders [8]	Knapsack, 0–1 GUB
	Fisher, Jaikumar & Van Wassenhove [21]	Knapsack
SET COVERING—PARTITIONING		
Covering	Etcheberry [12]	0–1 GUB
Partitioning	Nemhauser & Weber [38]	Matching

6. Determining u

It is clear that the best choice for u would be an optimal solution to the dual problem

$$Z_D = \max_u Z_D(u) \tag{D}$$

Most schemes for determining u have as their objective finding optimal or near optimal solutions to (D).

Problem (D) has a number of important structural properties that make it feasible to solve. We have assumed that the set $X = \{x \mid Dx \leqslant e,\ x \geqslant 0 \text{ and integral}\}$ of feasible solutions for (LR_u) is finite, so we can represent X as $X = \{x^t,\ t = 1, \ldots, T\}$. This allows us to express (D) as the following linear program with many constraints.

$$\begin{aligned} &Z_D = \max w, \\ &w \leqslant cx^t + u(Ax^t - b), \qquad t = 1, \ldots, T. \end{aligned} \tag{$\overline{\mathrm{D}}$}$$

The LP dual of $(\overline{\mathrm{D}})$ is a linear program with many columns.

$$\begin{aligned} Z_D = \min \quad & \sum_{t=1}^{T} \lambda_t cx^t, \\ & \sum_{t=1}^{T} \lambda_t Ax^t = b, \\ & \sum_{t=1}^{T} \lambda_t = 1, \\ & \lambda_t \geqslant 0, \qquad t = 1, \ldots, T. \end{aligned} \tag{$\overline{\mathrm{P}}$}$$

Problem $(\overline{\mathrm{P}})$ with λ_t required to be integral is equivalent to (P), although $(\overline{\mathrm{P}})$ and (LP) generally are not equivalent problems.

Both $(\overline{\mathrm{D}})$ and $(\overline{\mathrm{P}})$ have been important constructs in the formulation of algorithms for (D). Problem $(\overline{\mathrm{D}})$ makes it apparent that $Z_D(u)$ is the lower envelope of a finite family of linear functions. The form of $Z_D(u)$ is shown in Figure 1 for $m = 1$ and $T = 4$. The function $Z_D(u)$ has all the nice properties, like continuity and concavity, that make life easy for a hill-climbing algorithm, except one—differentiability. The function is nondifferentiable at any $\bar{u}$ where $(\mathrm{LR}_{\bar{u}})$ has multiple optima. Although it is differentiable almost everywhere, it generally is nondifferentiable at an optimal point.

An m-vector y is called a subgradient of $Z_D(u)$ at $\bar{u}$ if it satisfies

$$Z_D(u) \leqslant Z_D(\bar{u}) + y(u - \bar{u}), \quad \text{for all } u.$$

It's apparent that $Z_D(u)$ is subdifferentiable everywhere. The vector $(Ax^t - b)$ is a subgradient at any u for which x^t solves (LR_u). Any other subgradient is a convex combination of these primitive subgradients. With this perspective, the well-known result that u^* and λ^* are optimal for $(\overline{\mathrm{D}})$ and $(\overline{\mathrm{P}})$ if and only if they are feasible and

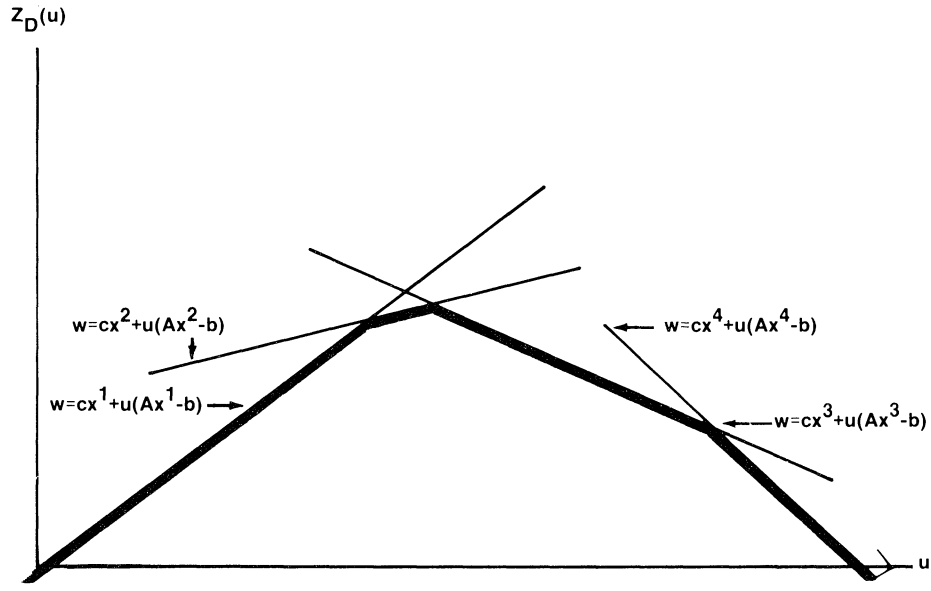

FIGURE 1. The Form of $Z_D(u)$.

satisfy a complementary slackness condition can be seen to be equivalent to the obvious fact that u^* is optimal in (D) if and only if 0 is a subgradient of $Z_D(u)$ at u^*.

