Chapter 9
Lagrangian Relaxation for Integer Programming

Arthur M. Geoffrion

Introduction by *Arthur M. Geoffrion*

It is a pleasure to write this commentary because it offers an opportunity to express my gratitude to several people who helped me in ways that turned out to be essential to the birth of [8]. They also had a good deal to do with shaping my early career and, consequently, much of what followed.

The immediate event that triggered my interest in this topic occurred early in 1971 in connection with a consulting project I was doing for Hunt-Wesson Foods (now part of ConAgra Foods) with my colleague Glenn Graves. It was a distribution system design problem: how many distribution centers should there be and where, how should plant outputs flow through the DCs to customers, and related questions. We had figured out how to solve this large-scale MILP problem optimally via Benders Decomposition, a method that had been known for about a decade but had not yet seen practical application to our knowledge. This involved repeatedly solving a large 0-1 integer linear programming master problem in alternation with as many pure classical transportation subproblems as there were commodity classes. The master problem was challenging, and one day Glenn, who did all the implementation, came up with a new way to calculate conditional "penalties" to help decide which variable to branch on in our LP-based branch-and-bound approach.

I regularly taught a doctoral course in those days that covered, *inter alia*, the main types of penalties used by branch-and-bound algorithms. But after studying the math that Glenn used to justify his, I didn't see a connection to any of the penalties I knew about. I did, however, notice that Glenn made use of a Lagrangean term, and I was very familiar with Lagrangeans owing to my earlier work on solving discrete optimization problems via Lagrange multipliers [2] and on duality theory in nonlinear programming [6]. It often happens that a mathematical result can be

Arthur M. Geoffrion
UCLA Anderson School of Management, Los Angeles, USA
e-mail: ageoffri@anderson.ucla.edu

M. Jünger et al. (eds.), *50 Years of Integer Programming 1958-2008*,
DOI 10.1007/978-3-540-68279-0_9,

derived in quite distinct ways, and so it was in this case: I found that not only Glenn's penalties, but several other kinds of penalties could be derived in a unified way as shown in Sec. 4 of [8], and that numerous special problem structures could be exploited to produce additional penalties. This pleased me greatly, because I had a passion for trying to unify and simplify results that others had derived from disparate viewpoints, especially in the context of exploiting special problem structure. At that point, I knew that I had to write this up.

Shortly it became clear that what I later dubbed *Lagrangean relaxation* was useful for exploiting various kinds of special structures of integer programming problems in other ways besides penalties. In particular, it can be used to tailor most of the main operations found in branch-and-bound algorithms as explained in Sec. 3 of [8]. It also rendered obsolete the need for so-called *surrogate constraints* as explained in Sec. 5, and it can be used to derive new cutting planes as explained in Sec. 6. Some basic theory of Lagrangean relaxation had to be filled in, the subject of Sec. 3, and this drew importantly on my earlier work on nonlinear duality. I had a working paper version of [8] by late 1971, and in late 1972 presented the main results at a symposium in Germany. When Glenn and I wrote up the work surrounding the Hunt-Wesson Foods project, we included a comment in Sec. 3.1 of [7] on the branching penalties used in our implementation.

To explain more fully where [8] came from, I should also explain how the triggering Hunt-Wesson project came about, especially since this was my first industrial consulting engagement since obtaining my Ph.D. 5 years earlier (how does one boot a consulting practice?), and I should comment on the prior research that supported [8] and the novel solution method used for the Hunt-Wesson problem. First a few words about the origin of the project.

A very senior UCLA colleague of mine, Professor Elwood Buffa, opened a door in 1970 that would change my life in unforeseen ways. A former doctoral student of his, Dr. William Taubert, was then a vice president of Hunt-Wesson Foods, which had been struggling for years to rationalize its network of distribution centers. El knew that I was working on large-scale optimization methods that might conceivably apply to such problems, but he couldn't have known whether I could adapt those methods successfully. Neither did I. With no prompting whatever, he decided to recommend me to Bill Taubert as a consultant. El didn't have to take that risk, nor did Bill in hiring me. If I failed—which my inexperience as a consultant and unfamiliarity with distribution systems should have made the safest bet—it would have been an embarrassment to El, Bill, and UCLA.

But a streak of good luck ensued, leading to a successful project at Hunt-Wesson Foods, to many more consulting engagements in what is now called supply chain management, to the founding of a consulting and software firm that celebrates its 30th anniversary this year (2008), to the discovery of several important research problems that would occupy most of the rest of my career, and to an appreciation for the synergies of research, practice, and teaching that has shaped my professional life, including my service to TIMS and INFORMS.

If fortune favors the prepared mind, mine must have been prepared by my previous work on topics that proved useful not only for the Hunt-Wesson Foods project

and what followed from it, but also for the paper which this commentary introduces. Especially my work on integer programming (especially [3, 4]), nonlinear duality theory [6], and large-scale optimization methods (especially [5]). Most of that work came about because of another door opened for me by my dissertation advisor at Stanford University, Professor Harvey Wagner.

When I accepted a job at UCLA's business school in 1964, just prior to finishing my thesis, Harvey suggested that I would benefit from being a day-a-week consultant at RAND Corporation, just a few miles from UCLA. He arranged it with Dr. Murray Geisler, head of RAND's Logistics Department. At that time, RAND was not far past its prime as the greatest think tank in the world, including its astonishing role as the fertile spawning ground or incubator of such important OR methods as discrete event and Monte Carlo simulation, dynamic programming, game theory, parts of inventory and logistics theory, network flow theory, and mathematical programming—linear, quadratic, stochastic, and integer. RAND was also a major contributor to the very early history of artificial intelligence, digital computing, the Internet, both systems analysis and policy analysis, the U.S. space program, and much more besides. That day a week, which lasted fairly steadily until the early 1970s, was disproportionately important to my early research life.

I had fine operations research colleagues at UCLA, but none did research in optimization, whereas at RAND I could interact with many staff members and A-list consultants who did, including Robin Brooks, Eric Denardo, Ben Fox, Ray Fulkerson, Glenn Graves, Al Madansky, Harry Markowitz, Bob Thrall, and Philip Wolfe. Moreover, at RAND I had excellent computer programming and clerical/data services (they had an IBM 7044 when I arrived), a full-service publication department that professionally edited and widely disseminated most of my research papers on optimization, and a good library that would even translate Russian-language articles at my request. I was in heaven there, and could not overstate the advantages gained from RAND's infrastructure and my second set of colleagues there as I launched my career.

It was at RAND that, very early in 1965, Murray handed me a somewhat beat up copy of Egon Balas' additive algorithm paper prior to its publication [1] (written while Egon was still in Rumania), and asked me to take a look at it since it was creating a stir. Thus commenced my enduring interest in integer programming. I recast this work as LP-based implicit enumeration in a limited-circulation manuscript dated August 23, 1965, published internally at RAND in September 1967 and externally about two years later [4]. Murray quickly arranged for Richard Clasen—an important early figure in mathematical programming in his own right—to be assigned to me to implement my first 0-1 integer programming code, the RIP30C incarnation of which RAND distributed externally starting mid-1968. Murray also arranged for others to assist me with the extensive numerical experiments.

My debt to RAND goes beyond even what is mentioned above: as a hotbed of OR for many years, RAND's influence on nearby UCLA for more than a decade prior to my arrival helped to build and shape an OR group with a vitality and local culture that provided a comfortable home for my entire career. The group's founding in the early 1950s originated independently of RAND, but its frequent interac-

tions with RAND staff and consultants in its early days were of incalculable value; there are records of visits in the 1950s by Kenneth Arrow, Richard Bellman, Abe Charnes, Bill Cooper, George Dantzig, Merrill Flood, Ray Fulkerson, Alan Manne, Harry Markowitz, Oscar Morgenstern, Lloyd Shapley, Andy Vaszonyi, and dozens of others. (As an aside, the phrase "management sciences" was coined during a conversation between Melvin Salveson, the former Tjalling Koopmans student who founded UCLA's OR group, and Merrill Flood of RAND in September, 1953, the same month when Mel hosted on campus the first pre-founding meeting—attended by many RAND OR people—of what became The Institute of Management Sciences (TIMS) three months later.) Some taught courses as lecturers, and some even joined the faculty. By the time of my arrival, these interactions had largely tailed off, but they left a palpable tradition of creativity and excellence in my group that inspired my best efforts as an impressionable young faculty member.

Let me summarize. The paper following this commentary did not appear out of nowhere. It was enabled by multiple gifts of wisdom and kindness toward me by Harvey Wagner, who taught me how to do research and arranged for me to consult at RAND; by Elwood Buffa, who dropped my first and all-important consulting job in my lap; by Murray Geisler, who turned my attention to integer programming and arranged generous assistance in support of my research; and by my early colleague/mentors at UCLA, Jim Jackson (an OR pioneer whose contributions included "Jackson networks") and Jacob Marschak (a world-class economist), who helped shape my understanding of what it means to be a professor, arranged for me to be supported from the outset on their research grants, and then helped me obtain my own grants (from NSF starting in 1970 and ONR starting in 1972). I will always be grateful to these people for the important roles they played in my professional life.

References

1. E. Balas, *An additive algorithm for solving linear programs with zero-one variables*, Operations Research 13 (1965) 517–546.
2. R. Brooks and A.M. Geoffrion, *Finding Everett's Lagrange multipliers by linear programming*, Operations Research 14 (1966) 1149–1153.
3. A.M. Geoffrion, *Integer programming by implicit enumeration and Balas' method*, Review of the Society for Industrial and Applied Mathematics 9 (1967) 178–190.
4. A.M. Geoffrion, *An improved implicit enumeration approach for integer programming*, Operations Research 17 (1969) 437–454.
5. A.M. Geoffrion, *Elements of large-scale mathematical programming*, Management Science 16 (1970) 652–691.
6. A.M. Geoffrion, *Duality in nonlinear programming*, SIAM Review 13 (1971) 1–37.
7. A.M. Geoffrion and G.W. Graves, *Multicommodity distribution system design by Benders decomposition*, Management Science 20 (1974) 822–844.
8. A.M. Geoffrion, *Lagrangean relaxation for integer programming*, Mathematical Programming Study 2 (1974) 82–114.

The following article originally appeared as:

A.M. Geoffrion, *Lagrangian Relaxation for Integer Programming*, Mathematical Programming Study 2 (1974) 82–114.

Mathematical Programming Study 2 (1974) 82–114. North-Holland Publishing Company

LAGRANGEAN RELAXATION FOR INTEGER PROGRAMMING *

A.M. GEOFFRION

University of California, Los Angeles, Calif., U.S.A.

Received 25 January 1974
Revised manuscript received 26 August 1974

Taking a set of "complicating" constraints of a general mixed integer program up into the objective function in a Lagrangean fashion (with fixed multipliers) yields a "Lagrangean relaxation" of the original program. This paper gives a systematic development of this simple bounding construct as a means of exploiting special problem structure. A general theory is developed and special emphasis is given to the application of Lagrangean relaxation in the context of LP-based branch-and-bound.

1. Introduction

The general integer linear programming problem can be written as

$$\text{(P)} \qquad \underset{x \geqq 0}{\text{minimize}} \; c\,x,$$

$$\text{subject to } A\,x \geqq b, \qquad B\,x \geqq d,$$
$$x_j \text{ integer}, \qquad j \in I,$$

where b, c and d are vectors, A and B are matrices of conformable dimensions, and the index set I denotes the variables required to be integer. The reason for distinguishing two types of constraints is that the second of these, $B\,x \geqq d$, is supposed to have special structure.

We define the *Lagrangean relaxation* of (P) relative to $A\,x \geqq b$ and a conformable nonnegative vector λ to be

$$(\text{PR}_\lambda) \qquad \underset{x \geqq 0}{\text{minimize}} \; c\,x + \lambda(b - A\,x),$$

$$\text{subject to } B\,x \geqq d,$$
$$x_j \text{ integer}, \qquad j \in I.$$

* An earlier version of this paper was presented at the IBM International Symposium on Discrete Optimization, Wildbad, Germany, October 30–November 1, 1972. This research was supported by the Office of Naval Research under Contract Number N00014-69-A-0200-4042 and by the National Science Foundation under Grants GP-26294 and GP-36090X.

The fruitful application of (PR_λ) in specific cases requires judicious partitioning of the constraints into the two types $A x \geqq b$ and $B x \geqq d$, and an appropriate choice of $\lambda \geqq 0$.

