An Applications Oriented Guide to Lagrangian Relaxation

MARSHALL L. FISHER

Department of Decision Sciences
The Wharton School
University of Pennsylvania
Philadelphia, Pennsylvania 19104

Lagrangian relaxation is a tool that is increasingly being used in large-scale mathematical programming applications, such as last year's CPMS/TIMS Management Achievement Award winner (Bell et al. 1983). In this tutorial, Marshall Fisher provides a practical guide to the use of the approach with many examples and illustrations.

In the last decade, Lagrangian relaxation has grown from a successful but largely theoretical concept to a tool that is the backbone of a number of large-scale applications. While there have been several surveys of Lagrangian relaxation (for example, Fisher [1981] and Geoffrion [1974]) and an excellent textbook treatment [Shapiro 1979], more extensive use of Lagrangian relaxation in practice has been hampered by the lack of a "how to do it" exposition similar to the treatment usually accorded linear, dynamic, and integer programming in operations research texts. This article is intended to at least partially fill that void and should be of interest to both developers and users of Lagrangian relaxation algorithms.

Lagrangian relaxation is based upon the observation that many difficult integer programming problems can be modeled as a relatively easy problem complicated by a set of side constraints. To exploit this observation, we create a Lagrangian problem in which the complicating constraints are replaced with a penalty term in the objective function involving the amount of violation of the constraints and their dual variables. The Lagrangian problem is easy to solve and provides an upper bound (for a maximization problem) on the optimal value of the original problem. It can thus

0092-2102/85/1502/0010$01.25
This paper was refereed.

PROGRAMMING
INTEGER-ALGORITHMS-LAGRANGIAN RELAXATION

INTERFACES **15**: 2 March-April 1985 (pp. 10-21)

be used in place of a linear programming relaxation to provide bounds in a branch and bound algorithm. The Lagrangian approach offers a number of important advantages over linear programming relaxations.

I will first formulate the Lagrangian relaxation concept in general terms and then demonstrate its use extensively on a numerical example. I begin with an integer programming problem of the following form:

$$Z = \max cx$$
$$Ax \leq b. \qquad \text{(P)}$$
$$Dx \leq e$$
$$x \geq 0 \text{ and integral,}$$

where x is $n \times 1$, b is $m \times 1$, e is $k \times 1$ and all other matrices have conformable dimensions.

We assume that the constraints of (P) have been partitioned into the two sets $Ax \leq b$ and $Dx \leq e$ so that (P) is relatively easy to solve if the constraint set $Ax \leq b$ is removed. To create the Lagrangian problem, we first define an m vector of nonnegative multipliers u and add the nonnegative term $u(b-Ax)$ to the objective function of (P) to obtain

$$\max cx + u(b-Ax)$$
$$\text{subject to } Ax \leq b$$
$$Dx \leq e$$
$$x \geq 0 \text{ and integral.}$$

It is clear that the optimal value of this problem for u fixed at a nonnegative value is an upper bound on Z because we have merely added a nonnegative term to the objective function. At this point, we create the Lagrangian problem by removing the constraints $Ax \leq b$ to obtain

$$Z_D(u) = \max cx + u(b-Ax)$$
$$Dx \leq e$$
$$x \geq 0 \text{ and integral.}$$

Since removing the constraints $Ax \leq b$ cannot decrease the optimal value, $Z_D(u)$ is also an upper bound on Z. Moreover, by assumption the Lagrangian problem is relatively easy to solve.

There are three major questions in designing a Lagrangian-relaxation-based system: (a) which constraints should be relaxed, (b) how to compute good multipliers u, and (c) how to deduce a good, feasible solution to the original problem, given a solution to the relaxed problem. Roughly speaking, the answer to (a) is that the relaxation should make the problem significantly easier, but not too easy. For (b) there is a choice between a general purpose procedure called the subgradient method and "smarter" methods which may be better but which are, however, highly problem specific. Similarly, the answer to (c) tends to be problem specific. I will use a numerical example to illustrate considerations (a), (b), and (c) as well as to compare Lagrangian relaxation to the use of linear programming to obtain bounds for use in a branch and bound algorithm. I will conclude with a small survey of past applications and an assessment of future prospects.