Stimulated in large part by applications in Lagrangian relaxation, the field of nondifferentiable optimization using subgradients has recently become an important topic of study in its own right with a large and growing literature. Our review of algorithms for (D) will be brief and limited to the following three approaches that have been popular in Lagrangian relaxation applications: (1) the subgradient method, (2) various versions of the simplex method implemented using column generation techniques, and (3) multiplier adjustment methods. References [17] and [29] contain general discussions on the solution of (D) within the context of Lagrangian relaxation. Reference [2] is a good general source on nondifferentiable optimization.

The subgradient method is a brazen adaptation of the gradient method in which gradients are replaced by subgradients. Given an initial value u^0 a sequence $\{u^k\}$ is generated by the rule

$$u^{k+1} = u^k + t_k(Ax^k - b)$$

where x^k is an optimal solution to (LR_{u^k}) and t_k is a positive scalar step size. Because the subgradient method is easy to program and has worked well on many practical problems, it has become the most popular method for (D). There have also been many papers, such as Camerini et al. [6], that suggest improvements to the basic subgradient method.

Computational performance and theoretical convergence properties of the subgradient method are discussed in Held, Wolfe and Crowder [29] and their references, and in several references on nondifferentiable optimization, particularly Goffin [25]. The fundamental theoretical result is that $Z_D(u^k) \to Z_D$ if $t_k \to 0$ and $\sum_{i=0}^{k} t_i \to \infty$. The

step size used most commonly in practice is

$$t_k = \frac{\lambda_k\left(Z^* - Z_D(u^k)\right)}{\|Ax^k - b\|^2}$$

where λ_k is a scalar satisfying $0 < \lambda_k \leqslant 2$ and Z^* is an upper bound on Z_D, frequently obtained by applying a heuristic to (P). Justification of this formula is given in [29]. Often the sequence λ_k is determined by setting $\lambda_0 = 2$ and halving λ_k whenever $Z_D(u)$ has failed to increase in some fixed number of iterations. This rule has performed well empirically, even though it is not guaranteed to satisfy the sufficient condition given above for optimal convergence.

Unless we obtain a u^k for which $Z_D(u^k)$ equals the cost of a known feasible solution, there is no way of proving optimality in the subgradient method. To resolve this difficulty, the method is usually terminated upon reaching an arbitrary iteration limit.

Usually $u^0 = 0$ is the most natural choice but in some cases one can do better. The generalized assignment problem is a good example. Assuming $c_{ij} > 0$ for all ij, the solution $x = 0$ is optimal in $(\mathrm{LR}1_u)$ for any u satisfying $u_j \leqslant c_{ij}$ for all i and j. Setting $u_j^0 = \min_i c_{ij}$ is thus a natural choice. It is clearly better than $u^0 = 0$ and, in fact, maximizes the lower bound over all u for which $x = 0$ is optimal in $(\mathrm{LR}1_u)$.

Another class of algorithms for (D) is based on applying a variant of the simplex method to (P), and generating an appropriate entering variable on each iteration by solving $(\mathrm{LR}_{\bar{u}})$, where $\bar{u}$ is the current value of the simplex multipliers. Of course, using the primal simplex method with column generation is an approach with a long history [24]. However, this approach is known to converge very slowly and does not produce monotonically increasing lower bounds. These deficiencies have prompted researchers to devise column generation implementations of dual forms of the simplex method, specifically the dual simplex method (Fisher [15]) and the primal-dual simplex method (Fisher, Northup, Shapiro [17]). The primal-dual simplex method can also be modified slightly to make it the method of steepest ascent for (D). Hogan, Marsten and Blankenship [30] and Marsten [33] have had success with an interesting modification of these simplex approaches that they call BOXSTEP. Beginning at given u^0, a sequence $\{u^k\}$ is generated. To obtain u^{k+1} from u^k, we first solve $(\bar{\mathrm{D}})$ with the additional requirement that $|u_i - u_i^k| \leqslant \delta$ for some fixed positive δ. Let $\bar{u}^k$ denote the optimal solution to this problem. If $|\bar{u}_i^k - u_i^k| < \delta$ for all i then $\bar{u}^k$ is optimal in (D). Otherwise set $u^{k+1} = u^k + t_k(\bar{u}^k - u^k)$ where t_k is the scalar that solves

$$\max_t Z_D\left(u^k + t(\bar{u}^k - u^k)\right).$$

This line search problem is easily solved by Fibonacci methods.

Generally, the simplex-based methods are harder to program and have not performed quite so well computationally as the subgradient method. They should not be counted out, however. Further research could produce attractive variants. We note also that the dual, primal-dual and BOXSTEP methods can all be used in tandem with the subgradient method by initiating them with a point determined by the subgradient method. Using them in this fashion to finish off a dual optimization probably best exploits their comparative advantages.