Lagrangean relaxation has been used by Held and Karp [20,21] in their highly successful work on the traveling-salesman problem; by Fisher [6] in his promising algorithm for scheduling in the presence of resource constraints and in his efficient machine scheduling algorithm [7]; by Fisher and Schrage [9] in their proposed algorithm for scheduling hospital admissions; by Ross and Soland [26] in their remarkably efficient algorithm for the generalized assignment problem; and by Shapiro [27] and Fisher and Shapiro [10] in the context of a group theoretic approach to pure integer programming. See also [8]. Other authors have also made special application of Lagrangean relaxation ideas implicitly if not explicitly in their work. Not to be forgotten is the general relevance of the literature on Lagrangean methods for nonconvex optimization (e.g., [2, 5, 18]).

The purpose of this paper is to develop the theory and explore the usefulness of Lagrangean relaxation in the context of branch-and-bound or implicit enumeration methods for (P). In contrast with most of the references just cited, our emphasis is on LP-based branch-and-bound algorithms rather than those whose bounding constructs do not involve linear programming. This is not to deny the great value of non-LP-based techniques for special problems, but rather to stress the as yet untapped potential of Lagrangean relaxation as a means of making the most widely used general purpose approach more efficient for problems with special structure. The development is intended for use at two levels. Pedagogically it strives for a unified exposition of a number of old and new developments in integer programming. As a research effort it aims to develop what appears to be a potent general approach to the design of improved algorithms for special classes of integer programs. Although the algorithmic context of this paper is the branch-and-bound approach to integer linear programs, it is clear that these ideas can also be applied to other classes of algorithms and problems.

The paper is organized as follows. The basic results concerning the relation between (P), (PR_λ) and related problems are collected in Section 2. Lagrangean duality theory turns out to play a surprisingly major role. In Section 3, a generic LP-based branch-and-bound approach for (P) is reviewed, and the basic uses and strategies of Lagrangean relaxation in this context are described. Section 4 derives the standard penalties of Driebeek [4] and Tomlin [28] from the viewpoint of Lagrangean relaxation, and

several new penalties are developed. In Section 5, the concept of surrogate constraints as developed by Glover [17] and the author [12] is shown to be subsumed by the Lagrangean relaxation viewpoint. Section 6 derives cutting-planes based on Lagrangean relaxation, including some which utilize the penalties of Section 4. Concluding comments are given in Section 7.

Three simple examples for the special constraints $Bx \geqq d$ will now be introduced. They will serve in the sequel to illustrate general ideas and to emphasize that Lagrangean relaxation is intended to be specialized to particular problem structures. The final subsection of this Introduction summarizes the special notations and assumptions commonly used in the sequel.

1.1. *Three examples*

The Lagrangean relaxation (PR_λ) must be much simpler to solve than (P) itself in order for it to yield any computational advantage. It should admit a closed form solution or be solvable by an efficient specialized algorithm. Thus the constraints $Bx \geqq d$ must possess considerable special structure. Three of the simplest possible examples of such structure are as follows. They will be referred to repeatedly in the sequel.

Example 1. The constraints $Bx \geqq d$ specify only upper bounds on some or all of the variables. For instance, in 0–1 programming problems the integer variables possess upper bounds of unity. It is easy to see that the optimal solution of (PR_λ) can be written down by inspection of the signs of the collected coefficient vector of x, namely $(c - \lambda A)$.

Example 2. The constraints $Bx \geqq d$ are as in Example 1 but also include some generalized upper bounding constraints of the form

$$\sum_{j \in J_k} x_j = 1, \qquad k = 1, 2, \ldots, K, \tag{1}$$

where $J_1, \ldots, J_K$ are disjoint subsets of I. Such constraints perform a "multiple choice" function. The optimal solution of (PR_λ) can again be written down by inspection, with a search for the smallest $(c - \lambda A)_j$ now being necessary over each subset J_k.

Example 3. The constraints $Bx \geqq d$ are as in Example 1 but also include some constraints of the form

$$\sum_{j \in J_k} \beta_{kj} x_j \leqq \beta_{kk} x_k, \qquad k = 1, \ldots, K, \tag{2}$$

where the K subsets $\{k, J_k\}$ are disjoint, $x_1, \ldots, x_K$ are 0–1 variables, the variables in J_k are continuous-valued, and all β coefficients are strictly positive. This type of constraint typically arises in location and expansion models. In the familiar capacitated plant location problem for example, x_k is 1 or 0, according to whether or not a plant of capacity β_{kk} is built at the k^{th} site, x_j for $j \in J_k$ corresponds to the amounts shipped from plant site k to various destinations, and the β_{kj}'s are all unity. The Lagrangean relaxation (PR_λ) can be solved easily because it separates into K independent problems of the form

$$\begin{aligned} &\text{minimize} \sum_{j \in J_k} (c - \lambda A)_j x_j + (c - \lambda A)_k x_k, \\ &\text{subject to} \sum_{j \in J_k} \beta_{kj} x_j \leqq \beta_{kk} x_k, \\ &\qquad 0 \leqq x_j \leqq u_j, \qquad j \in J_k, \\ &\qquad x_k = 0 \text{ or } 1, \end{aligned} \tag{3^k_λ}$$

where u_j is the upper bound on variable x_j. If $x_k = 0$, it follows from the positivity of β_{kj} that $x_j = 0$ must hold for all $j \in J_k$. If $x_k = 1$, (3^k_λ) becomes a trivial "continuous knapsack problem" with bounded variables. The best of the solutions obtained under the two cases $x_k = 0$ and $x_k = 1$ yields the true optimal solution of (3^k_λ). From these K solutions one may directly assemble the optimal solution of (PR_λ).

These three examples are among the simplest types of special constraints $B\,x \geqq d$ for which the associated Lagrangean relaxation can be optimized very efficiently. Whereas closed form solutions are available for these examples, other applications may call for specialized algorithms of a less trivial sort. In most practical applications of integer programming there are several obvious and tractable choices for the constraints to be designated as $B\,x \geqq d$. In Held and Karp's excellent work on the traveling-salesman problem [20, 21], for example, (PR_λ) is a minimum spanning "1-tree" problem for which highly efficient algorithms are available. And in Fisher and Schrage's algorithm for hospital admissions scheduling [9], (PR_λ) separates into a relatively simple scheduling problem for each patient.

1.2. *Notation and assumptions*

Notation and terminology is generally standard and consistent with that of [14], a survey paper containing additional background material. However, *the reader should memorize the following peculiar notations:* if $(\cdot)$ is an optimization problem, then $v(\cdot)$ is its optimal value, $F(\cdot)$ is its set of feasible solutions, and $(\bar{\ })$ refers to the same problem with all integrality conditions on the variables dropped; the vector $\bar{\lambda}$ denotes an optimal multiplier vector (dual solution) associated with the constraints $Ax \geqq b$ for the ordinary linear program $(\bar{\mathrm{P}})$.

We adopt the convention that the optimal value of an infeasible optimization problem is $+\infty$ (resp. $-\infty$) if it is a minimizing (resp. maximizing) problem. The inner product of two vectors, be they row or column, is denoted simply by their juxtaposition.

Two benign assumptions are made throughout this paper in the interest of decluttering the exposition, except where explicitly stated to the contrary. The first is that the nonspecial constraints $Ax \geqq b$ are all inequality constraints. If some of these constraints were given as *equalities*, then the corresponding components of λ would not be required to be nonnegative. This is the only change required to accommodate equality constraints. The second assumption is that the special constraints $Bx \geqq d$ include upper bounds on all variables. This obviates the need for special treatment of the case where (P) or one of its relaxations has optimal value equal to $-\infty$, and also permits certain notational economies. This assumption is consistent with the vast majority of potential applications. It is a simple exercise to allow for its absence in all of the results to follow.

2. Theory of Lagrangean relaxation

The term *relaxation* is used in this paper in the following sense: a minimizing problem (Q) is said to be a relaxation of a minimizing problem (P) if $F(\mathrm{Q}) \supseteq F(\mathrm{P})$ and the objective function of (Q) is less than or equal to that of (P) on $F(\mathrm{P})$. Clearly (PR_λ) is a relaxation in this sense for all $\lambda \geqq 0$, for the extra Lagrangean term $\lambda(b - Ax)$ in the objective function of (PR_λ) must be nonpositive when $Ax \geqq b$ is satisfied. Notice that the common practice of relaxation by simply throwing away some of the constraints is equivalent to Lagrangean relaxation with $\lambda = 0$. Permitting $\lambda \neq 0$ (but always $\geqq 0$) allows the relaxation to be tighter.

The potential usefulness of any relaxation of (P), and of a Lagrangean relaxation in particular, is largely determined by how near its optimal

value is to that of (P). This furnishes a criterion by which to measure the "quality" of a particular choice for λ. The ideal choice would be to take λ as an optimal solution of the (concave) program

$$\text{(D)} \qquad \underset{\lambda \geqq 0}{\text{maximize}}\ v(\text{PR}_\lambda),$$

which is designated by (D) because it coincides with the formal Lagrangean dual of (P) with respect to the constraints $A\,x \geqq b$ (see, e.g., [13]). This problem in turn is intimately linked to the following relaxation of (P):

$$\text{(P*)} \qquad \underset{x}{\text{minimize}}\ c\,x,$$

$$\text{subject to } A\,x \geqq b,$$

$$x \in \text{Co}\,\{x \geqq 0\colon B\,x \geqq d \text{ and } x_j \text{ integer}, j \in I\},$$

where Co denotes the convex hull of a set. It may be difficult to express the convex hull in (P*) as an explicit set of linear constraints, but in principle this is always possible and so (P*) may be regarded as a linear program. In fact, as we shall see, (P*) and (D) are essentially LP duals. An optimal multiplier vector corresponding to $A\,x \geqq b$ will be denoted by λ^* when (P*) has finite optimal value.

Theorem 1 describes some of the basic relationships between (P), (PR_λ), (D), (P*), and ($\bar{\text{P}}$) (the usual LP relaxation which drops the integrality requirements).

Theorem 1. (a)

$$\text{(a)} \qquad F(\bar{\text{P}}) \supseteq F(\text{P*}) \supseteq F(\text{P}), \qquad F(\text{PR}_\lambda) \supseteq F(\text{P}),$$

$$v(\bar{\text{P}}) \leqq v(\text{P*}) \leqq v(\text{P}), \qquad v(\text{PR}_\lambda) \leqq v(\text{P}) \quad \textit{for all } \lambda \geqq 0.$$

(b) *If* ($\bar{\text{P}}$) *is feasible, then* $v(\bar{\text{P}}) \leqq v(\text{PR}_{\bar{\lambda}})$.

(c) *If for a given* λ *a vector* x *satisfies the three conditions*

(i) x *is optimal in* (PR_λ),
(ii) $A\,x \geqq b$,
(iii) $\lambda\,(b - A\,x) = 0$,

then x *is an optimal solution of* (P). *If* x *satisfies* (i) *and* (ii) *but not* (iii), *then* x *is an* ε*-optimal solution of* (P) *with* $\varepsilon = \lambda\,(A\,x - b)$.

(d) *If* (P*) *is feasible, then*

$$v(\text{D}) \equiv \max_{\lambda \geqq 0} v(\text{PR}_\lambda) = v(\text{PR}_{\lambda^*}) = v(\text{P*}).$$

Proof. Parts (a) and (c) are very easy. Let $(\overline{\mathrm{P}})$ be feasible. Then it has finite optimal value, for $Bx \geqq d$ includes upper bounds on all variables, and

$$\begin{aligned} v(\overline{\mathrm{P}}) &= \max_{\lambda \geqq 0} v(\overline{\mathrm{PR}_\lambda}) && \text{(by the dual theorem of linear programming)}^1, \\ &= v(\overline{\mathrm{PR}_{\bar{\lambda}}}) && \text{(by the definition of } \bar{\lambda}), \\ &\leqq v(\mathrm{PR}_{\bar{\lambda}}) && \text{(because } F(\overline{\mathrm{PR}_{\bar{\lambda}}}) \supseteq F(\mathrm{PR}_{\bar{\lambda}}). \end{aligned}$$

This proves part (b). An identical argument (the third portion is not needed) applied to (P*) yields the conclusion of part (d) if one uses the following observation in the obvious way:

$$v(\mathrm{PR}_\lambda) = [\min_x c x + \lambda(b - A x),$$
$$\text{s.t.} \quad x \in \mathrm{Co}\{x \geqq 0 \colon B x \geqq d \text{ and } x_j \text{ integer}, j \in I\}],$$

which holds because the minimum value of a linear function over any compact set is not changed if the set is replaced by its convex hull.