An Example

The example shown below will be used throughout the paper to demonstrate concepts.

$$Z = \max 16x_1 + 10x_2 + 4x_4 \qquad (1)$$
$$\text{subject to } 8x_1 + 2x_2 + x_3 + 4x_4 \leq 10 \qquad (2)$$
$$x_1 + x_2 \leq 1 \qquad (3)$$
$$x_3 + x_4 \leq 1 \qquad (4)$$
$$0 \leq x_j \leq 1, j=1, \ldots, 4 \qquad (5)$$
$$x_j \text{ integral}, j=1, \ldots, 4 \qquad (6)$$

If we dualize constraint (2), we obtain the

following Lagrangian relaxation.

$$Z_D(u) = \max (16\text{-}8u)x_1 + (10\text{-}2u)x_2 + (0\text{-}u)x_3 + (4\text{-}4u)x_4 + 10u$$

$$x_1 + x_2 \leq 1 \quad (3)$$

$$x_3 + x_4 \leq 1 \quad (4)$$

$$0 \leq x_j \leq 1, j=1, \ldots, 4 \quad (5)$$

$$x_j \text{ integral}, j=1, \ldots, 4 \quad (6)$$

It is easy to solve this relaxation if the dual variable u is fixed at some nonnegative value. Note that if the objective coefficient of any variable is not positive, we can set that variable to 0. Otherwise, we choose either x_1 or x_2 and either x_3 or x_4 to set to 1, depending on which has the larger objective function coefficient.

The Subgradient Method for Setting the Dual Variables

Here I'll use our numerical example to develop and demonstrate a method for obtaining dual variable values that produce a tight bound. Ideally, u should solve the following dual problem.

$$Z_D = \min Z_D(u), \qquad u \geq 0.$$

Before presenting an algorithm for this problem, it will be useful to develop some insight by trying different values for the single dual variable u in the example. Table 1 gives a list of seven values for u, together with the associated Lagrangian relaxation solution, the bound $Z_D(u)$, and Z for those Lagrangian solutions that are feasible in (P). In the case of $u = 1$, there are four alternative optimal Lagrangian solutions which are all shown. The values for u exhibited were obtained using an intelligent trial-and-error process. It is useful to think of the single constraint (2) that we have dualized as a resource constraint with the right side representing the available supply of some resource and the left side the amount of the resource demanded in a particular solution. We can then interpret the dual variable u as a "price" charged for the resource. It turns out that if we can discover a price for which the supply and demand for the resource are equal, then this value will also give a tight upper bound. However, such a price might not exist. Beginning with $u = 0$, we discover that the Lagrangian relaxation solution demand for the resource exceeds the available supply by two units, suggesting that we should use a larger value for u. We next try $u = 6$ and discover that we have over-corrected in the sense that all variables are 0 in the Lagrangian solution and none of the resource is used. We next try a sequence of dual values in the interval between 0 and 6, obtaining the results shown in Table 1.

	Lagrangian Solution					Value of Lagrangian solution if feasible
u	x_1	x_2	x_3	x_4	$Z_D(u)$	
0	1	0	0	1	20	
6	0	0	0	0	60	0
3	0	1	0	0	34	10
2	0	1	0	0	26	10
1	1	0	0	0	18	16
	1	0	0	1	18	
	0	1	0	0	18	10
	0	1	0	1	18	14
½	1	0	0	1	19	
¾	1	0	0	1	18.5	

Table 1: Lagrangian solutions for possible dual variable values.

For the values tested, the tightest bound of 18 was obtained with $u = 1$, but at the moment we lack any means for confirming that it is optimal. It is possible to demonstrate that 18 is the optimal value for $Z_D(u)$ by observing that if we substitute any x into the objective function for

the Lagrangian problem, we obtain a linear function in u. Figure 1 exhibits this family of linear functions for all Lagrangian relaxation solutions that are optimal for at least one value of u. The fact that we must maximize the Lagrangian objective means that for any particular value of u, $Z_D(u)$ is equal to the largest of these linear functions. Thus, the $Z_D(u)$ function is given by the upper envelope of this family of linear equations that is shown as a darkened piecewise linear function in Figure 1. From this figure it is easy to see that $u = 1$ minimizes $Z_D(u)$.