The third approach, multiplier adjustment methods, are specialized algorithms for (D) that exploit the structure of a particular application. In these methods, a sequence u^k is generated by the rule $u^{k+1} = u^k + t_k d_k$ where t_k is a positive scalar and d_k is a direction. To determine d_k we define a finite and usually small set of primitive directions S for which it is easy to evaluate the directional derivative of $Z_D(u)$. Usually directions in S involve changes in only one or two multipliers. For directions in S, it should be easy to determine the directional derivative of $Z_D(u)$. Directions in S are scanned in fixed order and d_k is taken to be either the first direction found along which $Z_D(u)$ increases or the direction of steepest ascent within S. The step size t_k can be chosen either to maximize $Z_D(u^k + td_k)$ or to take us to the first point at which the directional derivative changes. If S contains no improving direction we terminate, which, of course, can happen prior to finding an optimal solution to (D).

Successful implementation of primitive-direction ascent for a particular problem requires an artful specification of the set S. S should be manageably small, but still include directions that allow ascent to at least a near optimal solution. Held and Karp [27] experimented with primitive-direction ascent in their early work on the traveling salesman problem. They had limited success using a set S consisting of all positive and negative coordinate vectors. This seemed to discourage other researchers for some time, but recently Erlenkotter [11] devised a multiplier adjustment method for the Lagrangian relaxation of the uncapacitated location problem given in [10] in the case where the number of facilities located is unconstrained. Although discovered independently, Erlenkotter's algorithm is a variation on a method of Bilde and Krarup that was first described in 1967 in a Danish working paper and later published in English as [4]. While there has been no direct comparison, Erlenkotter's method appears to perform considerably better than the subgradient method. Fisher and Hochbaum [19] have experimented with multiplier adjustment for another location problem and found the method to work well, but not quite so well as the subgradient method.

Fisher, Jaikumar, and Van Wassenhove [21] have successfully developed a multiplier adjustment method for the generalized assignment problem in which one multiplier at a time is increased. This method has led to a substantially improved algorithm for the generalized assignment problem.

7. How Good are the Bounds?

The "answer" to this question that is available in the literature is completely problem specific and largely empirical. Most of the empirical results are summarized in Table 2. Each line of this table corresponds to a paper on a particular application of Lagrangian relaxation and gives the problem type, the source in which the computational experience is given, the number of problems attempted, the percentage of problems for which a u was discovered with $Z_D(u) = Z_D = Z$, and the average value of $Z_D(u^*) \times 100$ divided by the average value of Z, where $Z_D(u^*)$ denotes the largest bound discovered for each problem instance. Except as noted for the generalized assignment problem, all samples included a reasonable number of large problems. In some cases the sample included significantly larger problems than had been previously attempted. Frequently, standard test problems known for their difficulty were included. Table 2 is based on the results reported in each reference for all problems for which complete information was given. Of course, Table 2 gives highly aggregated information, and interested readers are urged to consult the appropriate references.

TABLE 2

Computational Experience With Lagrangian Relaxation

Problem Type	Source	Number of Problems Solved	Percentage of Problems with $Z_D = Z$	$\frac{\text{Ave. } Z_D(u^*)}{\text{Ave. } Z} \times 100$
TRAVELING SALESMAN				
Symmetric	[28]	18	55.5	99.5
Asymmetric	[3]	8	0.0	97.5
SCHEDULING				
n/m weighted Tardiness	[15]	8	37.5	96.2
1 Machine Weighted Tardiness	[18]	63	49.2	99.6
Power Generation Systems	[36]	15	0.0	98.9
GENERAL IP	[17]	11	0.0	83.2
LOCATION				
Uncapacitated	[10]	33	66.6	99.9
Capacitated	[23]	6	50.0	99.4
Databases in Computer Networks	[19]	29	51.7	95.9
GENERALIZED ASSIGNMENT				
Lagrangian* Relaxation 1	[8]	249**	96.0	99.8
Lagrangian* Relaxation 1	[21]	15	80.0	98.6
Lagrangian* Relaxation 2	[8]	249**	0.0	69.1

*See §3 for a definition of Lagrangian relaxations 1 and 2.

**Mostly small problems. The largest had $m = 9$ and $n = 17$.

These results provide overwhelming evidence that the bounds provided by Lagrangian relaxation are extremely sharp. It is natural to ask why Lagrangian bounds are so sharp. I am aware of only one analytic result that even begins to answer this question. This result was developed by Cornuejols, Fisher and Nemhauser [10] for the K-median problem.