A few comments are in order. Part (a) simply records the most obvious relations between (P) and its relaxations $(\overline{\mathrm{P}})$, (P*) and (PR_λ). Part (b) shows that $\bar{\lambda}$, an immediate by-product of optimally solving the standard LP relaxation $(\overline{\mathrm{P}})$, yields a Lagrangean relaxation that is at least as good as $(\overline{\mathrm{P}})$ itself (hopefully it will be better). Part (c) indicates the well-known conditions under which a solution of a Lagrangean relaxation is also optimal or near-optimal in (P). This is in recognition of the fact that Lagrangean relaxation is of interest not only for the lower bounds it yields on v(P), but also for the possibility that it may actually yield an optimal or near-optimal solution of (P). It follows, incidentally, that (PR_λ) can yield in this manner a proven ε-optimal solution ($\varepsilon \geqq 0$) of (P) only if $v(\mathrm{PR}_\lambda) \geqq v(\mathrm{P}) - \varepsilon$. Thus the provable quality of the feasible solutions obtainable from Lagrangean relaxation by invoking part (c) is limited by the gap (if any) between v(P) and v(D). Part (d) establishes that Lagrangean relaxation can do as well as, but no better than (P*). Thus, the position of v(P*) in the interval $[v(\overline{\mathrm{P}}), v(\mathrm{P})]$ is the question of central concern when analyzing the potential value of Lagrangean relaxation applied to a particular problem class.

[1] We have taken here the "partial" dual of $(\overline{\mathrm{P}})$ with respect to the constraints $Ax \geqq b$, rather than the "full" dual customarily used in linear programming. See [13] (especially Sec. 6.1) for an account of this generalization of the traditional duality theory. It is easily verified that $\bar{\lambda}$ is a bona fide optimal solution of the partial dual even though it may be defined in terms of the full dual.

It bears emphasis that the conclusion of part (d) is really a simple consequence of the fact that (P*) and (D) are essentially LP duals of one another. The true dual of (P*) when multipliers are introduced just for the $A x \geqq b$ constraints is

$$\begin{aligned} \underset{\lambda \geqq 0}{\text{maximize}} \; [\min_{x} \; & c x + \lambda (b - A x), \\ & \text{s.t.} \quad x \in \text{Co}\,\{x \geqq 0 \colon B x \geqq d \text{ and } x_j \text{ integer}, j \in I\}]. \end{aligned}$$

But, as observed in the proof of part (d), the maximand of this problem equals $v(\text{PR}_\lambda)$, so (D) *must have the same optimal value and optimal solution set as the true dual of* (P*). Thus one may invoke most of the rich optimality/duality theory for linear programming to say much more about the relationship between (P*) and (D) than is said in Theorem 1 (d). For instance, one may assert when (P*) is feasible that the set of its optimal multipliers coincides with the set of optimal solutions of (D) and also with the negative of the set of subgradients at $y = 0$ of its (convex) b-perturbation function

$$\begin{aligned} \phi_b^*(y) \overset{\text{d}}{=} [\text{inf.} \; & c x, \\ \text{s.t.} \quad & A x \geqq b - y, \\ & x \in \text{Co}\,\{x \geqq 0 \colon B x \geqq d \text{ and } x_j \text{ integer}, j \in I\}], \end{aligned}$$

(see, e.g., [13, Th. 1 and 3]).

Theorem 1 leaves open two important questions: the relations between $v(\overline{\text{P}})$ and $v(\text{P}^*)$ and between $v(\text{P}^*)$ and $v(\text{P})$. These relations are taken up in the next two theorems.

Theorem 2 shows that $v(\overline{\text{P}}) = v(\text{P}^*)$ when the following holds:

Integrality Property. The optimal value of (PR_λ) is not altered by dropping the integrality conditions on its variables, i.e., $v(\text{PR}_\lambda) = v(\overline{\text{PR}_\lambda})$ for all $\lambda \geqq 0$.

Theorem 2. *Let* $(\overline{\text{P}})$ *be feasible and* (PR_λ) *have the Integrality Property. Then* (P*) *is feasible and*

$$v(\overline{\text{P}}) = v(\text{PR}_{\bar{\lambda}}) = v(\text{D}) = v(\text{PR}_{\lambda^*}) = v(\text{P}^*).$$

Proof. In view of Theorem 1 (b), (d), it is enough to show that $v(\overline{\text{P}}) = v(\text{P}^*)$. We have

$$\begin{aligned} v(\overline{\text{P}}) &= \max_{\lambda \geqq 0} v(\overline{\text{PR}_\lambda}) \quad \text{(by duality)}, \\ &= \max_{\lambda \geqq 0} v(\text{PR}_\lambda) \quad \text{(by the Integrality Property)}, \end{aligned}$$

$$= \max_{\lambda \geqq 0} \left[\min_{x} c\,x + \lambda\,(b - A\,x), \right.$$

$$\text{s.t.} \quad x \in \text{Co}\,\{x \geqq 0 \colon B\,x \geqq d \text{ and } x_j \text{ integer}, j \in I\}\Big]$$

(by the observation used in the proof of Theorem 1 (d)),

$$= v(\text{P}^*) \qquad \text{(by duality)}.$$

Notice that the feasibility of (P*) is a consequence of the fact that its dual has finite optimal value.

Thus Lagrangean relaxation can do no better than the standard LP relaxation ($\overline{\text{P}}$) when the constraint partition $A\,x \geqq b$, $B\,x \geqq d$ is such that the Integrality Property holds.[2] The best choice of λ for (PR_λ) is then $\bar{\lambda}$ from ($\overline{\text{P}}$). In this circumstance, Lagragean relaxation seems to be of questionable value unless a near-optimal solution of (D) can be found by specialized means more rapidly than ($\overline{\text{P}}$) can be solved by linear programming methods. Generally it is more promising to use Lagrangean relaxations for which the Integrality Property does not hold.

The Integrality Property clearly holds for Examples 1 and 2 (one may assume without loss of generality that the upper bounds on the integer variables are integers), but it does not hold for Example 3. The presence or absence of the Integrality Property is evident in many applications upon inspection of (PR_λ) in light of the special structure of the constraints $B\,x \geqq d$. In other applications one may be able to appeal to the total unimodularity characterization of natural integer solutions of linear programming problems (e.g., [30]).

We now turn to the relationship between $v(\text{P}^*)$ [or $v(\text{D})$] and $v(\text{P})$. A sufficient condition for $v(\text{P}^*) = v(\text{P})$ obviously is

$$F(\text{P}^*) = \text{Co}\,[F(\text{P})],$$

but this is likely to be difficult to verify in specific cases because the "integer polyhedron" is a notoriously difficult object to study.

Most of what is known about the relationship in question is a consequence of the fact that (D) is the formal Lagrangean dual of (P) with respect to the constraints $A\,x \geqq b$. Careful examination of Lagrangean duality theory shows that many of the results do not require convexity of the primal

[2] This fact has been noted by Nemhauser and Ullman [25] in the special context of Example 1.

problem. For instance, convexity is not used in the proofs of the key Lemmas 3, 4 and 5 of [13]. These results yield Theorem 3, which uses the following definitions. The *b-perturbation function* associated with (P) is defined as

$$\phi_b(y) \triangleq [\inf_{x \geqq 0} c\,x, \quad \text{s.t. } A\,x \geqq b - y, \quad B\,x \geqq d, \quad x_j \text{ integer}, \quad j \in I].$$

A vector γ conformable with y is said to be a *global subgradient* of ϕ_b at $y = 0$ (assuming $\phi_b(0) \equiv v(\mathrm{P})$ is finite) if

$$\phi_b(y) \geqq v(\mathrm{P}) + \gamma\,y \quad \text{for all } y.$$

The adjective "global" is used to emphasize that the subgradient definition used here relates to a global rather than local aspect of ϕ_b (which is generally nonconvex).

Theorem 3. *Assume that* (P) *is feasible* (*and therefore has an optimal solution, since all variables are bounded*).

(a) *The following are equivalent:*

(1) $v(\mathrm{P}) = v(\mathrm{D})$.

(2) *There exists a global subgradient of* ϕ_b *at* $y = 0$.

(3) *There exists a pair* (x, λ) *satisfying* $\lambda \geqq 0$ *and conditions* (i), (ii) *and* (iii) *of Theorem* 1(c).

(b) *If* $v(\mathrm{P}) = v(\mathrm{D})$, *then each optimal solution of* (D) *is the negative of a global subgradient of* ϕ_b *at* $y = 0$ *and conversely, and any such solution* λ^* *yields the set of all optimal solutions of* (P) *as the vectors* x *which satisfy conditions* (i), (ii) *and* (iii) *of Theorem* 1 (c) *with* $\lambda = \lambda^*$.

The most interesting aspect of Theorem 3 is the criterion for the equality $v(\mathrm{P}) = v(\mathrm{D})$ in terms of the existence of a global subgradient of ϕ_b at the origin and the identification of these subgradients with the solutions of (D). The theorem also confirms that Lagrangean relaxation does indeed yield the optimal solutions of (P) when $v(\mathrm{P}) = v(\mathrm{D})$, via the optimality conditions of Theorem 1 (c).

The b-perturbation function ϕ_b thus emerges as a key object for study if one wishes to understand when $v(\mathrm{P}) = v(\mathrm{D})$ is likely to hold. What is known about ϕ_b? Clearly it is nonincreasing. It can also be shown to be lower semicontinuous. It is piecewise-linear and convex over any region

where the optimal values of the integer variables stay constant, for in such a region the perturbed (P) reduces to a perturbed linear program. And ϕ_b is obviously bounded below by the piecewise-linear convex perturbation function ϕ_b^* defined earlier for (P*). In fact, this last observation can be strengthened to assert that ϕ_b^* is actually the *best* possible convex function which nowhere exceeds ϕ_b in value. This geometrically obvious but important result is stated more precisely as follows.

Theorem 4. *The b-perturbation function ϕ_b^* associated with* (P*) *is precisely the lower convex envelope of the b-perturbation function ϕ_b associated with* (P).

Proof. An alternative way of phrasing the result is to say that the epigraph of ϕ_b^* is the convex hull of the epigraph of ϕ_b; that is, $\text{Epi}[\phi_b^*] = \text{Co}\{\text{Epi}[\phi_b]\}$. Clearly,

$$\begin{aligned}\text{Epi}[\phi_b] &\triangleq \{(\mu, y): \mu \geqq \phi_b(y)\} \\ &= \{(\mu, y): \mu \geqq c\,x \text{ and } b - A\,x \leqq y \text{ for some } x \in X\},\end{aligned}$$

where

$$X \triangleq \{x \geqq 0: B\,x \geqq d \text{ and } x_j \text{ integer for } j \in I\},$$

and similarly for $\text{Epi}[\phi_b^*]$ with X replaced by $\text{Co}\{X\}$. Suppose that $(\bar{\mu}, \bar{y}) \in \text{Epi}[\phi_b^*]$. Then $\bar{\mu} \geqq c\,\bar{x}$ and $b - A\,\bar{x} \leqq \bar{y}$ for some $\bar{x} \in \text{Co}\{X\}$. Let $\bar{x} = \sum_h \theta_h x^h$, where $x^h \in X$, $\theta_h \geqq 0$ for all h and $\sum_h \theta_h = 1$. Define $\mu^h = c\,x^h$ and $y^h = b - A\,x^h$ for all h. Clearly $(\mu^h, y^h) \in \text{Epi}[\phi_b]$ for all h. Hence $\sum_h \theta_h(\mu^h, y^h) \in \text{Co}\{\text{Epi}[\phi_b]\}$. But

$$\begin{aligned}\sum_h \theta_h(\mu^h, y^h) &= (\textstyle\sum_h \theta_h \mu^h, \sum_h \theta_h y^h) \\ &= (\textstyle\sum_h \theta_h c\,x^h, \sum_h \theta_h(b - A\,x^h)) = (c\,\bar{x}, b - A\,\bar{x}) \leqq (\bar{\mu}, \bar{y})\end{aligned}$$

and so $(\bar{\mu}, \bar{y})$ must also be in $\text{Co}\{\text{Epi}[\phi_b]\}$. This shows that $\text{Epi}[\phi_b^*] \subseteq \text{Co}\{\text{Epi}[\phi_b]\}$. Now suppose $(\bar{\mu}, \bar{y}) \in \text{Co}\{\text{Epi}[\phi_b]\}$. Let $(\bar{\mu}, \bar{y}) = \sum_h \theta_h (\mu^h, y^h)$, where $(\mu^h, y^h) \in \text{Epi}[\phi_b]$, $\theta_h \geqq 0$ for all h and $\sum_h \theta_h = 1$. Let x^h be any point in X satisfying $\mu^h \geqq c\,x^h$ and $b - A\,x^h \leqq y^h$. Then

$$\sum_h \theta_h \mu^h \geqq \sum_h \theta_h c\,x^h = c\,\bar{x},$$

$$\sum_h \theta_h y^h \geqq \sum_h \theta_h(b - A\,x^h) = b - A\,\bar{x},$$

where $\bar{x} \stackrel{d}{=} \sum_h \theta_h x^h$. Thus $(\bar{\mu}, \bar{\nu}) \geqq (c\,\bar{x}, b - A\,\bar{x})$ with $\bar{x} \in \text{Co}\,\{X\}$, which shows that $(\bar{\mu}, \bar{\nu}) \in \text{Epi}\,[\phi_b^*]$. This completes the proof.