Figure 1 also provides motivation for a general algorithm for finding u. As shown, the $Z_D(u)$ function is convex and differentiable except at points where the Lagrangian problem has multiple optimal solutions. At differentiable points, the derivative of $Z_D(u)$ with respect to u is given by $8x_1 + 2x_2 + x_3 + 4x_4 - 10$, where x is an optimal solution to (LR_u). These facts also hold in general with the gradient of the $Z_D(u)$ function at differentiable points given by $Ax - b$. These observations suggest that it might be fruitful to apply a gradient method to minimization of $Z_D(u)$ with some adaptation at the points where $Z_D(u)$ is nondifferentiable. This has been nicely accomplished in a procedure called the subgradient method. At points where $Z_D(u)$ is nondifferentiable, the subgradient method chooses arbitrarily from the set of alternative optimal Lagrangian solutions and uses the vector $Ax - b$ for this solution as though it were the gradient of $Z_D(u)$. The result is a procedure that determines a sequence of values for u by beginning at an initial point u^o and applying the formula

$$u^{k+1} = \max\{0,\ u^k - t_k(b - Ax^k)\}. \quad (6)$$

In this formula, t_k is a scalar stepsize and x^k is an optimal solution to $(LR_u{}^k)$, the Lagrangian problem with dual variables set to u^k.

The nondifferentiability also requires some variation in the way the stepsize is normally set in a gradient method. To gain insight into a sensible procedure for setting t_k, we have provided in Tables 2, 3, and 4 the results of the subgradient method applied to the example with three different rules for t_k. In the first case, t_k is fixed at one on all iterations, and we see that the subgradient method oscillates between the values $u = 0$ and $u = 2$. In the second example, t_k converges to 0 with each successive value equal to half the value on the previous iteration. In this case, the subgradient method behaves

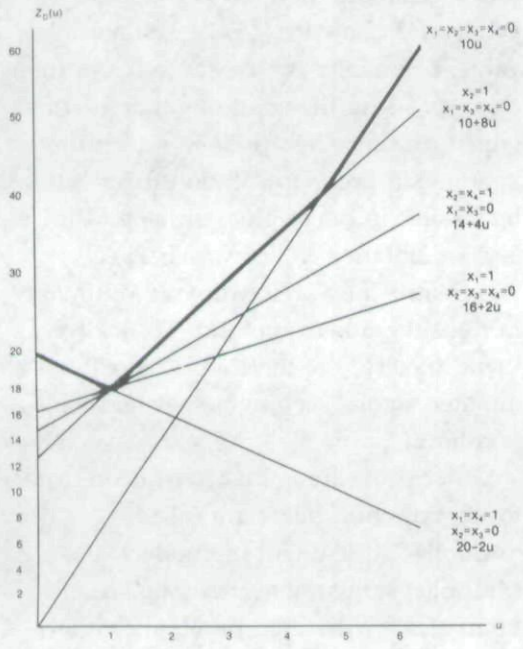

Figure 1: The piecewise linear $Z_D(u)$ function.

$u^0 = 0$
$u^1 = 0 - (-2) = 2$
$u^2 = \max\{0, 2\text{-}8\} = 0$
$u^3 = 0 - (-2) = 2$
$u^4 = \max\{0, 2\text{-}8\} = 0$

Table 2: Subgradient method with $t_k = 1$ for all k.

$u^0 = 0$
$u^1 = 0 - (-2) = 2$
$u^2 = \max\{0, 2 - \frac{1}{2}(8)\} = 0$
$u^3 = 0 - \frac{1}{4}(-2) = \frac{1}{2}$
$u^4 = \frac{1}{2} - \frac{1}{8}(-2) = \frac{3}{4}$
$u^5 = \frac{3}{4} - 1/16(-2) = \frac{7}{8}$
$u^6 = \frac{7}{8} - 1/32(-2) = 15/16$

Table 3: Subgradient method with $t_k = 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots$

$u^0 = 0$
$u^1 = 2$
$u^2 = \max\{0, 2 - 1/3(8)\} = 0$
$u^3 = 0 - 1/9(-2) = 2/9$
$u^4 = 2/9 - 1/27(-2) = .296$
$u^5 = .296 - 1/81(-2) = .321$
$u^6 = .321 - 1{-}243(-2) = .329$
$u^7 = .329 - 1/729(-2) = .332$

Table 4: Subgradient method with $t_k = 1, 1/3, 1/9, 1/27, 1/81, \ldots$

nicely and converges to the optimal value of $u = 1$. In the final case, t_k also converges to 0, but more quickly. Each successive value is equal to one-third the value on the previous iteration. In this case the subgradient method converges to $u = 1/3$, showing that if the stepsize converges to 0 too quickly, then the subgradient method will converge to a point other than the optimal solution.