Given n possible facility locations, m markets, and a nonnegative value c_{ij} for serving market i from a facility at location j, the K-median problem asks where K facilities should be located to maximize total value. Let

$$y_j = \begin{cases} 1, & \text{if a facility is placed in location } j, \\ 0, & \text{otherwise;} \end{cases}$$

$$x_{ij} = \begin{cases} 1, & \text{if market } i \text{ is served from location } j, \\ 0, & \text{otherwise.} \end{cases}$$

If $y_j = 0$ we must have $x_{ij} = 0$ for all i. Thus the K-median problem can be formulated as the integer program

$$Z = \max \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}, \tag{5}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad i = 1, \ldots, m, \tag{6}$$

$$\sum_{j=1}^{n} y_j = K, \tag{7}$$

$$0 \leqslant x_{ij} \leqslant y_j \leqslant 1, \quad \text{for all } i \text{ and } j, \tag{8}$$

$$x_{ij} \text{ and } y_j \text{ integral}, \quad \text{for all } i \text{ and } j. \tag{9}$$

A Lagrangian relaxation is obtained by dualizing constraints (6).

$$Z_D(u) = \max \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{i=1}^{m} u_i \left(\sum_{j=1}^{m} x_{ij} - 1 \right)$$

subject to (7), (8) and (9)

$$= \max \sum_{i=1}^{m} \sum_{j=1}^{n} (c_{ij} + u_i) x_{ij} - \sum_{i=1}^{m} u_i$$

subject to (7), (8) and (9).

This problem has the 0-1 VUB structure described in §4. To solve, we first observe that the VUB constraints (8) and the objective of the Lagrangian problem imply that

$$x_{ij} = \begin{cases} y_j, & \text{if } c_{ij} + u_i \geqslant 0, \\ 0, & \text{otherwise.} \end{cases}$$

Hence, defining $\bar{c}_j = \sum_{i=1}^{m} \max(0, c_{ij} + u_i)$, optimal y_j's must solve

$$\max \quad \sum_{j=1}^{n} \bar{c}_j y_i,$$

$$\sum_{j=1}^{n} y_j = K,$$

$$y_j = 0 \text{ or } 1, \qquad j = 1, \ldots, n,$$

which is a trivial problem.

Let $Z_D = \min_u Z_D(u)$ and assume $Z_D > 0$. Cornuejols, Fisher and Nemhauser [10] proved that

$$(Z_D - Z)/Z_D \leqslant \left(\frac{K-1}{K}\right)^K < \frac{1}{e}$$

and exhibited examples that show this bound to be the best possible.

This is an interesting first step towards understanding why Lagrangian relaxation has worked well on so many problems. Further study of this type is needed to understand and better exploit the power of Lagrangian relaxation.

8. Selecting Between Competing Relaxations

Two properties are important in evaluating a relaxation: the sharpness of the bounds produced and the amount of computation required to obtain these bounds. Usually selecting a relaxation involves a tradeoff between these two properties; sharper bounds require more time to compute. It is generally difficult to know whether a relaxation with sharper bounds but greater computation time will result in a branch and bound algorithm with better overall performance. However, it is usually possible to at least compare the bounds and computation requirements for different relaxations. This will be demonstrated for the generalized assignment example.

Two Lagrangian relaxations, ($LR1_u$) and ($LR2_v$), were defined for this problem. The linear programming relaxation of formulation (1)–(4) provides a third relaxation.

Consider first the computational requirements for each relaxation. We know that solving ($LR1_u$) requires time bounded by $n\sum_{i=1}^{m} b_i$ and solving ($LR2_v$) requires time proportional to nm. From this it would seem that the first relaxation requires greater computation, although it is difficult to know how many times each Lagrangian problem must be solved in optimizing the duals. It is also impossible to know analytically the time required to solve the LP relaxtion of (1)–(4).

Reference [8] reports computational times for the three relaxations for examples ranging in size from $m = 4$ and $n = 6$ to $m = 9$ and $n = 17$. The subgradient method was used to optimize the dual problems. On average, the first relaxation required about 50% more computational time than the second. This is much less than would be expected from comparison of worst-case bounds on times to solve Lagrangian problems because the subgradient method converged more quickly for the first relaxation. Solving the LP relaxation required one-fourth of the time for ($LR1_u$) for small problems but 2.5 times for large problems.

Now consider the relative sharpness of the bounds produced by these relaxations. Let $Z_{D1} = \max_u Z_{D1}(u)$, let $Z_{D2} = \max_{v \geqslant 0} Z_{D2}(v)$, and let Z_{LP}^{GA} denote the optimal value of the LP relaxation of (1)–(4).

A glance at the computational experience reported in the last two lines of Table 2 for the two Lagrangian relaxations strongly suggests that relaxation 1 produces much sharper bounds than relaxation 2. This observation can be verified using an analytic result given by Geoffrion [22]. This result will also allow us to compare Z_{D1} and Z_{D2} with Z_{LP}^{GA}.

The result states that in general $Z_D \geqslant Z_{LP}$. Conditions are also given for $Z_D = Z_{LP}$.