This is the central connection between (P) and (P*) – actually, between two parameterized families of problems of which (P) and (P*) are members of special significance. The duality gap (if any), $v(P) - v(D)$, is precisely equal to the difference between the b-perturbation function of (P) and its lower convex envelope, both evaluated at the origin. This characterization provides the basis for a qualitative understanding of duality gaps—and hence of the potential of Lagrangean relaxation—when applied to specific classes of problems with reference to salient characteristics of the data.

Some of these ideas are illustrated in Fig. 1 for a hypothetical mixed integer program with but a single A-type constraint (so that y is a scalar). Suppose that only two sets of values for the integer variables enter into an optimal solution of (P) as b varies. The piecewise-linear and convex b-perturbation functions for the two corresponding linear programs (with the integer variables fixed) are drawn as light lines. One of these linear programs becomes infeasible for $y \leqq y^1$, while the other becomes infeasible for $y \leqq y^2$. The pointwise minimum of these two functions is $\phi_b(y)$, which is superimposed as a heavy line. The lower convex envelope of $\phi_b(y)$, namely $\phi_b^*(y)$, is superimposed as a line with alternating dots and dashes. It is clear that there is no duality gap (difference between $\phi_b(y)$ and $\phi_b^*(y)$) for $y^2 \leqq y \leqq y^3$ or $y^4 \leqq y$. A global subgradient of ϕ_b at $y = 0$ will exist

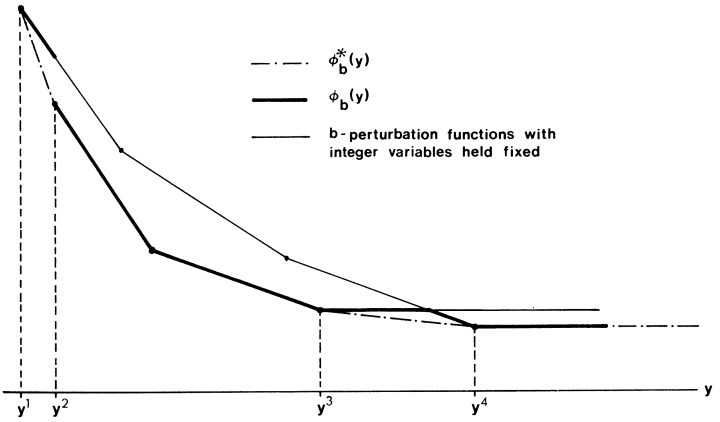

Fig. 1. Hypothetical illustration of b-perturbation functions.

(any subgradient of ϕ_b^* at $y = 0$ will do) if $y = 0$ falls in either of these intervals. If $y = 0$ falls between y^1 and y^2 or between y^3 and y^4, on the other hand, there will be a gap and no global subgradient of ϕ_b at $y = 0$ will exist.

We note in closing that the duality gap tends to be rather small for the class of problems with which we have numerical experience, namely capacitated facility location problems with additional constraints. The special constraints are as in Example 3. For four practical problems the values averaged as follows (after normalization via division by 0.01 v(P)):

$$\begin{aligned} v(\mathrm{P}) &= 100.00, \\ v(\mathrm{D}) &= 99.93, \\ v(\mathrm{PR}_{\bar{\lambda}}) &= 98.97, \\ v(\overline{\mathrm{P}}) &= 97.46. \end{aligned}$$

Notice that the duality gap is small by comparison with the gap between the integer problem and its usual LP relaxation, and that the LP multipliers $\bar{\lambda}$ yield a Lagrangean relaxation quite a bit better than the LP relaxation itself. See [15] for further details.

3. The use of Lagrangean relaxation in LP-based branch-and-bound

Virtually all of the current generally successful integer linear programming algorithms are of the branch-and-bound type with linear programming as the primary source of bounds [14]. This section and those to follow discuss the use of Lagrangean relaxation as a device for possibly improving the efficiency of such algorithms for special classes of problems.

A brief review of the usual LP-based branch-and-bound approach to (P) is necessary at this point. The terminology adopted is that of [14] which can be consulted for further details. At any given time there is a list of so-called *candidate problems*, each of which is simply (P) with certain additional "separation" constraints appended. The union of the feasible regions of the candidate problems constitutes a partition of the unenumerated portion of the feasible region of (P). There is also a number z^* representing the objective value of the *incumbent*, the best currently known feasible solution of (P) (initially z^* can be taken to be a suitably large number). The primary iterative step is to select one of the candidate problems, say (CP), and to examine it for the existence of a feasible solution of (P) with value better than z^*. The examination may be conclusive or inconclusive, depending on how much effort is expended; the usual practice involves solving the linear program ($\overline{\mathrm{CP}}$), which ignores all integrality conditions on the

variables of (CP). A conclusive examination is one for which the outcome is

(i) that (CP) is infeasible (e.g. $(\overline{\text{CP}})$ is infeasible), or
(ii) that $v(\text{CP}) \geqq z^*$ (e.g. $v(\overline{\text{CP}}) \geqq z^*$), or
(iii) that $v(\text{CP}) < z^*$ and an optimal solution of (CP) is at hand (e.g. the optimal solution $\bar{x}$ of $(\overline{\text{CP}})$ happens to satisfy the integrality conditions); this solution replaces the current incumbent and z^* is updated.

Then (CP) is said to be *fathomed* and is deleted from the list of candidate problems. Otherwise, (CP) is not fathomed and must be separated into two or more simpler candidate (sub)problems to be added to the list. This is accomplished via mutually exclusive and exhaustive separation constraints. The usual practice (cf. [3]) is to select a particular *separation variable* $j_0 \in I$ and to invoke an interval dichotomy on its range. For instance, for $j_0 = 3$ one subproblem might receive the new constraint $x_3 \leqq 2$ and the other the new constraint $x_3 \geqq 3$. Candidate problems continue to be examined in this fashion, with fathoming or separation occurring each time, until the list of candidate problems is exhausted.

It should be evident that a Lagrangean relaxation of (CP), say (CPR_λ), is just as amenable as the usual linear programming relaxation $(\overline{\text{CP}})$ as a device for examining candidate problems: the infeasibility of (CPR_λ) implies that of (CP); $v(\text{CPR}_\lambda) \geqq z^*$ implies $v(\text{CP}) \geqq z^*$; and an optimal solution of (CPR_λ), say x^{R}, is optimal in (CP) if it is feasible in (CP) and satisfies complementary slackness (see Theorem 1 (c)). Note that if x^{R} is feasible in (CP) but does not satisfy complementary slackness, it may still improve on the incumbent, in which case it should be used to update the incumbent and z^* even though (CP) is not fathomed. In cases where x^{R} is not feasible in (CP) it may be worth trying to adjust it in some problem-specific manner so as to gain feasibility and, hopefully, to improve thereby on the incumbent. This is exactly the same tactic as is commonly used with $(\overline{\text{CP}})$ when the (fractional) LP solution is rounded to satisfy integrality in the hope of obtaining an improved feasible solution.

The usual linear programming relaxation $(\overline{\text{CP}})$ is also used commonly to derive conditional bounds for use in guiding separation, for tagging newly created candidate subproblems with lower bounds on their optimal value, and for reducing the range restrictions on integer variables without sacrificing optimality. Lagrangean relaxations of (CP) can be used for these same purposes. Suppose that some variable $j \in I$ has a fractional value in the LP solution $\bar{x}$ of $(\overline{\text{CP}})$. We are interested in lower bounds on $v(\text{CP} \mid x_j \leqq \lfloor \bar{x}_j \rfloor)$ and $v(\text{CP} \mid x_j \geqq \lceil \bar{x}_j \rceil + 1)$, where " $\mid$ " signifies that the constraint following

it is appended to the problem, and $[\bar{x}_j]$ stands for the integer part of $\bar{x}_j$. Such conditional bounds are given, respectively, by

$$\begin{aligned} v_D(j) &\triangleq v(\mathrm{CPR}_\lambda \mid x_j \leqq [\bar{x}_j]), \\ v_U(j) &\triangleq v(\mathrm{CPR}_\lambda \mid x_j \geqq [\bar{x}_j] + 1). \end{aligned} \tag{4}$$

If $v_D(j) \geqq z^*$ holds, then the lower limit for x_j obviously can be tightened to $[\bar{x}_j] + 1$. Similarly, $v_U(j) \geqq z^*$ implies that the upper range restriction can be lowered to $[\bar{x}_j]$. It is even possible that both $v_D(j) \geqq z^*$ and $v_U(j) \geqq z^*$ hold, in which case it is clear that (CP) is fathomed. The bounds (4) can also be used to guide separation in the event that (CP) is not fathomed. Let $V_D(j)$ and $V_U(j)$ be computed for every $j \in I$ such that $\bar{x}_j$ is fractional. One appealing choice for the separation variable would be the one which maximizes the larger of $V_D(j)$ and $V_U(j)$ over all eligible j. Several successful integer programming codes have employed an analogous criterion based on $(\overline{\mathrm{CP}})$ rather than (CPR_λ). Once a separation variable j_0 is selected, $V_D(j_0)$ and $V_U(j_0)$ yield lower bounds for future reference on the newly created candidate subproblems.

The computation of conditional bounds like (4) is taken up in more detail in Section 4. We note here only that the bounding problems have the same structure as (CPR_λ) since we have assumed that range restrictions on all variables are incorporated into the special constraints $Bx \geqq d$, just as (CPR_λ) will have the same structure as (PR_λ) if, as is usually the case, the separation constraints employed are simple range restrictions on the variables.

Thus we see that Lagrangean relaxation can be used for the standard branch-and-bound tasks of fathoming, generating improved feasible solutions, range reduction, and guiding separation. It can also be used to derive surrogate constraints and cutting-planes. These uses are taken up in Section 5 and 6.

We turn now to a discussion of the strategy questions which arise in connection with the use of (CPR_λ) as an adjunct to $(\overline{\mathrm{CP}})$. The two main questions concern how λ is to be chosen and whether (CPR_λ) should be used before or after or even in place of $(\overline{\mathrm{CP}})$. These questions cannot be answered definitively in general, but an obviously important consideration is whether or not the Integrality Property defined in Section 2 holds for the particular constraint partition under consideration.

Suppose the Integrality Property does hold. Then (CPR_λ) can be infeasible only if $(\overline{\mathrm{CP}})$ is infeasible, and if $(\overline{\mathrm{CP}})$ is feasible, then by Theorem 2 it

must yield the best possible choice of λ for (CPR_λ) and $v(\mathrm{CPR}_{\lambda^*}) = v(\overline{\mathrm{CP}})$. Thus (CPR_λ) cannot fathom (CP) by infeasibility or by value unless $(\overline{\mathrm{CP}})$ would also do so. One can also show that at least one of the conditional bounds $V_{\mathrm{D}}(j)$ and $V_{\mathrm{U}}(j)$ must coincide with $v(\overline{\mathrm{CP}})$ for each variable that is fractional in an optimal solution of $(\overline{\mathrm{CP}})$ when the natural choice $\bar{\lambda}$ from $(\overline{\mathrm{CP}})$ is used in (4). Moreover, *both* of the bounds coincide with $v(\overline{\mathrm{CP}})$ in the special case of Examples 1 and 2 and perhaps in other cases as well. These facts argue against the use of a Lagrangean relaxation for which the Integrality Property holds. It has little to offer that cannot already be achieved by $(\overline{\mathrm{CP}})$, though it may possibly prove to be more fruitful than $(\overline{\mathrm{CP}})$ as a source of improved feasible solutions. It is important to recognize, however, that this negative conclusion rests on the implicit assumption that $(\overline{\mathrm{CP}})$ is of manageable size as a linear program. If this is not the case, then (CPR_λ) may be a comparatively attractive computational alternative. A beautiful illustration is provided by Held and Karp's work on the traveling-salesman problem. Here $(\overline{\mathrm{CP}})$ has such an enormous number of constraints that it is not practical to solve directly. Of course, the omission of $(\overline{\mathrm{CP}})$ necessitates the introduction of some method for computing a near optimal λ (see below). And even if $(\overline{\mathrm{CP}})$ *is* of manageable size it may still be sufficiently burdensome computationally that (CPR_λ) is attractive as a surrogate to be invoked *prior* to $(\overline{\mathrm{CP}})$ during the examination of a candidate problem. The hope is that the Lagrangean relaxation will permit (CP) to be fathomed without having to resort to the more expensive linear program $(\overline{\mathrm{CP}})$. The best choice for λ is likely to be a multiplier vector saved from the linear program corresponding to the prior candidate problem most closely related to the current one. Section 5 indicates how this tactic coincides in special cases with the use of surrogate constraints – a device which has proven quite effective computationally in some applications (cf. [14, Sec. 3.1.5]).