From these examples we suspect that the stepsize in the subgradient method should converge to 0, but not too quickly. These observations have been confirmed in a result (see Held, Wolfe, and Crowder [1974]) that states that if as $k \longrightarrow \infty$,

$$t_k \to 0 \quad \text{and} \quad \sum_{i=1}^{k} t_i \to \infty$$

then $Z_D(u^k)$ converges to its optimal value Z_D. Note that Table 3 actually violates the second condition since $\Sigma\, t_i \longrightarrow 2$, thus showing that these conditions are sufficient but not necessary. A formula for t_k that has proven effective in practice is

$$t_k = \frac{\lambda_k\left(Z_D(u^k) - Z^*\right)}{\sum_{i=1}^{m}\left(b_i - \sum_{j=1}^{n} a_{ij} x_j^k\right)^2}.$$

In this formula, Z^* is the objective value of the best known feasible solution to (P) and λ_k is a scalar chosen between 0 and 2. Frequently, the sequence λ_k is determined by starting with $\lambda_k = 2$ and reducing λ_k by a factor of two whenever $Z_D(u^k)$ has failed to decrease in a specified number of iterations. Justification for this formula, as well as many other interesting results on the subgradient method, is given in Held, Wolfe and Crowder [1974]. The feasible value Z^* initially can be set to 0 and then updated using the solutions that are obtained on those iterations in which the Lagrangian problem solution turns out to be feasible in the original problem (P). Unless we obtain a u^k for which $Z_D(u^k) = Z^*$, there is no way of proving optimality in the subgradient method. To resolve this difficulty, the method is usually terminated upon reaching a specified iteration limit.

Other procedures that have been used for setting multipliers are called multiplier-adjustment methods. Multiplier-adjustment methods are heuristics for the dual problem that are developed for a specific application and exploit some special structure of the dual

problem in that application. The first highly successful example of a multiplier-adjustment method was Erlenkotter's [1978] algorithm for the uncapacitated location problem.

By developing a multiplier-adjustment method specifically tailored for some problem class, one is usually able to improve on the subgradient method. However, because the subgradient method is easy to program and has performed robustly in a wide variety of applications, it is usually at least the initial choice for setting the multipliers in Lagrangian relaxation.

Returning to our example, we have obtained through the application of Lagrangian relaxation and the subgradient method a feasible solution with a value of 16 and an upper bound on the optimal value of 18. At this point, we could stop and be content with a feasible solution proven to be within about 12 percent of optimality, or we could complete solution of the example to optimality using branch and bound, with bounds provided by our Lagrangian relaxation. In the next section I'll show how such an approach would compare with more traditional linear programming based branch and bound algorithms.

Comparison with Linear Programming Based Bounds

In this section I will compare Lagrangian relaxation with the upper bound obtained by relaxing the integrality requirement on x and solving the resulting linear program.

Let Z_{LP} denote the optimal value of (P) with integrality on x relaxed. Let's start by comparing Z_{LP} for the example with the best upper bound of 18 obtained previously with Lagrangian relaxation. To facilitate this comparison, we first write out the standard *LP* dual of the example. Let u, v_1, and v_2 denote dual variables on constraints (2), (3), and (4) and w_j a dual variable on the constraint $x_j \leq 1$. Then the *LP* dual of example (1) – (5) is

$$\begin{aligned} \min\ & 10u + v_1 + v_2 + w_1 + w_2 + w_3 \\ & + w_4 \\ & 8u + v_1 + w_1 \geq 16 \\ & 2u + v_1 + w_2 \geq 10 \\ & u + v_2 + w_3 \geq 0 \\ & 4u + v_2 + w_4 \geq 4 \\ & u, v_1, v_2, w_1, \ldots, w_4 \geq 0. \end{aligned}$$

The optimal solution to the primal *LP* is $x_1 = 1$, $x_2 = 0$, $x_3 = 0$, $x_4 = 1/2$ and the optimal solution to the dual *LP* is $u = 1$, $v_1 = 8$, $v_2 = w_1 = \ldots w_4 = 0$. To verify that each of these solutions is optimal, simply substitute them in the primal and dual and observe that each is feasible and gives the same objective value 18.