The fact that $Z_D \geqslant Z_{LP}$ can be established by the following series of relations between optimization problems.

$$Z_D = \max_u \left\{ \min_x cx + u(Ax - b) \right\}$$

$$\text{s.t.} \begin{array}{l} Dx \geqslant e, \\ x \geqslant 0 \text{ and integral;} \end{array}$$

$$\geqslant \max_u \left\{ \min_x cx + u(Ax - b) \right\}$$

$$\text{s.t.} \begin{array}{l} Dx \geqslant e, \\ x \geqslant 0 \end{array}$$

$$\text{(By LP duality)} = \max_u \max_{v \geqslant 0} ve - ub$$

$$\text{s.t. } vD \leqslant c + uA$$

$$\text{(By LP duality)} = \min_x cx$$

$$\text{s.t.} \begin{array}{l} Ax = b, \\ Dx \geqslant e, \\ x \geqslant 0, \end{array}$$

$$= Z_{LP}.$$

This logic also reveals a sufficient condition for $Z_D = Z_{LP}$. Namely, $Z_D = Z_{LP}$ whenever $Z_D(u)$ is not increased by removing the integrality restriction on x from the constraints of the Lagrangian problem. Geoffrion [22] calls this the integrality property.

Applying these results to the generalized assignment problem establishes that $Z_{D1} \geqslant Z_{D2} = Z_{LP}$ since the second Lagrangian relaxation has the integrality property while the first does not.

It should be emphasized that the integrality property is *not* defined relative to a given problem class but relative to a given *integer programming formulation* of a problem class. This is an important distinction because a problem often has more than one formulation. The Lagrangian relaxation of the K-median problem given in Section 7 has the integrality property if one takes (P) to be formulation (5)–(9). This fact alone is misleading since there is another formulation of the K-median problem in which constraints (8) are replaced by

$$\sum_{i=1}^{m} x_{ij} \leqslant my_j, \qquad j = 1, \ldots, n, \tag{8'a}$$

$$0 \leqslant x_{ij} \leqslant 1, \quad \text{for all } i \text{ and } j, \tag{8'b}$$

$$0 \leqslant y_j \leqslant 1, \qquad j = 1, \ldots, n. \tag{8'c}$$

This formulation is much more compact than (5)–(9) and is the one used in most LP-based branch and bound algorithms for the K-median problem. The Lagrangian relaxation given previously can be defined equivalently in terms of this formulation

but relative to this formulation, it does not have the integrality property. In fact, it is shown in [10] that the Lagrangian bound Z_D and the LP value of (5), (6), (7), (8), (9) are substantially sharper than the LP value of (5), (6), (7), (8′) and (9). Others (Williams [45, 46] and Mairs, et al. [32]) have also noted that there are frequently alternative IP formulations for the same problem that have quite different LP properties.

It is also worth noting that many other successful Lagrangian relaxations (including Held and Karp [27], [28], Etcheberry [12], Etcheberry, et al. [13], and Fisher and Hochbaum [19]) have had the integrality property. For these applications Lagrangian relaxation was successful because the LP relaxation closely approximated (P) and because the method used to optimize (D) (usually the subgradient method) was more powerful then methods available for solving the (generally large) LP relaxation of (P). The important message of these applications is that combinatorial optimization problems frequently can be formulated as a large IP whose LP relaxation closely approximates the IP and can be solved quickly by dual methods. To exploit this fact, future research should be broadly construed to develop methods for solving the large structured LP's arising from combinatorial problems and to understand the properties of combinatorial problems that give rise to good LP approximations. There has already been significant research on methods other than Lagrangian relaxation for exploiting the special structure of LP's derived from combinatorial problems. Schrage [41], Miliotis [34], [35], and Christofides and Whitlock [9] have given clever LP solution methods that exploit certain types of structure that are common in formulations of combinatorial problems.

9. Feasible Solutions

This section is concerned with using (LR_u) to obtain feasible solutions for (P). It is possible in the course of solving (D) that a solution to (LR_u) will be discovered that is feasible in (P). Because the dualized constraints $Ax = b$ are equalities, this solution is also optimal for (P). If the dualized constraints contain some inequalities, a Lagrangian problem solution can be feasible but nonoptimal for (P). However, it is rare that a feasible solution of either type is discovered. On the other hand, it often happens that a solution to (LR_u) obtained while optimizing (D) will be nearly feasible for (P) and can be made feasible with some judicious tinkering. Such a method might be called a Lagrangian heuristic. After illustrating this approach for the generalized assignment problem and ($LR1_u$), we will discuss computational experience with Lagrangian heuristics for other problems.

It is convenient to think of the generalized assignment problem as requiring a packing of n items into m knapsacks using each item exactly once. In ($LR1_u$) the constraints $\sum_{i=1}^{m} x_{ij} = 1, j = 1, \ldots, n$ requiring that each item be used exactly once are dualized and may be violated. Let $\bar{x}$ denote an optimal solution to ($LR1_u$). Partition $N = \{1, \ldots, n\}$ into three sets defined by

$$S_1 = \left\{ j \in J \mid \sum_{i=1}^{m} \bar{x}_{ij} = 0 \right\},$$

$$S_2 = \left\{ j \in J \mid \sum_{i=1}^{m} \bar{x}_{ij} = 1 \right\},$$

$$S_3 = \left\{ j \in J \mid \sum_{i=1}^{m} \bar{x}_{ij} > 1 \right\}.$$

The constraints of (P) which are violated by $\bar{x}$ correspond to $j \in S_1 \cup S_3$. We wish to modify $\bar{x}$ so that these constraints are satisfied. This is easy for a $j \in S_3$. Simply remove item j from all but one knapsack. A variety of rules could be used to determine in which knapsack to leave item j. For example, it would be reasonable to choose the knapsack that maximizes $(u_i - c_{ij})/a_{ij}$.