Now suppose that the Integrality Property does *not* hold. Then $(\overline{\mathrm{CP}})$ does not necessarily yield the best choice for λ, and (CPR_λ) may succeed in fathoming where $(\overline{\mathrm{CP}})$ fails. It makes strategic sense to invoke (CPR_λ) either before or after $(\overline{\mathrm{CP}})$ or even in lieu of it, depending on the relative tightness and computational expense of the two relaxations. The most effective strategy also depends on the role played by $(\overline{\mathrm{CP}})$ in generating the λ to be used by (CPR_λ), since $(\overline{\mathrm{CP}})$ can be used to generate a starting (or even final) value of λ which can then be improved upon by some independent method. To indicate the possible methods for finding a suitable λ we shall consider for the sake of notational convenience the situation encountered before any

branching has taken place. Then (CP) is (P) itself and (CPR_λ) is just (PR_λ). The general situation is entirely analogous.

There are two broad approaches to computing a suitable λ for (PR_λ): (sub)optimization of the concave Lagrangean dual problem (D) and (sub) optimization of the linear program (P*). The first approach yields λ directly, whereas the second yields λ indirectly as the multiplier vector associated with the $A\,x \geqq b$ constraints in (P*). The distinction should not be thought of as a rigid one; some methods can be described naturally from either viewpoint.

Consider the first approach. One of the most promising methods for seeking an optimal solution of (D) is via the Agmon–Motzkin–Schoenberg method as revived by Held and Karp [21]. See also the recent and extensive study of this method by Held, Wolfe and Crowder [22]. The idea is very simple. Let $\lambda^v \geqq 0$ be the current estimate of an optimal solution of (D) and let x^v be an optimal solution of (PR_{λ^v}). Then the new estimate is

$$\lambda^{v+1} = \max\{\lambda^v + \theta^v(b - A\,x^v), 0\},$$

where the max operator is applied component-wise and θ^v is a positive step size satisfying certain requirements [22]. The vector $(b - A\,x^v)$ is a subgradient of $v(PR_\lambda)$ at $\lambda = \lambda^v$ but the sequence $\langle v(PR_{\lambda^v})\rangle$ is not necessarily monotone. Favorable computational experience has been reported for several different applications [8, 21, 22]. An alternative is to carry out an ascent method for (D); see [8, 10, 20]. Still another method is to optimize (D) by tangential approximation (outer linearization/relaxation) making use of the fact that the evaluation of $v(PR_\lambda)$ for a given λ yields a linear support at that point. The available evidence [20, 24] suggests that convergence is slow in some applications. A combination of ascent and tangential approximation is possible with the BOXSTEP method of Hogan, Marsten and Blankenship [23].

Consider now the indirect approach via (P*). Perhaps the most obvious method is to apply generalized programming (Dantzig–Wolfe decomposition, inner linearization/restriction) with the convex hull portion of the constraints of (P*) represented in terms of its extreme points. The column-generation problems are precisely of the form (PR_λ). Since this method is equivalent to the tangential approximation method for (D), however, its efficiency is suspect. Another possibility is to apply the primal-dual simplex method to (P*) with special provisions to accomodate the convex hull. This method, developed by Fisher and by Fisher and Shapiro, can also be interpreted as an ascent method for (D). Some encouraging computational

experience has been reported [8]. In some applications the form of the constraints describing the convex hull in (P*) is known. Then it may be possible to apply the dual simplex method to (P*) with (most) violated constraints generated as needed. This is probably one of the best methods for obtaining a near-optimal λ fairly quickly when it applies. It has the added advantage of yielding valid constraints that may be appended to $(\overline{P})$ to make it a tighter relaxation of (P).

Other specialized techniques, both exact and heuristic, can be devised for (D) or (P*) in particular applications.

4. Penalties

The so-called "penalty" concept in integer programming was propelled to prominence by Driebeek [4], although the essential notion was used earlier by Dakin [3] and Healy [19]. The original idea was to underestimate the amount by which the optimal value of the LP relaxation of the current candidate problem would increase if separation were carried out using a particular separation variable. The estimates of change, referred to as penalties, can be used to help guide separation and may also permit fathoming or range reduction. An important subsequent refinement of this original idea was the recognition that it is the candidate problems and subproblems themselves, and not their LP relaxations, which are central to the underlying enumerative process. Tomlin [28, 29] showed how to modify the penalty formulae so as to take at least partial account of the integrality conditions. The resulting penalties are underestimates of the difference between $v(\overline{CP})$ and the optimal value of a candidate subproblem derived from (CP). See [14] for a discussion of current practice in the computation and use of penalties.

Lagrangean relaxation furnishes a convenient setting for deriving the simple and strengthened penalties alluded to above. This is done in subsection 4.1. More importantly, it leads naturally to extensions and specializations which do not follow as easily from the more traditional viewpoints. These are illustrated in subsections 4.2–4.4 for Examples 1–3. It is hoped that these improved penalties and their counterparts for other structures will add new vitality to the penalty concept by overcoming the limitations of standard penalties pointed out so clearly by Forrest, Hirst and Tomlin [11].

4.1. *Basic results: $B x \geqq d$ vacuous*

The first task is to show how the formulae for simple and strengthened penalties are related to Lagrangean relaxation. This requires taking $\lambda = \bar{\lambda}$ and specializing $B x \geqq d$ to be vacuous (in contrast to our usual convention, in this subsection, $B x \geqq d$ will not include upper bounds on the variables). Define I_f to be the indices in I such that $\bar{x}_j$ is fractional ($\bar{x}$ is the optimal solution of $(\overline{\text{CP}})$). It is easy to verify that the objective function coefficient of $(\text{CPR}_{\bar{\lambda}})$ vanishes for all such $j \in I_f$, and hence for all such j we have

$$V_{\text{D}}(j) \overset{\text{d}}{=} v(\text{CPR}_{\bar{\lambda}} \mid x_j \leqq [\bar{x}_j]) = v(\overline{\text{CP}}),$$
$$V_{\text{U}}(j) \overset{\text{d}}{=} v(\text{CPR}_{\bar{\lambda}} \mid x_j \geqq [\bar{x}_j] + 1) = v(\overline{\text{CP}}).$$

Thus the Lagrangean relaxation $(\text{CPR}_{\bar{\lambda}})$ appears to yield zero "down" and "up" penalties for separation on x_j.

A simple remedy is to employ an alternative representation for x_j in terms of variables whose objective function coefficients in $(\text{CPR}_{\bar{\lambda}})$ do not vanish. Such a representation is available from the final tableau of the linear program $(\overline{\text{CP}})$ since $j \in I_f$ must be basic therein:

$$x_j = \bar{x}_j - \sum_{i \in N} \bar{a}_{ji} x_i,$$

where N is the set of nonbasic variables. The use of this representation in the definition of $V_{\text{D}}(j)$ and $V_{\text{U}}(j)$ leads to the following conditional bounds: for $j \in I_f$,

$$\begin{aligned} V_{\text{D}}^*(j) &\overset{\text{d}}{=} v(\text{CPR}_{\bar{\lambda}} \mid \bar{x}_j - \textstyle\sum_{i \in N} \bar{a}_{ji} x_i \leqq [\bar{x}_j]), \\ V_{\text{U}}^*(j) &\overset{\text{d}}{=} v(\text{CPR}_{\bar{\lambda}} \mid \bar{x}_j - \textstyle\sum_{i \in N} \bar{a}_{ji} x_i \geqq [\bar{x}_j] + 1). \end{aligned} \tag{5}$$

Clearly,

$$V_{\text{D}}^*(j) \leqq v(\text{CP} \mid x_j \leqq [\bar{x}_j]),$$
$$V_{\text{U}}^*(j) \leqq v(\text{CP} \mid x_j \geqq [\bar{x}_j] + 1)$$

for all $j \in I_f$; that is, these conditional bounds really do underestimate the optimal value of the candidate problems that would result if (CP) were separated using x_j as the separation variable.

Unfortunately the computation of $V_{\text{D}}^*(j)$ and $V_{\text{U}}^*(j)$ may be onerous if $\bar{a}_{ji} \neq 0$ for some variables $i \in N \cap I$. The computation then requires solving a knapsack-type problem with some integer variables. Hence it is natural

to think of estimating $V_D^*(j)$ and $V_U^*(j)$ from below by simply dropping all integrality conditions:

$$\begin{aligned} V_D^{*0}(j) &\triangleq v(\overline{\mathrm{CPR}}_{\bar\lambda} \mid \bar x_j - \textstyle\sum_{i\in N} \bar a_{ji} x_i \leqq [\bar x_j]), \\ V_U^{*0}(j) &\triangleq v(\overline{\mathrm{CPR}}_{\bar\lambda} \mid \bar x_j - \textstyle\sum_{i\in N} \bar a_{ji} x_i \geqq [\bar x_j] + 1). \end{aligned} \tag{6}$$

The computation of each of these conditional bounds merely requires minimizing a linear function with all nonnegative coefficients [$(c - \bar\lambda A) \geqq 0$, by duality] subject to a single linear constraint and $x \geqq 0$. This is sometimes referred to as a "continuous knapsack" type problem and it is easy to write down an explicit solution:

$$V_D^{*0}(j) = v(\overline{\mathrm{CP}}) + (\bar x_j - [\bar x_j]) \underset{i\in N:\ \bar a_{ji} > 0}{\text{minimum}} \{(c - \bar\lambda A)_i / \bar a_{ji}\}, \tag{7}$$

$$V_U^{*0}(j) = v(\overline{\mathrm{CP}}) + ([\bar x_j] + 1 - \bar x_j) \underset{i\in N:\ \bar a_{ji} < 0}{\text{minimum}} \{(c - \bar\lambda A)_i / (-\bar a_{ji})\}$$

(we have used the fact that $\bar\lambda b = v(\overline{\mathrm{CP}})$ by LP duality). These conditional bounds are identical to those associated with the simplest penalties mentioned earlier (cf. (5) in [4]).

The strengthened penalties of Tomlin can also be recovered from this viewpoint by retaining the condition that x_j must not be in the open interval $(0, 1)$ for $j \in N \cap I$. Then we obtain

$$\begin{aligned} V_D^{*1}(j) &\triangleq v(\overline{\mathrm{CPR}}_{\bar\lambda} \mid \bar x_j - \textstyle\sum_{i\in N} \bar a_{ji} x_i \leqq [\bar x_j] \text{ and } x_i \notin (0,1) \text{ for all } i \in N \cap I), \\ V_U^{*1}(j) &\triangleq v(\overline{\mathrm{CPR}}_{\bar\lambda} \mid \bar x_j - \textstyle\sum_{i\in N} \bar a_{ji} x_i \geqq [\bar x_j] + 1 \text{ and } x_i \notin (0,1) \text{ for all } i \in N \cap I). \end{aligned} \tag{8}$$

It is not difficult to see that at most one variable need be at a positive level in an optimal solution of the modified continuous knapsack problems defined in (8). This observation leads to the explicit formulae:

$$V_D^{*1}(j) = v(\overline{\mathrm{CP}}) + \underset{i\in N:\ \bar a_{ji} > 0}{\text{minimum}} \begin{cases} (c - \bar\lambda A)_i (\bar x_j - [\bar x_j]) / \bar a_{ji} & \text{if } i \notin I, \\ (c - \bar\lambda A)_i \max\{(\bar x_j - [\bar x_j])/\bar a_{ji}, 1\} & \text{if } i \in I, \end{cases}$$

$$V_U^{*1}(j) = v(\overline{\mathrm{CP}}) + \underset{i\in N:\ \bar a_{ji} < 0}{\text{minimum}} \begin{cases} (c - \bar\lambda A)_i ([\bar x_j] + 1 - \bar x_j)/(-\bar a_{ji}) & \text{if } i \notin I, \\ (c - \bar\lambda A)_i \max\{([\bar x_j] + 1 - \bar x_j)/(-\bar a_{ji}), 1\} & \text{if } i \in I. \end{cases} \tag{9}$$

These formulae are identical with the strengthened penalties of Tomlin (cf. (10) and (11) of [28] or (3.5) and (3.6) of [29]). It is evident from the very definitions (5), (6) and (8) that

$$V_D^{*0}(j) \leqq V_D^{*1}(j) \leqq V_D^*(j),$$
$$V_U^{*0}(j) \leqq V_U^{*1}(j) \leqq V_U^*(j) \tag{10}$$

for all $j \in I_f$.