This exercise has demonstrated two interesting facts. First, $Z_{LP} = 18$, the same upper bound we obtained with Lagrangian relaxation. Secondly, the *LP* dual variable value of $u = 1$ on constraint (2) is exactly the value that gave the minimum upper bound of 18 on the Lagrangian problem. These observations are part of a pattern that holds generally and is nicely summarized in a result from Geoffrion [1974] which states that $Z_D \leq Z_{LP}$ for any Lagrangian relaxation. This fact is established by the following sequence of relations between optimization problems.

$$\begin{aligned} Z_D &= \min_{u \geq 0} \{\max_{x \quad Dx \leq e} (cx + u(b - Ax))\} \\ &\qquad x \geq 0 \text{ and integral} \\ &\leq \min \{\max (cx + u(b - Ax))\} \end{aligned}$$

$$
\begin{aligned}
& u \geq 0 \quad x \qquad Dx \leq e \\
& \qquad\qquad\qquad\quad x \geq 0 \\
\text{(by } LP \text{ duality)} \quad & = \min \{\min ub + ve\} \\
& \quad u \geq 0 \;\; v \geq 0 \\
& \qquad\qquad vD \geq c - uA \\
& = \min \quad ub + ve \\
& \quad u,v \geq o \;\; uA + vD \geq c \\
\text{(by } LP \text{ duality)} \quad & = \max cx \\
& \qquad\quad Ax \leq b \\
& \qquad\quad Dx \leq e \\
& \qquad\quad x \geq 0 \\
& = Z_{LP}
\end{aligned}
$$

Besides showing that $Z_D \leq Z_{LP}$, the preceding logic indicates when $Z_D = Z_{LP}$ and when $Z_D < Z_{LP}$. The inequality in the sequence of relations connecting Z_D and Z_{LP} is between the Lagrangian problem and the Lagrangian problem with integrality relaxed. Hence, we can have $Z_D < Z_{LP}$ only if this inequality holds strictly or conversely, $Z_D = Z_{LP}$ only if the Lagrangian problem is unaffected by removing the integrality requirement on x.

In the Lagrangian problem for the original example, the optimal values of the variables will be integer whether we require it or not. This implies that we must have $Z_D = Z_{LP}$, something that we have already observed numerically.

This result also shows that we can improve the upper bound by using a Lagrangian relaxation in which the variables are not naturally integral.

An Improved Relaxation

An alternative relaxation for the example is given below.

$$
\begin{aligned}
& Z_D(v_1,v_2) = \max (16\text{-}v_1)x_1 + (10\text{-}v_1)x_2 \\
& \quad + (0\text{-}v_2)x_3 + (4\text{-}v_2)x_4 + v_1 + v_2 \\
& \text{subject to } 8x_1 + 2x_2 + x_3 + 4x_4 \leq 10 \qquad (2) \\
& \qquad\qquad 0 \leq x_j \leq 1,\ j=1, \ldots, 4 \qquad (5) \\
& \qquad\qquad x_j \text{ integral},\ j=1, \ldots, 4 \qquad (6)
\end{aligned}
$$

In this relaxation, we have dualized constraints (3) and (4) and obtained a relaxation which is a knapsack problem. Although this problem is known to be difficult in the worst case, it can be solved practically using a variety of efficient knapsack algorithms such as dynamic programming. Because the continuous and integer solutions to the knapsack problem can differ, the analytic result obtained in the previous section tells us that this relaxation may provide bounds that are better than linear programming.

This is confirmed empirically in Table 5, which shows the application of the subgradient method to this relaxation. We begin with both dual variables equal to 0, and in four iterations, we converge to a dual solution in which the upper bound of 16 is equal to the objective value of the feasible solution obtained when we solve the Lagrangian problem. Hence, Lagrangian relaxation has completely solved the original problem. In this example we have set the stepsize using the formula given previously with $\lambda_k = 1$.

This example illustrates that with careful choice of which constraints to dualize, Lagrangian relaxation can provide results that are significantly superior to *LP*-based branch and bound. The choice of which constraints to dualize is to some extent an

			Lagrangian Solution					
v_1	v_2	λ_k	x_1	x_2	x_3	x_4	$Z_D(v_1,v_2)$	Z^*
0	0	1	1	1	0	0	26	0
13	0	1	0	0	0	1	17	4
			(feasible with $Z = 4$)					
0	0	1	1	1	0	0	26	4
11	0	1	1	0	0	0	16	16
			(feasible with $Z = 16$)					

Table 5: The subgradient method applied to the improved relaxation.