To complete the construction of a feasible solution it is only necessary to assign items in S_1 to knapsacks. While there is no guarantee that this can be done, the chances of success should be good unless the knapsack constraints are very tight. Many assignment rules are plausible, such as the following one that is motivated by bin packing heuristics. Order items in S_1 by decreasing value of $\sum_{i=1}^{m} a_{ij}$ and place each item in turn into a knapsack with sufficient capacity that maximizes $(u_i - c_{ij})/a_{ij}$.

Several researchers have reported success using Lagrangian problem solutions obtained during the application of the subgradient method to construct primal feasible solutions. For example, this is easy to do for the K-median problem. Let $\bar{x}, \bar{y}$ denote a feasible solution to the Lagrangian problem defined in §7 for the K-median problem. Let $S = \{j \mid \bar{y}_j = 1\}$ and for each i set $\bar{\bar{x}}_{ij} = 1$ for a j that solves $\max_{j' \in S} c_{ij'}$. Set $\bar{\bar{x}}_{ij} = 0$ for remaining ij. The solution $\bar{\bar{x}}, \bar{y}$ is feasible and represents the best assignment of x given $\bar{y}$. Cornuejols, Fisher and Nemhauser [10] found that this approach performed as well as the best of several other heuristics they tested.

Fisher [18] reports experience for the problem of sequencing n jobs on one machine to minimize a tardiness function. A Lagrangian solution is a set of start times $\bar{x}_1, \ldots, \bar{x}_n$ for the n jobs that may violate the machine constraints. A primal feasible solution is obtained by sequencing jobs on the machine in order of increasing $\bar{x}_j$ values. This rule was tested on 63 problems. It was applied in conjunction with the subgradient method after an initial feasible solution had been generated by a greedy heuristic. The greedy heuristic found an optimal solution for 18 of the problems. The Lagrangian heuristic found optimal solutions to 21 of the remaining 45 problems. On average the greedy value was 100.4% of the optimal value while the value of the solution produced by the Lagrangian heuristic was 100.16% of the optimal value.

10. Using Lagrangian Relaxation in Branch and Bound

The issues involved in designing a branch and bound algorithm that uses a Lagrangian relaxation are essentially the same as those that arise when a linear programming relaxation is used. Some of these issues are illustrated here for the generalized assignment problem and ($LR1_u$) derived in §3.

A natural branching tree for this problem is illustrated in Figure 2. This tree exploits the structure of constraints (2) by selecting a particular index j when branching and requiring exactly one variable in the set x_{ij}, $i = 1, \ldots, m$, to equal 1 along each branch.

A Lagrangian relaxation (presumably $LR1_u$ given the discussion in §8) can be used at each node of this tree to obtain lower bounds and feasible solutions.

We note that the Lagrangian problem defined at a particular node of this tree has the same structure as ($LR1_u$) and is no harder to solve. This is an obvious property that must hold for any application. Sometimes it is desirable to design the branching rules to achieve this property (e.g., Held and Karp [28]).

There are several tactical decisions that must be made in any branch and bound scheme such as which node to explore next and what indices (j_1, j_2 and j_3 in Figure 2) to use in branching. Lagrangian relaxation can be used in making these decisions in

much the same way that linear programming would be used. For example, we might choose to branch on an index j for which $u_j(\sum_{i=1}^{m} x_{ij} - 1)$ is large in the current Lagrangian problem solution in order to strengthen the bounds as much as possible.

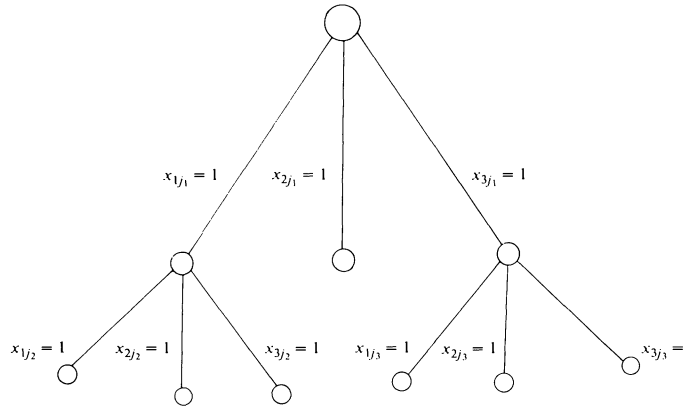

FIGURE 2. Partial Branching Tree for the Generalized Assignment Problem with $m = 3$.