This completes the recovery of known formulae for Driebeek and Tomlin penalties for $j \in I_f$. Exactly the same type of analysis holds for penalties associated with *basic* variables of $I - I_f$. Such penalties are of interest as a means of obtaining tighter ranges on integer variables which happen to be naturally integer in the optimal solution of $(\overline{\text{CP}})$. For a *nonbasic* variable variable x_j in $I - I_f$ the quantity of interest is $v(\text{CPR}_\lambda \mid x_j \geqq 1)$; no alternative representation in terms of nonbasic variables is possible. Evidently,

$$v(\text{CPR}_{\bar\lambda} \mid x_j \geqq 1) = v(\overline{\text{CP}}) + (c - \bar\lambda A)_j. \tag{11}$$

Again this is a standard result.

Another technique for strengthening (6) makes use of the following elementary and well-known result.

Theorem 5. *Let* (IP) *be a minimizing integer linear program in which exactly one variable, say* x_h, *is declared to be integer-valued. Suppose that* $\bar x_h$, *the optimal level of* x_h *when* (IP) *is solved ignoring the integrality requirement, is fractional. Then the optimal value of* (IP) *is given by*

$$v(\text{IP}) = \min\{v(\overline{\text{IP}} \mid x_h = [\bar x_h]),\ v(\overline{\text{IP}} \mid x_h = [\bar x_h] + 1)\}.$$

The possibility that $(\overline{\text{IP}} \mid x_h = [\bar x_h])$ or $(\overline{\text{IP}} \mid x_h = [\bar x_h] + 1)$ or both are infeasible is not excluded (recall that our convention is to define a minimum over an empty set as $+\infty$).

Let $i_D(j)$ be the minimizing nonbasic i in the formula for $V_D^{*0}(j)$ given in (7). The index $i_U(j)$ is defined similarly. Then application of Theorem 5 in the obvious way permits the following improvement on (6) to be computed with only a little extra effort:

$$V_D^{*2}(j) \overset{\text{d}}{=} v(\overline{\text{CPR}}_{\bar\lambda} \mid \bar x_j - \textstyle\sum_{i \in N} \bar a_{ji} x_i \leqq [\bar x_j] \text{ and } x_{i_D(j)} \text{ integer}),$$
$$V_U^{*2}(j) \overset{\text{d}}{=} v(\overline{\text{CPR}}_{\bar\lambda} \mid \bar x_j - \textstyle\sum_{i \in N} \bar a_{ji} x_i \geqq [\bar x_j] + 1 \text{ and } x_{i_U(j)} \text{ integer}). \tag{12}$$

Neither (12) nor (8) necessarily dominates the other; one may verify the following relationship for $j \in I_f$:

$$(\bar x_j - [\bar x_j])/\bar a_{ji_D(j)} \left\{\begin{matrix}\leqq\\ \geqq\end{matrix}\right\} 1 \Rightarrow V_D^{*1}(j) \left\{\begin{matrix}\geqq\\ \leqq\end{matrix}\right\} V_D^{*2}(j),$$
$$([\bar x_j] + 1 - \bar x_j)/(-\bar a_{ji_U(j)}) \left\{\begin{matrix}\leqq\\ \geqq\end{matrix}\right\} 1 \Rightarrow V_U^{*1}(j) \left\{\begin{matrix}\geqq\\ \leqq\end{matrix}\right\} V_U^{*2}(j). \tag{13}$$

We are unable to supply a reference to the conditional bounds (12) in the published literature. However, Armstrong and Sinha [1] have independently and very recently proposed a precisely analogous strengthening of (8) for the mixed integer 0–1 case. They report favorable computational experience.

So far we have required $B\,x \geqq d$ to be vacuous; that is, all upper bounds and other special constraints are treated as general A-type constraints. Analogs of the previous penalty results as well as new penalty results emerge easily by allowing $B\,x \geqq d$ to be nonvacuous. This will now be illustrated for the three examples.

4.2. *Penalties for Example* 1

Example 1 differs from the previous development only in that ($\mathrm{CPR}_{\bar{\lambda}}$) now has upper-bounded variables. As indicated in Section 3, it can be shown that both $V_{\mathrm{D}}(j)$ and $V_{\mathrm{U}}(j)$ equal $v(\overline{\mathrm{CP}})$ for all $j \in I_f$ due to the vanishing of the corresponding objective function coefficients in ($\mathrm{CPR}_{\bar{\lambda}}$). The remedy for these vanishing penalties is again to invoke the representation for x_j which is available from the final LP tableau of ($\overline{\mathrm{CP}}$). This representation will be written as

$$x_j = \alpha_{j0} - \sum_{i \neq j} \alpha_{ji} x_i \quad \text{for } j \in I_f, \tag{14}$$

where, of course, many of the coefficients α_{ji} may be 0. The resulting strengthened conditional lower bounds on $v(\mathrm{CP} \mid x_j \leqq [\bar{x}_j])$ and $v(\mathrm{CP} \mid x_j \geqq [\bar{x}_j] + 1)$ for $j \in I_f$ are

$$\begin{aligned} V_{\mathrm{D}}^*(j) &\triangleq v(\mathrm{CPR}_{\bar{\lambda}} \mid x_j = \alpha_{j0} - \textstyle\sum_{i \neq j} \alpha_{ji} x_i \leqq [\bar{x}_j]), \\ V_{\mathrm{U}}^*(j) &\triangleq v(\mathrm{CPR}_{\bar{\lambda}} \mid x_j = \alpha_{j0} - \textstyle\sum_{i \neq j} \alpha_{ji} x_i \geqq [\bar{x}_j] + 1). \end{aligned} \tag{15}$$

We have used the notations V_{D}^* and V_{U}^* as in (5) because (15) is an exact counterpart of (5). Like (5), (15) could be too expensive computationally because each estimate requires solving a knapsack-type problem in integer variables. The fact that the knapsack problem now has upper-bounded variables is a dubious advantage. The most easily computed lower approximation to (15) is obtained by dropping the integrality requirements as in (6):

$$\begin{aligned} V_{\mathrm{D}}^{*0}(j) &\triangleq v(\overline{\mathrm{CPR}}_{\bar{\lambda}} \mid x_j = \alpha_{j0} - \textstyle\sum_{i \neq j} \alpha_{ji} x_i \leqq [\bar{x}_j]), \\ V_{\mathrm{U}}^{*0}(j) &\triangleq v(\overline{\mathrm{CPR}}_{\bar{\lambda}} \mid x_j = \alpha_{j0} - \textstyle\sum_{i \neq j} \alpha_{ji} x_i \geqq [\bar{x}_j] + 1). \end{aligned} \tag{16}$$

The notations V_D^{*0} and V_U^{*0} have again been carried over. The differences

$$V_U^{*0}(j) - v(\overline{CP}), \qquad v_D^{*0}(j) = v(\overline{CP}) \tag{17}$$

are Driebeek-like up and down penalties for Example 1. The computation of (16) requires only slightly more effort than the computation of (6). A "continuous knapsack" problem with upper-bounded variables must now be solved. Explicit formulae for V_D^{*0} and V_U^{*0} are slightly more cumbersome than expression (7), but are easily programmed for a computer.

To strengthen (16) one may formally write the counterpart of (8), but unfortunately explicit calculation may be nearly as costly as that of (15) itself. This is because the upper bounds generally invalidate the key property of (8) that at most one variable need be at a positive level in an optimal solution of each associated optimization problem. Thus the strengthened penalties of Tomlin do not generalize usefully to $Bx \geqq d$ when it includes upper bounds on variables.

But the other technique based on Theorem 5 for strenthening $V_D^{*0}(j)$ and $V_U^{*0}(j)$ *does* generalize nicely. Let $i_D(j)$ and $i_U(j)$ be respectively the fractional-valued variables in the solutions of the optimizations corresponding to $V_D^{*0}(j)$ and $V_U^{*0}(j)$. It is easy to see that at most one variable need be fractional in each of these solutions; if none is, then that penalty cannot be strengthened by the present device. The strengthened conditional bounds analogous to (12) are:

$$\begin{aligned} V_D^{*2}(j) &\triangleq v(\overline{CPR}_{\bar{x}} \mid x_j = \alpha_{j0} - \textstyle\sum_{i \neq j} \alpha_{ji} x_i \leqq [\bar{x}_j] \text{ and } x_{i_D(j)} \text{ integer}), \\ V_U^{*2}(j) &\triangleq v(\overline{CPR}_{\bar{x}} \mid x_j = \alpha_{j0} - \textstyle\sum_{i \neq j} \alpha_{ji} x_i \\ &\qquad \geqq [\bar{x}_j] + 1 \text{ and } x_{i_U(j)} \text{ integer}). \end{aligned} \tag{18}$$

The required optimizations are inexpensive to carry out. Clearly,

$$\begin{aligned} V_D^{*0}(j) \leqq V_D^{*2}(j) \leqq V_D^{*}(j) \leqq v(CP \mid x_j \leqq [\bar{x}_j]), \\ V_U^{*0}(j) \leqq V_U^{*2}(j) \leqq V_U^{*}(j) \leqq v(CP \mid x_j \geqq [\bar{x}_j] + 1). \end{aligned} \tag{19}$$

Exactly the same types of penalties can be constructed for $j \in I - I_f$ when the objective function coefficient of x_j vanishes in $(CPR_{\bar{x}})$.

4.3. *Penalties for Example 2*

The development of penalties for Example 2 closely parallels that for Example 1. For $j \in I_f$, both up and down penalties again vanish, and it is

necessary to use the final LP tableau representation of the form (14).[3] The resulting conditional bounds $V_D^*(j)$ and $V_U^*(j)$ defined in (15) may still be too expensive computationally to use in general, but the multiple choice constraints do tend to make the computation easier by comparison with Example 1. There are nontrivial situations where $V_D^*(j)$ and $V_U^*(j)$ can be computed relatively economically by a simple enumerative procedure. But in general one may have to fall back on the Driebeek-like penalties defined by (16). The required computations are no longer simple continuous knapsack problems with upper-bounded variables, but they can still be carried out efficiently by specialized techniques (e.g. by parametric optimization applied to the dual of $(\overline{\text{CPR}}_{\bar{\lambda}})$ with respect to the added constraint). Strengthening these penalties along the lines suggested by Tomlin as in (8) appears to be no easier in general than (15) itself. But again, as with Example 1, the strengthening of (18) based on Theorem 5 is attractive. The indices $i_D(j)$ and $i_U(j)$ may be selected to be any of the fractional-valued variables in the solutions of the optimizations corresponding to $V_D^{*0}(j)$ and $V_U^{*0}(j)$. The implementation of (18) on a computer is only slightly more expensive than that of (16). Naturally, (19) continues to hold. The reader should have no difficulty seeing what to do if penalties are desired for variables in $I - I_f$.

The special nature of the multiple choice constraints (1) makes it possible to define "cumulative" conditional bounds on the "upward" problems as follows:

$$V_U^{*0}(j; J_k) \overset{d}{=} \max \{ V_U^{*0}(j), V_D^{*0}(i) \quad \text{for } i \in \{J_k - j\} \} \tag{20}$$

where it is understood that j is in J_k in these definitions. That this provides true lower bounds on $v(\text{CP} \mid x_j = 1)$ follows from the fact that $x_j = 1$ implies $x_i = 0$ for all $i \neq j$ in the same multiple choice set. Similar cumulative bounds hold for V^{*2} and V^*.

4.4. *Penalties for Example* 3

For Example 3 we must distinguish between the "switching" (x_k) versus the "nonswitching" variables in I_f. The up and down penalties associated with $V_D(j)$ and $V_U(j)$ are highly unlikely to vanish for the fractional switching variables. In fact, one can argue that they are likely to be quite large. Our experience with the practical facility location problems mentioned at the end of Section 2 has been that these penalties tend to be at least *an order of*

[3] Numbered displays from the discussion of Example 1 will be used here with the understanding that (CP), etc., have the structure of Example 2 rather than Example 1.

magnitude greater than the standard Tomlin penalties when $\bar{\lambda}$ is used, and yet take less time to compute [15]. For the nonswitching variables in I_f, however, it is easy to see that the naive penalties vanish and thus that alternative representations from the final LP tableau may be useful. The detailed discussion would be so close to that for Example 2 that it will not be given here.