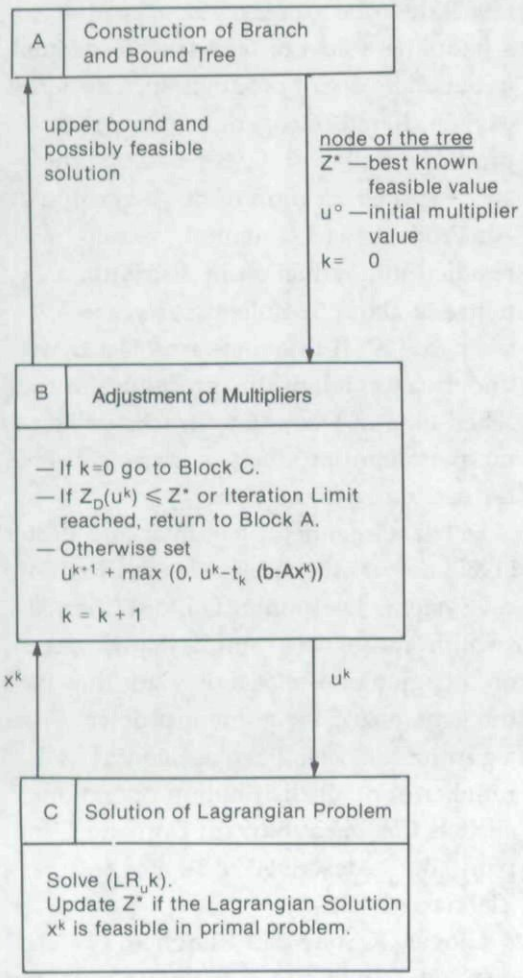

Figure 2: Generic Lagrangian relaxation algorithm.

art, much like formulation itself. Typically, one will construct several alternative relaxations and evaluate them, both empirically and analytically, using the result on quality of bounds presented in the previous section. The alternative relaxations can be constructed in one of two ways. One way is to begin with an integer programming formulation and select different constraints to dualize. Alternatively, one can begin with some easy-to-solve model such as the knapsack problem or shortest-route problem which is close to, but not exactly the same as, the problem one wishes to solve. Then try to add a set of side constraints to represent those aspects of the problem of interest which are missing in the simpler model. A Lagrangian relaxation can be obtained by dualizing the side constraints that have been added.

Summary of Concepts

Up to this point, I have developed the concept of Lagrangian relaxation "piecemeal" on an example. We can now formulate and present a generic Lagrangian relaxation algorithm.

Figure 2 shows a generic Lagrangian relaxation algorithm consisting of three major steps. The first step is the standard branch and bound process in which a tree of solution alternatives is constructed with certain variables fixed to specified values at each mode of the tree. These specified values are passed from block A to block B together with Z^*, the objective value of the currently best known feasible solution, and starting multipliers u^0.

In blocks B and C, we iterate between adjusting the multipliers with the subgradient update formula (6) to obtain a new multiplier value u^k and solving the Lagrangian problem to obtain a new Lagrangian solution x^k. This process continues until we either reach an iteration limit or discover an upper bound for this node that is less than or equal to the current best known feasible value Z*. At this point, we pass back to block A the best upper bound we have discovered together with any feasible solution that may have been obtained as a result of solving

the Lagrangian problem. In my experience, it is rare in practice that the Lagrangian solution will be feasible in the original problem (P). However, it is not uncommon that the Lagrangian solution will be nearly feasible and can be made feasible with some minor modifications. A systematic procedure for doing this can be applied in block C and constitutes what might be called a "Lagrangian heuristic." Lagrangian heuristics have been vital to the computational success of many applications, such as those described in Fisher [1981; 1982], and may well prove to be as important as the use of Lagrangians to obtain upper bounds.

It is not uncommon in large-scale applications to terminate the process depicted in Figure 2 before the branch and bound tree has been explored sufficiently to prove optimality. In this case the Lagrangian algorithm is really a heuristic with some nice properties, such as an upper bound on the amount by which the heuristic solution deviates from optimality.

Past Applications and Future Prospects

A brief description of several instances in which Lagrangian relaxation has been used in practice should give the flavor of the kinds of problems for which Lagrangian relaxation has been successful.

Bean [1984] is concerned with the problem of determining divestitures over time from a portfolio in order to maximize total return subject to a return on equity minimum in each period imposed by an outside force such as a parent organization. The algorithm he has developed has been applied in the land development industry.