Finally, we note that the method for optimizing (D) must be carefully integrated into the branch and bound algorithm to avoid doing unnecessary work when (D) is reoptimized at a new node. A common strategy when using the subgradient method is to take u^0 equal to the terminal value of u at the previous node. The subgradient method is then run for a fixed number of iterations that depends on the type of node being explored. At the first node a large number of iterations is used. When branching down a small number is used, and when backtracking, an intermediate number.

11. Conclusions and Future Research Directions

Lagrangian relaxation is an important new computational technique in the management scientist's arsenal. This paper has documented a number of successful applications of this technique, and hopefully will inspire other applications. Besides additional applications, what opportunities for further research exist in this area? The most obvious is development of more powerful technology for optimizing the nondifferentiable dual function. Nondifferentiable optimization has become an important general research area that surely will continue to grow. One corner of this area that seems to hold great promise for Lagrangian relaxation is the development of multiplier adjustment methods of the type described at the end of §6. The enormous success that has been obtained with this approach on the uncapacitated location [11] and the generalized assignment problems [21] suggests that it should be tried on other problems. Two other research areas that deserve further attention are the development and analysis of Lagrangian heuristics as described in §9 and the analysis (worst-case or probabilistic) of the quality of the bounds produced by Lagrangian relaxation as discussed in §7 and [10].[1]

[1]This paper was supported in part by NSF Grant ENG-7826500 to the University of Pennsylvania.

References

1. BALAS, E. AND CHRISTOFIDES, N., Talk presented at the Ninth International Synposium on Mathematical Programming, Budapest, August, 1976.
2. BALINSKI, M. L. AND WOLFE, P., Eds. *Nondifferentiable Optimization*, Math Programming Study, Vol. 3 (November 1975).
3. BAZARRA, M. S. AND GOODE, J. J., "The Traveling Salesman Problem: A Duality Approach," *Math. Programming*, Vol. 13 (1977), pp. 221–237.
4. BILDE, O. AND KRARUP, J., "Sharp Lower Bounds and Efficient Algorithms for the Simple Plant Location Problem," *Ann. Discrete Math.*, Vol. 1 (1977), pp. 79–97.
5. BURDET, C. A. AND JOHNSON, E. L., "A Subadditive Approach to Solve Linear Integer Programs," *Ann. Discrete Math.*, Vol. 1 (1977), pp. 117–143.
6. CAMERINI, P. M., FRATTA, L. AND MAFFIOLI, F., "On Improving Relaxation Methods by Modified Gradient Techniques," *Math. Programming Study*, Vol. 3 (1975), pp. 26–34.
7. ——— AND MAFFIOLI, F., "Heuristically Guided Algorithms for K-Parity Matroid Problems," *Discrete Math.*, (1978), pp. 103–116.
8. CHALMET: L. G. AND GELDERS, L. F., "Lagrangian Relaxations for a Generalized Assignment-Type Problem." *Proc. Second European Congress on Operations Research*, North-Holland, Amsterdam, 1976, pp. 103–109.
9. CHRISTOFIDES, N. AND WHITLOCK, C., "An LP-based TSP Algorithm," Imperial College Report 78-79 (1978).
10. CORNUEJOLS, G., FISHER, M. L. AND NEMHAUSER, G. L., "Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Approximate Algorithms," *Management Sci.*, Vol. 23 (1977), pp. 789–810.
11. ERLENKOTTER, D., "A Dual-Based Procedure for Uncapacitated Facility Location," *Operations Res.*, Vol. 26, No. 1 (1978), pp. 992–1009.
12. ETCHEBERRY, J., "The Set-Covering Problem: A New Implicit Enumeration Algorithm," *Operations Res.*, Vol. 25 (1977), pp. 760–772.
13. ———, CONCA, C. AND STACCHETTI, E., "An Implicit Enumeration Approach for Integer Programming Using Subgradient Optimization," Pub. No. 78/04/c, Universidad de Chile, March 1978.
14. EVERETT, H., "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources," *Operations Res.*, Vol. 11 (1963), pp. 399–417.
15. FISHER, M. L., "Optimal Solution of Scheduling Problems Using Lagrange Multipliers: Part I," *Operations Res.*, Vol. 21 (1973), pp. 1114–1127.
16. ——— AND SHAPIRO, J. F., "Constructive Duality in Integer Programming." *SIAM J. Appl. Math.*, Vol. 27 (1974), pp. 31–52.
17. ———, NORTHUP, W. D. AND SHAPIRO, J. F., "Using Duality to Solve Discrete Optimization Problems: Theory and Computational Experience," *Math. Programming Study*, Vol. 3 (1975), pp. 56–94.
18. ———, "A Dual Algorithm for the One-Machine Scheduling Problem," *Math. Programming*, Vol. 11 (1976), pp. 229–251.
19. ——— AND HOCHBAUM, D. S., "Database Location in Computer Networks," *J. Assoc. Comput. Mach.* (to appear).
20. ———, NEMHAUSER, G. L. AND WOLSEY, L. A., "An Analysis of Approximations for Finding a Maximum Weight Hamiltonian Circuit," *Operations Res.*, Vol. 27, No. 4 (1979), pp. 799–809.
21. ———, JAIKUMAR, R. AND VAN WASSENHOVE, L., "A Multiplier Adjustment Method for the Generalized Assignment Problem," Decision Sciences Working Paper, University of Pennsylvania, May, 1980.
22. GEOFFRION, A. M., "Lagrangian Relaxation and its Uses in Integer Programming," *Math. Programming Study*, Vol. 2 (1974), pp. 82–114.
23. ——— AND MCBRIDE, R., "Lagrangian Relaxation Applied to Capacitated Facility Location Problems," *AIIE Trans.*, Vol. 10 (1978), pp. 40–47.
24. GILMORE, P. C. AND GOMORY, R. E., "A Linear Programming Approach to the Cutting-Stock Problem, Part II," *Operations Res.*, Vol. 11 (1963), pp. 863–888.
25. GOFFIN, J. L., "On the Convergence Rates of Subgradient Optimization Methods," *Math. Programming*, Vol. 13 (1977), pp. 329–347.
26. HELBIG HANSEN, K. AND KRARUP, J., "Improvements of the Held-Karp Algorithm for the Symmetric Traveling-Salesman Problem," *Math. Programming*, Vol. 7 (1974), pp. 87–96.