5. Surrogate constraints

Consider the case where (P) is a pure 0–1 integer program with $Bx \geqq d$ consisting solely of unit upper bounds on all variables. The present author proposed [12] the use of "surrogate" constraints (after Glover [16]) of the form

$$cx + \lambda(b - Ax) < z^*, \tag{21}$$

with the prescription that $\lambda \geqq 0$ be chosen as the optimal dual vector corresponding to $Ax \geqq b$ in (P) or some (CP). Clearly such a constraint must be satisfied by every feasible solution to (P) with lower objective value than that of the incumbent. Two uses of this type of surrogate constraint were proposed in connection with the examination of a typical candidate problem: as a possible means of fathoming via the easy test

$$\underset{x=0,1}{\text{minimum}} \{cx + \lambda(b - Ax)\} \overset{?}{\geqq} z^* \tag{22}$$

and as a possible means of range reduction via the following easy tests applied to a typical (say the jth) variable:

$$\underset{x=0,1}{\text{minimum}} \{cx + \lambda(b - Ax) \text{ s.t. } x_j = 0\} \overset{?}{\geqq} z^*, \tag{23a}$$

$$\underset{x=0,1}{\text{minimum}} \{cx + \lambda(b - Ax) \text{ s.t. } x_j = 1\} \overset{?}{\geqq} z^*. \tag{23b}$$

If (22) holds, then (P) is fathomed. If (23a) [resp. (23b)] holds, then x_j must be 1 [resp. 0] in any feasible solution of (P) which is superior in value to the current incumbent. It is understood, naturally, that all separation constraints must also be honored in taking the minima in (22) and (23) when examining a candidate problem subsequent to (P). If all separation constraints involve only additional range restrictions on the variables, as is usually the case, then (22) and (23) remain computationally trivial.

It is easy to interpret (22) and (23) from the viewpoint of Lagrangean relaxation (remember that $Bx \geqq d$ consists of just the upper bound constraints $x_i \leqq 1$). Test (22) can be rewritten as

$$v(\mathrm{PR}_\lambda) \overset{?}{\geqq} z^*, \tag{24}$$

which is precisely the elementary fathoming criterion normally associated with (PR_λ). Similarly, (23) can be rewritten as

$$v(\mathrm{PR}_\lambda \mid x_j = 0) \overset{?}{\geqq} z^*, \tag{25a}$$

$$v(\mathrm{PR}_\lambda \mid x_j = 1) \overset{?}{\geqq} z^*. \tag{25b}$$

This is precisely the ordinary range reduction criterion described in Section 3. And the injunction to obtain λ from the usual linear programming relaxation is a consequence of Theorem 2, which implies that the strongest tests are obtained in this way.

Thus the surrogate constraint (21) and the tests based on it are seen to be completely subsumed by the simplest Lagrangean relaxation techniques for the special case of Example 1. Generalizations of (21)–(23) when (P) is not a pure 0–1 program or when $Bx \geqq d$ includes more than simple upper bounds can be obtained without difficulty. Some such generalizations were developed several years ago by this author in unpublished lecture notes and by Glover [17] using the surrogate constraint viewpoint, but in each case the same results may be obtained easily as special cases of more general and powerful results based on Lagrangean relaxation.

6. Cutting planes

For present purposes, a *cutting-plane* is any linear inequality which must be satisfied by all of the feasible solutions of a candidate problem but is violated by an optimal solution of its usual linear programming relaxation $(\overline{\mathrm{CP}})$. Appending cutting-planes to $(\overline{\mathrm{CP}})$ makes it a tighter relaxation of (CP) and thereby yields better bounds for use in a hybrid branch-and-bound algorithm (cf. [14, Sec. IV]). Cutting-planes may also, of course, be used in a purely cutting-plane approach.

This section explores the uses of Lagrangean relaxation as a source of cutting-planes. For notational convenience we only consider cuts relative to the initial candidate problem (P) itself. It is a simple matter to apply the ideas developed below to any candidate problem.

The simplest type of cutting-plane for (P) is (here $\lambda \geqq 0$)

$$v(\mathrm{PR}_\lambda) \leqq cx + \lambda(b - Ax). \tag{26}$$

A special case of this cut was proposed by Shapiro [27], who showed that it can be at least as strong as all of the cuts in a well-known group theoretic

class. The validity of this constraint for any feasible solution of (P) follows from the definition of $v(\mathrm{PR}_\lambda)$ and the fact that the feasible region of (P) is contained in that of (PR_λ). It will be violated at $\bar{x}$, an optimal solution of $(\bar{\mathrm{P}})$, if $v(\mathrm{PR}_\lambda) > v(\bar{\mathrm{P}})$ holds, because $\lambda \geqq 0$ and $A\bar{x} \geqq b$ imply

$$v(\bar{\mathrm{P}}) = c\bar{x} \geqq c\bar{x} + \lambda(b - A\bar{x}).$$

The condition $v(\mathrm{PR}_\lambda) > v(\bar{\mathrm{P}})$ will hold when $v(\bar{\mathrm{P}}) < v(\mathrm{D})$ and λ is sufficiently near optimal in (D). Of course this condition is impossible when the Integrality Property holds; in fact, the Integrality Property implies that (26) cannot be violated by any solution of $(\bar{\mathrm{P}})$ whatever, because then

$$v(\mathrm{PR}_\lambda) = v(\overline{\mathrm{PR}}_\lambda) \leqq cx + \lambda(b - Ax)$$

for all x feasible in $(\overline{\mathrm{PR}}_\lambda)$ and thus for all x feasible in $(\bar{\mathrm{P}})$. Thus (26) can be a true cutting-plane only when the Integrality Property does not hold. Appending it to $(\bar{\mathrm{P}})$ must increase the optimal value of $(\bar{\mathrm{P}})$ at least to $v(\mathrm{PR}_\lambda)$ because (26) implies

$$cx \geqq v(\mathrm{PR}_\lambda) - \lambda(b - Ax) \geqq v(\mathrm{PR}_\lambda) \quad \text{for all } x \text{ feasible in } (\bar{\mathrm{P}}).$$

An improvement of (26) is obtained by replacing $v(\mathrm{PR}_\lambda)$ with $v(\mathrm{PR}_\lambda \mid x_{j_1}, \ldots, x_{j_P})$, which denotes the optimal value of (PR_λ) as a function of specified values for the distinguished *cut variables* $x_{j_1}, \ldots, x_{j_P}$. (If the values of the cut variables are such that no completion exists which is feasible in (PR_λ)—e.g., if an integer cut variable takes on a fractional value—then by convention, $v(\mathrm{PR}_\lambda \mid x_{j_1}, \ldots, x_{j_P})$ is defined to be $+\infty$ at such a point.) The constraint

$$v(\mathrm{PR}_\lambda \mid x_{j_1}, \ldots, x_{j_P}) \leqq cx + \lambda(b - Ax) \tag{27}$$

is valid by an argument similar to that for (26) and is uniformly at least as tight because

$$v(\mathrm{PR}_\lambda) \leqq v(\mathrm{PR}_\lambda \mid x'_{j_1}, \ldots, x'_{j_P}) \tag{28}$$

obviously holds for every feasible solution x' of (P). Strict inequality holds in (28) except when $x'_{j_1}, \ldots, x'_{j_P}$ happens to be part of an optimal solution of (PR_λ). This fact also renders (27) less susceptible to neutralization by the Integrality Property.

The difficulty with (27), of course, is that $v(\mathrm{PR}_\lambda \mid x_{j_1}, \ldots, x_{j_P})$ need not be a linear function. It depends upon the structure of (PR_λ) and the choice of cut variables. One source of nonlinearity has to do with the domain on which it is $+\infty$. Fortunately, (27) need only hold for *feasible* solutions of

(P), and so $v(\mathrm{PR}_\lambda \mid x_{j_1}, \ldots, x_{j_P})$ can be redefined arbitrarily wherever it is $+\infty$. It is clear that this redefinition should be linearly interpolative in nature. Of course, this still may not render (27) linear. It may be necessary to determine a linear lower bounding function $l_\lambda(x_{j_1}, \ldots, x_{j_P})$,

$$l_\lambda(x_{j_1}, \ldots, x_{j_P}) \leqq v(\mathrm{PR}_\lambda \mid x_{j_1}, \ldots, x_{j_P}) \quad \text{for all } x \text{ feasible in (P).} \tag{29}$$

Clearly, l_λ should be as "tight" as possible, especially in the vicinity of $\bar{x}$. Thus the linear constraint to be appended to $(\overline{\mathrm{P}})$ is of the form

$$l_\lambda(x_{j_1}, \ldots, x_{j_P}) \leqq c\,x + \lambda\,(b - A\,x), \tag{30}$$

where $(j_1, \ldots, j_P)$ is an arbitrary set of cut variable indices, $\lambda \geqq 0$, and (29) must hold.

The above ideas can be illustrated with reference to the three examples of Section 1. Examples 1 and 2 satisfy the Integrality Property and so constraint (26) cannot be violated at $\bar{x}$. Furthermore, it can be shown for these examples that no constraint of the form (30) can be violated at $\bar{x}$, no matter what λ or cut variables are chosen. Example 3, on the other hand, does lend itself to the derivation of useful cutting-planes. Cut (26) tends to be quite good, even when $\bar{\lambda}$, an immediate by-product of $(\overline{\mathrm{P}})$, is used. We see from the computational experience cited at the end of Section 2 that, in the four practical problems studied, a single cut of the form (26) with $\lambda = \bar{\lambda}$ raised the optimal value of $(\overline{\mathrm{P}})$ an average of at least 59.6% of the distance from $v(\overline{\mathrm{P}})$ to $v(\mathrm{P})$. If the effort to find an optimal λ were expended, (26) would raise the optimal value an average of at least 97.1% of the way to $v(\mathrm{P})$. It should also be noted that a cut of the form (30) is available as an immediate by-product of the evaluation of (PR_λ). Recall that (PR_λ) separates into independent subproblems of the form (3^k_λ):

$$\begin{aligned} v(\mathrm{PR}_\lambda) = \lambda\,b + \sum_{k=1}^{K} v(3^k_\lambda) \\ + \operatorname*{minimum}_{x_j,\ j \in T} \{\textstyle\sum_{j \in T} (c - \lambda A)\,x_j \text{ s.t. } 0 \leqq x_j \leqq u_j, \\ j \in T \text{ and } x_j \text{ integer}, j \in T \cap I\}, \end{aligned} \tag{31}$$

where T comprises the indices of all variables not appearing in any of the subproblems of type (3^k_λ). To evaluate $v(\mathrm{PR}_\lambda)$ one makes use of the fact that

$$v(3^k_\lambda) = \min\{v(3^k_\lambda \mid x_k = 0), v(3^k_\lambda \mid x_k = 1)\} \tag{32}$$

and of the fact that the last term involving $j \in T$ in (31) is trivially evaluated

by inspection. Thus the binary variables x_k are obvious choices for cut variables. We have

$$v(\mathrm{PR}_\lambda \mid x_1, \ldots, x_K) = \mathrm{CON}_\lambda + \sum_{k=1}^{K} v(3_\lambda^k \mid x_k), \tag{33}$$

where the constant CON_λ equals the first and last terms of (31). The binary nature of the variables makes it easy to write down a linear function l_λ satisfying (29) with equality in this case:

$$\mathrm{CON}_\lambda + \sum_{k=1}^{K} v(3_\lambda^k \mid x_k = 1)\, x_k = v(\mathrm{PR}_\lambda \mid x_1, \ldots, x_K)$$

for all binary $(x_1, \ldots, x_K)$. Thus (30) becomes

$$\mathrm{CON}_\lambda + \sum_{k=1}^{K} v(3_\lambda^k \mid x_k = 1)\, x_k \leqq c\,x + \lambda\,(b - A\,x). \tag{34}$$

Our experience with the same four practical problems as mentioned above is that a single cut of this type raised $v(\overline{\mathrm{P}})$ an average of 69.7% of the way from $v(\overline{\mathrm{P}})$ to $v(\mathrm{P})$ when $\bar{\lambda}$ was used [15].