Fisher, Jaikumar, Greenfield and Kedia [1982] describe a Lagrangian algorithm for scheduling a fleet of tank trucks engaged in bulk delivery of products such as liquid oxygen, liquid nitrogen, or petroleum products. Bell et al. [1983] describe the successful application of this algorithm at Air Products and Chemicals, which has resulted in a reduction in distribution expense of about $2 million per year. Shepard [1984] describes a project that is underway to adapt this procedure to the scheduling of Exxon tank trucks delivering petroleum products, such as gasoline, for use at service stations.

Fisher, Greenfield, Jaikumar and Lester [1982] discuss the application in the Clinical Systems Division of DuPont of an algorithm for vehicle routing that is based on a Lagrangian relaxation algorithm for the generalized assignment problem. This algorithm has also been applied to a number of other distribution operations, such as the one at Edward Don and Company that is described in Walter and Zielinski [1983].

Glover, Karney and Klingman [1979] deal with a manpower planning problem in which individuals must be assigned to jobs to make use of their skills and also be provided with adequate job satisfaction. The Lagrangian relaxation algorithm they develop has been applied to a very large instance of this problem faced by the US Navy.

Glover, Klingman and Ross [1984] address a large-scale product development and distribution problem. The results of their work have been used for over 20 months at a company interested in establishing a five-year monthly operating plar for introducing new products and deter-

mining product distribution to customers consistent with its standing contracts, technological limitations, and governmental regulations. Their algorithm was credited with providing insights and suggesting strategies that led to very large savings.

Graves and Whitney [1979] and Graves and Lamar [1983] treat the problem of designing an assembly system by choosing from available technology a group of resources to perform certain operations. The choices cover people, single purpose machines, narrow purpose pickplace robots, and general purpose robots. Their work has been applied in a number of industries, including the design of robot assembly systems for production of automobile alternators. Graves [1982] has also discussed the use of Lagrangian relaxation to address production planning problems from an hierarchical perspective.

Manero [1984] has applied Lagrangian relaxation to the check processing operations of a large New York bank. The specific decisions involved included routing of vehicles that were picking up checks at branch banks and scheduling personnel involved in check processing.

Mulvey [1980] is concerned with condensing a large data base by selecting a subset of "representative elements." He has developed a Lagrangian-relaxation-based clustering algorithm that determines a representing subset for which the loss in information is minimized in a well defined sense. He has used this algorithm to reduce the 1977 US Statistics of Income File for Individuals maintained by the Office of Tax Analysis from 155,212 records to 74,762 records.

The application described in Shepardson and Marsten [1980] involves the scheduling of personnel who must work two duty periods, a morning shift and an afternoon shift. Their algorithm determines optimal schedules for each worker so as to minimize cost and satisfy staffing requirements. Helsinki City Transport has applied this algorithm to bus crew scheduling.

Van Roy and Gelders [1981] discuss the use of Lagrangian relaxation for a particular problem arising in distribution.

In each of the applications described above, development of the Lagrangian relaxation algorithm required a level of involvement on part of skilled analysts that is similar to that required in the use of dynamic programming. Just as some insight into a problem is required before dynamic programming can be applied fruitfully, it is generally nontrivial to discover a Lagrangian relaxation that is computationally effective. Moreover, once this has been done, the various steps in the algorithm must be programmed more or less "from scratch." Often this process can be made easier by the availability of an "off the shelf" algorithm for the Lagrangian problem if it is a well-known model, such as a network flow, shortest route, minimum spanning tree, or knapsack problem.

Despite the level of effort required in implementing Lagrangian relaxation, the concept is growing in popularity because the ability it affords to exploit special problem structure often is the only hope for coping with large real problems. For the future, it remains to be seen whether

Lagrangian relaxation will continue to exist as a technique that requires a significant ad hoc development effort or whether the essential building blocks of Lagrangian relaxation will find their way into user-friendly mathematical programming codes such as LINDO or IFPS OPTIMUM. Such a development could provide software for carrying out steps A and B in the generic flowchart as well as a selection of algorithms for performing step C for the most popular easy-to-solve models. It would then be left to the analyst to decide which constraints to dualize and to specify which of the possible Lagrangian problem algorithms to use.