27. HELD, M. AND KARP, R. M., "The Traveling Salesman Problem and Minimum Spanning Trees," *Operations Res.*, Vol. 18 (1970), pp. 1138–1162.
28. ——— AND ———, "The Traveling Salesman Problem and Minimum Spanning Trees: Part II," *Mathematical Programming*, Vol. 1 (1971), pp. 6–25.
29. ———, WOLFE, P. AND CROWDER, H. D., "Validation of Subgradient Optimization," *Mathematical Programming*, Vol. 6 (1974), pp. 62–88.
30. HOGAN, W. W., MARSTEN, R. E. AND BLANKENSHIP, J. W., "The Boxstep Method for Large Scale Optimization " *Operations Res.*, Vol. 23 (1975), p. 3.
31. LORIE, J. AND SAVAGE, L. J., "Three Problems in Capital Rationing," *J. Business*, (1955), pp. 229–239.
32. MAIRS, T. G., WAKEFIELD, G. W., JOHNSON, E. L. AND SPIELBERG, K., "On a Production Allocation and Distribution Problem," *Management Sci.*, Vol. 24, No. 15, (1978), pp. 1622–1630.
33. MARSTEN, R., "The Use of the Boxstep Method in Discrete Optimization," *Math. Programming Study*, Vol. 3 (1975), pp. 127–144.
34. MILIOTIS, T., "Integer Programming Approaches to the Travelling Salesman Problem:" *Math. Programming*, Vol. 10 (1976), pp. 367–378.
35. ———, "An All-Integer Arithmetic LP-Cutting Planes Code Applied to the Travelling Salesman Problem." London School of Economics Working Paper, 1976.
36. MUCKSTADT, J. AND KOENIG, S. A., "An Application of Lagrangian Relaxation to Scheduling in Power Generation Systems," *Operations Res.*, Vol. 25 (1977), pp. 387–403.
37. MULVEY: J. AND CROWDER, H., "Cluster Analysis: An Application of Lagrangian Relaxation," *Management Sci.*, Vol. 25 (1979), pp. 329–340.
38. NEMHAUSER, G. L. AND WEBER, G., "Optimal Set Partitioning, Matchings and Lagrangian Duality," Talk delivered at the New York ORSA/TIMS Meeting, May 3, 1978.
39. POJAK, B. T., "A General Method for Solving Extremum Problems," *Soviet Math. Dokl.*, Vol. 8 (1967), pp. 593–597.
40. ROSS, G. T. AND SOLAND, R. M., "A Branch and Bound Algorithm for the Generalized Assignment Problem," *Math. Programming*, Vol. 8 (1975), pp. 91–103.
41. SCHRAGE, L., "Implicit Representation of Variable Upper Bounds in Linear Programming," *Math. Programming Studies*, Vol. 4 (1975), pp. 118–132.
42. SHAPIRO, J. F., "Generalized Lagrange Multipliers in Integer Programming," *Operations Res.*, Vol. 19 (1971), pp. 68–76.
43. ———, "A Survey of Lagrangian Techniques for Discrete Optimization," *Ann. Discrete Math.*, Vol. 5 (1979), pp. 113–138.
44. ———, *Mathematical Programming: Structures and Algorithms*, Wiley, New York, 1979.
45. WILLIAMS, H. P., "Experiments in the Formulation of Integer Programming Problems," *Math. Programming Study*, Vol. 2 (1974), pp. 180–197.
46. ———, "The Reformulation of Two Mixed Integer Programming Problems," *Math. Programming*, Vol. 14 (1975), pp. 325–331.