The derivation of a type (30) cut for Example 3 generalizes easily to the frequent situation where (PR_λ) separates into a number of independent subproblems involving 0–1 variables. Suppose

$$v(\mathrm{PR}_\lambda) = \lambda\, b + \sum_{k=1}^{P} v(\mathrm{PR}_\lambda^k),$$

where (PR_λ^k) involves the variables J_k ($J_1, \ldots, J_P$ is a mutually exclusive and exhaustive partition) among which is a 0–1 variable j_k. Suppose further that both $v(\mathrm{PR}_\lambda^k \mid x_{j_k} = 0)$ and $v(\mathrm{PR}_\lambda^k \mid x_{j_k} = 1)$ can be obtained inexpensively in the course of evaluating $v(\mathrm{PR}_\lambda^k)$. Then j_k is a natural choice for a cut variable and a type (30) constraint is

$$\lambda\, b + \sum_{k=1}^{P} v(\mathrm{PR}_\lambda^k \mid x_{j_k} = 0)(1 - x_{j_k}) + v(\mathrm{PR}_\lambda^k \mid x_{j_k} = 1)\, x_{j_k} \leqq$$
$$\leqq c\,x + \lambda\,(b - A\,x). \tag{35}$$

We have made use of the relations

$$v(\mathrm{PR}_\lambda \mid x_{j_1}, \ldots, x_{j_P}) = \lambda\, b + \sum_{k=1}^{P} v(\mathrm{PR}_\lambda^k \mid x_{j_k}),$$

$$v(\mathrm{PR}_\lambda^k \mid x_{j_k}) = v(\mathrm{PR}_\lambda^k \mid x_{j_k} = 0)(1 - x_{j_k}) + v(\mathrm{PR}_\lambda^k \mid x_{j_k} = 1)\, x_{j_k} \quad \text{for } x_{j_k} = 0,1.$$

The latter relation furnishes the required l_λ function with equality in (29). Constraint (35) is likely to improve on the counterpart of (26), namely

$$\lambda b + \sum_{k=1}^{P} v(\mathrm{PR}_\lambda^k) \leqq c x + \lambda (b - A x), \tag{36}$$

because

$$\begin{aligned} v(\mathrm{PR}_\lambda^k) &= \min \{ v(\mathrm{PR}_\lambda^k \mid x_{j_k} = 0), v(\mathrm{PR}_\lambda^k \mid x_{j_k} = 1) \} \\ &\leqq v(\mathrm{PR}_\lambda^k \mid x_{j_k} = 0)(1 - x_{j_k}) \\ &\quad + v(\mathrm{PR}_\lambda^k \mid x_{j_k} = 1)\, x_{j_k} \quad \text{for } 0 \leqq x_{j_k} \leqq 1. \end{aligned}$$

It should also be pointed out that (35) and (36) can be decomposed into P component inequalities:

$$v(\mathrm{PR}_\lambda^k \mid x_{j_k} = 0)(1 - x_{j_k}) + v(\mathrm{PR}_\lambda^k \mid x_{j_k} = 1)\, x_{j_k} \leqq \sum_{j \in J_k} (c - \lambda A)_j x_j, \tag{35_k}$$

$$v(\mathrm{PR}_\lambda^k) \leqq \sum_{j \in J_k} (c - \lambda A)_j x_j. \tag{36_k}$$

The validity of (35_k) and (36_k) should be evident. Their sum over all k yields (35) and (36), respectively.

Other types of cutting-planes can be devised with the help of the penalty formulae of Section 4. In particular, useful cutting-planes for Examples 1 and 2 can be determined (recall that neither (26) nor (30) were useful in this context). Both the simple penalties based on (16) and the strengthened penalties based on (18) can be used to generate cuts violated by $\bar{x}$ so long as at least one of these penalties is nonzero. This may be done as follows. Consistency of notation requires that we let (CP) equal (P) when applying the results of Section 4.

Consider first the simple conditional bounds (16). Select any $j \in I_f$ such that at least one of the penalties is strictly positive and take this j to be the one cut variable. Clearly

$$\begin{aligned} &v(\mathrm{PR}_\lambda \mid x_j = \alpha_{j0} - \textstyle\sum_{i \neq j} \alpha_{ji} x_i) \leqq \\ &\leqq c x + \bar{\lambda} \leqq c x + \bar{\lambda}(b - A x) \quad \text{for all } x \text{ feasible in (P).} \end{aligned} \tag{37}$$

The left-hand side of (37) is convex as a function of x_j, and thus the unique linear function passing through it at the points $[\bar{x}_j]$ and $[\bar{x}_j] + 1$ does not overestimate it for any integer value of x_j:

$$\begin{aligned} &V_{\mathrm{D}}^{*0}(j) + (V_{\mathrm{U}}^{*0}(j) - V_{\mathrm{D}}^{*0}(j))(x_j - [\bar{x}_j]) \leqq \\ &\leqq v(\overline{\mathrm{PR}}_{\bar{\lambda}} \mid x_j = \alpha_{j0} - \sum_{i \neq j} \alpha_{ji} x_i) \quad \text{for all integer } x_j. \end{aligned} \tag{38}$$

Together, (37) and (38) imply that

$$V_D^{*0}(j) + (V_U^{*0}(j) - V_D^{*0}(j))(x_j - [\bar{x}_j]) \leqq c\,x + \bar{\lambda}\,(b - A\,x) \quad (39)$$

is a legitimate cut. Notice that if there are several $j \in I_f$ for which $V_D^{*0}(j)$ and $V_U^{*0}(j)$ are computed, then it is an easy matter to select j so as to yield the cut of type (39) which is most violated by $\bar{x}$.

Now consider the strengthened conditional bounds (18). An analog of inequality (37) holds, but the analog of (38) does not because of the added integrality requirement in (18). It appears necessary to require that $j \in I_f$ be a 0–1 variable if a cut is to be based on (18) with j as the single cut variable. Then

$$V_D^{*2}(j) + (V_U^{*2}(j) - V_D^{*2}(j))\,x_j \leqq c\,x + \bar{\lambda}\,(b - A\,x) \quad (40)$$

is a legitimate cut. By (19), (40) is clearly a superior cut to (39). It is a simple matter to select j so as to yield the cut which is deepest at $\bar{x}$ among those of the form (40).

For Example 2 one should of course use in (39) and (40) the cumulative penalties defined in (20) in place of $V_U^{*0}(j)$ or $V_U^{*2}(j)$ if the necessary quantities are at hand. One may further improve cuts (39) and (40) when j is a multiple choice variable by using one of the obvious cuts

$$\sum_{j \in J_k} V_U^{*0}(j; J_k)\,x_j \leqq c\,x + \bar{\lambda}\,(b - A\,x), \qquad k = 1, \ldots, K \quad (41)$$

or the still stronger cuts

$$\sum_{j \in J_k} V_U^{*2}(j; J_k)\,x_j \leqq c\,x + \bar{\lambda}\,(b - A\,x), \qquad k = 1, \ldots, K. \quad (42)$$

Each of these cuts takes all of J_k as the set of cut variables. It is easy to verify that cuts of the form (41) [resp. (42)] are at least as strong as those of the form (39) [resp. (40)] for all x feasible in (P).

A cut similar to (41) was proposed by Healy [19]. To be precise, for the kth cut he omitted the term $\bar{\lambda}\,(b - A\,x)$ and used $V_U^{*0}(j)$ as the coefficient of x_j, where $V_U^{*0}(j)$ is computed with $B\,x \geqq d$ taken to consist of only the kth multiple choice constraint (no upper bounds or other multiple choice constraints are included). This cut is dominated by (41).

We note in closing that penalty-based cuts with more than one cut variable can often be obtained for Examples 1 and 2 and other structures by: (i) adding to ($PR_{\bar{\lambda}}$) relations of the form (14) for any subset of j's in I_f so long as no variable appears with a nonzero coefficient in more than one of these relations, and then (ii) exploiting separability.

7. Conclusion

Lagrangean relaxation is a systematic exploitation of the formal Lagrangean dual problem in integer programming. This dual problem need not be solved optimally and need not be devoid of a duality gap in order to be useful. It provides a means for fathoming, range reduction, generating improved feasible solutions, and guiding separation (Sec. 3). It also provides new penalties (Sec. 4) and cutting-planes (Sec. 6) and supplants the narrower notion of surrogate constraints (Sec. 5). All of these functions usually can be tailored to the special structure of the particular problem class at hand, beginning with the judicious choice of the subset of constraints to play the role of $B\,x \geqq d$. This has been carried out in detail for three of the simplest structures. Some of the uses of Lagrangean relaxation have been explored by other authors for several more complex structures [6], [7], [8], [9], [10], [20], [21], [26]. Yet it remains to work out the full import of Lagrangean relaxation even for these structures and for many others of importance. It is hoped that the framework of this paper will facilitate this effort and encourage new applications.

Acknowledgment

This paper has benefited from careful readings by Marshall L. Fisher and Roy E. Marsten.

References

[1] R.D. Armstrong and P. Sinha, "Improved penalty calculations for a mixed integer branch-and-bound algorithm", *Mathematical Programming* 6 (1974) 212–223.
[2] R. Brooks and A. Geoffrion, "Finding Everett's Lagrange multipliers by linear programming", *Operations Research* 14 (1966) 1149–1153.
[3] Dakin, R.J., "A tree search algorithm for mixed integer programming problems", *Computer Journal*, 8 (1965) 250–255.
[4] N.J. Driebeek, "An algorithm for the solution of mixed integer programming problems", *Management Science* 12 (1966) 576–587.
[5] Everett, H.M., "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources", *Operations Research* 11 (1966) 399–417.
[6] M.L. Fisher, "Optimal solution of scheduling problems using Lagrange multipliers: Part I", *Operations Research* 21 (1973) 1114–1127.
[7] M.L. Fisher, "A dual algorithm for the one-machine scheduling problem", Graduate School of Business Rept., University of Chicago, Chicago, Ill. (1974).
[8] M.L. Fisher, W.D. Northup and J.F. Shapiro, "Using duality to solve discrete optimization problems: Theory and computational experience", Working Paper OR 030-74, Operations Research Center, M.I.T. (1974).

[9] M.L. Fisher and L. Schrage, "Using Lagrange multipliers to schedule elective hospital admissions", Working Paper, University of Chicago, Chicago, Ill. (1972).
[10] M.L. Fisher and J.F. Shapiro, "Constructive duality in integer programming", *SIAM Journal on Applied Mathematics*, to appear.
[11] J.J.H. Forrest, J.P.H. Hirst and J.A. Tomlin, "Practical solution of large mixed integer programming problems with UMPIRE", *Management Science* 20 (1974) 733–773.
[12] A.M. Geoffrion, "An improved implicit enumeration approach for integer programming", *Operations Research* 17 (1969) 437–454.
[13] A.M. Geoffrion, "Duality in nonlinear programming", *SIAM Review* 13 (1971) 1–37.
[14] A.M. Geoffrion and R.E. Marsten, "Integer programming algorithms: A framework and state-of-the-art survey", *Management Science* 18 (1972) 465–491.
[15] A.M. Geoffrion and R.D. McBride, "The capacitated facility location problem with additional constraints", paper presented to the Joint National Meeting of AIIE, ORSA, and TIMS, Atlantic City, November 8–10, 1972.
[16] F. Glover, "A multiphase-dual algorithm for the zero–one integer programming problem", *Operations Research* 13 (1965) 879–919.
[17] F. Glover, "Surrogate constraints", *Operations Research* 16 (1968) 741–749.
[18] H.J. Greenberg and T.C. Robbins, "Finding Everett's Lagrange multipliers by Generalized Linear Programming, Parts I, II, and III", Tech. Rept. CP-70008, Computer Science/Operations Research Center, Southern Methodist University, Dallas, Tex. revised (June 1972).
[19] W.C. Healy, Jr., "Multiple choice programming", *Operations Research* 12 (1964) 122–138.
[20] M. Held and R.M. Karp, "The traveling salesman problem and minimum spanning trees", *Operations Research* 18 (1970) 1138–1162.
[21] M. Held and R.M. Karp, "The traveling salesman problem and minimum spanning trees: Part II", *Mathematical Programming* 1 (1971) 6–25.
[22] M. Held, P. Wolfe and H.P. Crowder, "Validation of subgradient optimization", Mathematical Sciences Department, IBM Watson Research Center, Yorktown Heights, N.Y. (August 1973).
[23] W.W. Hogan, R.E. Marsten and J.W. Blankenship, "The BOXSTEP method for large scale optimization", Working Paper 660-73, Sloan School of Management, M.I.T. (December 1973).
[24] R.E. Marsten, private communication (August 22, 1973).
[25] G.L. Nemhauser and Z. Ullman, "A note on the generalized Lagrange multiplier solution to an integer programming problem", *Operations Research* 16 (1968) 450–452.
[26] G.T. Ross and R.M. Soland, "A branch and bound algorithm for the generalized assignment problem", *Mathematical Programming*, to appear.
[27] J.F. Shapiro, "Generalized Lagrange multipliers in integer programming", *Operations Research* 19 (1971) 68–76.
[28] J.A. Tomlin, "An improved branch and bound method for integer programming", *Operations Research* 19 (1971) 1070–1075.
[29] J.A. Tomlin, "Branch and bound methods for integer and non-convex programming", in: J. Abadie, ed., *Integer and nonlinear programming* (North-Holland, Amsterdam, 1970).
[30] A.F. Veinott and G.B. Dantzig, "Integral extreme points", *SIAM Review* 10 (1968) 371–372.