Acknowledgments

I'd like to express my appreciation to Gary Lilien for suggesting this project and for his encouragement and advice during its execution. I am indebted to the following individuals who read an early draft of the paper and suggested significant improvements: James Bean, Mike Carter, Ron Dembo, Donald Erlenkotter, Rob Garfinkel, Arthur Geoffrion, Fred Glover, Stephen Graves, Michael Held, M. J. Magazine, John Mulvey, Alexander Rinnooy Kan, Richard Rosenthal, Linus Schrage, and Jeremy Shapiro. The paper has also benefited from a careful reading by the referee, Terry Harrison, and was supported in part by NSF Grant ECS 82-05 438 to the University of Pennsylvania.

References

Bean, J. C. 1984, "Asset divestiture strategies with return on equity requirements," Department of Industrial and Operations Engineering Working Paper, University of Michigan.

Bell, W. J.; Dalberto, L. M.; Fisher, M. L.; Greenfield, A. J.; Jaikumar, R.; Kedia, P.; Mack, R. G.; Prutzman, P. J. 1983, "Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer," *Interfaces*, Vol. 13, No. 6 (December), pp. 4-23.

Erlenkotter, D. 1978, "A dual-based procedure for uncapacitated facility location," *Operations Research*, Vol. 26, No. 1, pp. 992-1009.

Fisher, M. L. 1981, "The Lagrangian relaxation method for solving integer programming problems," *Management Science*, Vol. 27, No. 1, pp. 1-18.

Fisher, M. L.; Greenfield, A. J.; Jaikumar, R.; and Lester, J. T., III 1982, "A computerized vehicle routing application," *Interfaces*, Vol. 12, No. 4 (August), pp. 42-52.

Fisher, M. L.; Greenfield, A. J.; Jaikumar, R.; and Kedia, P., 1982, "Real-time scheduling of a bulk delivery fleet: Practical application of Lagrangian relaxation," Decision Sciences Working Paper 82-10-11, The Wharton School, University of Pennsylvania.

Geoffrion, A. M. 1974, "Lagrangian relaxation and its uses in integer programming," *Mathematical Programming Study 2*, pp. 82-114.

Glover, F.; Karney, D.; and Klingman, D. 1979 "A study of alternative relaxation approaches for a manpower planning problem," *Quantitative Planning and Control*, Yuji Ijiri and Andrew B. Whinston, eds., Academic Press, New York, pp. 141-164.

Glover, F.; Klingman, D.; and Ross, G. J. 1984, "Integrating modeling, algorithm design and computational implementation to solve a large-scale non-linear mixed integer programming problem," Center for Busines Decision Analysis Report 105, University of Texas, Austin.

Graves, S. C. 1982, "Using Lagrangian techniques to solve hierarchical production planning problems," *Management Science*, Vol. 28, No. 3, pp. 260-275.

Graves, S. C. and Whitney, D. E. 1979, "A mathematical programming procedure for equipment selection and system evaluation in programmable assembly," in *Proceedings of the 18th IEEE Conference on Decision and Control*, Ft. Lauderdale, Florida, pp. 531-536.

Graves, S. C. and Lamar, B. W. 1983, "An integer programming procedure for assembly system design problems," *Operations Re-*

search, Vol. 31, No. 3 (May-June), pp. 522-545.

Held, M. H.; Wolfe, P. and Crowder, H. D. 1974, "Validation of subgradient optimization," *Mathematical Programming*, Vol. 6, No. 1, pp. 62-88.

Manero, R. 1984, "A model for the coordination of check collection and processing in a network of bank branches," PhD Dissertation, Columbia University Graduate School of Business.

Mulvey, J. 1980, "Reducing the US Treasury's taxpayer data base by optimization," *Interfaces*, Vol. 10, No. 5 (October), pp. 101-112.

Shapiro, J. F. 1979, *Mathematical Programming: Structures and Algorithms*, John Wiley, New York.

Shepard, S. W. 1984, "Computerized fleet planning for bulk petroleum products," Exxon Corporation CCS Department Paper, Florham Park, New Jersey.

Shepardson, F. and Marsten, R. E. 1980, "A Lagrangian relaxation algorithm for the two duty period scheduling problem," *Management Science*, Vol. 26, No. 3 (March), pp. 274-281.

Van Roy, T. J. and Gelders, L. 1981, "Solving a distribution problem with side constraints," *European Journal of Operational Research*, Vol. 6, No. 1 (January), pp. 61-66.

Walter, D. F. and Zielinski, E. 1983, "Computerized vehicle routing package — management issues," Talk delivered at the 1983 Annual Meeting of the National Council of Physical Distribution Management, New Orleans, Louisiana.