COMPUTER SCIENCES DEPARTMENT

University of Wisconsin-Madison

NEW SUFFICIENT OPTIMALITY CONDITIONS FOR INTEGER PROGRAMMING AND THEIR APPLICATION

by

J. M. Fleisher and R. R. Meyer

Computer Sciences Technical Report #278

October 1976

ABSTRACT

The purpose of this report is to present a new class of sufficient optimality conditions for pure and mixed integer programming problems. Some of the sets of sufficient conditions presented can be thought of as generalizations of optimality conditions based on primal-dual complementarity in linear programming, and these sufficient conditions are particularly useful for the construction of difficult integer programming problems with known optimal solutions. These problems may then be used to test and/or "benchmark" integer programming codes.

NEW SUFFICIENT OPTIMALITY CONDITIONS FOR INTEGER PROGRAMMING AND THEIR APPLICATION

by

J. M. Fleisher and R. R. Meyer

This research was supported by the National Science Foundation (Grant DCR 74-20584).

1. Introduction

In the past, integer programming test problems have generally been obtained from practical applications or have been randomly generated. These problems have had the disadvantage that their solutions could not be known for the purposes of benchmarking integer programming codes without explicitly solving them. If the problems are of a degree of difficulty appropriate to integer programming test problems, solving them to completion is an expensive if not a practically impossible procedure.

In this report, we present an alternate approach to test problem construction motivated by a procedure of Rosen [9] for the construction of (continuous variable) nonlinear programming test problems with known optimal solution via the use of the Kuhn-Tucker conditions [7]. Since the Kuhn-Tucker conditions themselves cannot serve as sufficient optimality conditions for integer programs, which are inherently non-convex, we have developed a new class of sufficient optimality conditions particularly appropriate for integer programs. In certain cases these conditions can be thought of as generalizations of linear programming complementarity conditions.

The class of problems to be considered are of the form[+]:

$$(P) \quad \begin{aligned} &\text{Maximize} && cx \\ &\text{subject to} && Ax \leq b \\ & && 0 \leq x \leq d \\ &x_j \text{ integer}, && j \in I \subseteq N = \{1,2,\ldots,n\} \end{aligned}$$

where A is an $m \times n$ matrix, c, d, and $x \in R^n$, $b \in R^m$, and I contains the set of indices corresponding to integer variables

[+]Most mixed integer programming formulations of physical models have bounded feasible regions and most integer programming codes require that upper bounds on the variables be specified. Corollary 4 considers the case in which upper bounds are absent. Vectors may be row vectors or column vectors. If A is an $m \times n$ matrix, x and $y \in R^n$ and $u \in R^m$, then xy will denote $\sum_{j=1}^{n} x_j y_j$ and uAx will denote $\sum_{i=1}^{m}\sum_{j=1}^{n} u_i A_{ij} x_j$. It is also assumed throughout the body of this report that all data elements are rational.

($I = N$ for pure integer programs). Given $x^* \in R^n$, conditions will be established which imply that x^* solves a problem of the form (P).

Existing sufficient optimality criteria for problems of the form (P) include the relaxation criterion [4]. The relaxation criterion states that if x^* is optimal for any problem whose objective function is cx and whose feasible region contains the feasible region of (P) (and is thus called a "relaxation" of P), then x^* solves (P) if and only if x^* is feasible for (P). This criterion forms the basis of branch and bound and cutting plane algorithms [4]. The relaxation criterion, however, is not amenable to constructing test problems because it leads to problems whose solutions are easily computed.

In Section 2, new sufficient optimality conditions are derived. In Section 3, examples of the sufficient optimality conditions are given. In Section 4, application of the sufficient conditions to test problem generation is discussed. In Section 5, some directions for further research are presented.

2. Optimality Conditions

In order to develop sufficient optimality conditions for x^* in (P), it is convenient to introduce a problem obtained from (P) by the transformation of variables $x = y + x^*$.

Lemma 1:

Consider the mixed integer program

$$(Q) \quad \begin{aligned} &\text{Maximize} && cy \\ &\text{subject to} && Ay \leq s^* \\ & && -x^* \leq y \leq d - x^* \\ &y_i \text{ integer}, && i \in I \subset N = \{1,2,\ldots,n\} \end{aligned}$$

where A is an $m \times n$ matrix, c, d, and $y \in R^n$, $x^* \in R^n_+(I,d)^\dagger$, and $s^* \in R^m_+$. Then $y^* = 0$ solves (Q) if and only if $x = x^*$ solves (P) with $b = Ax^* + s^*$.

$^\dagger R^n_+$ will denote $\{x \in R^n | x \geq 0\}$ and $R^n_+(I,d)$ will denote $\{x \in R^n | 0 \leq x \leq d$ and x_i integer for $i \in I\}$.

Proof:

The proof follows directly from the transformation of variables. ∎

For purposes of future reference, we state a lemma giving sufficient optimality conditions for (P) obtained from linear programming (LP). These conditions are of little practical interest since they will hold only if the integrality constraints of (P) are irrelevant. The new optimality conditions to be developed below are a generalization of these LP conditions in which the "complementarity" requirement ((3) below) is replaced by a "quasi-complementarity" requirement.

In what follows, A, b, c, and d will refer to data for (P) and m, n, and I will denote respectively the number of constraints, number of variables, and index set of integer variables for (P) .

Lemma 2:

Let $x^* \in R_+^n(I,d)$, s^*, $u^o \in R_+^m$, and $v^o, w^o \in R_+^n$. If

(1) $c = A^T u^o - v^o + w^o$,

(2) $b = Ax^* + s^*$,

(3) $s^* u^o + x^* v^o + (d-x^*)w^o = 0$,

then x^* solves (P) .

Proof:

Since $s^* \geqq 0$ and $0 \leqq x^* \leqq d$, $y^* = 0$ is feasible for (Q) . For y feasible for (Q) we have

(4) $Ay \leqq s^*$

(5) $-y \leqq x^*$,

(6) $y \leqq d - x^*$.

Multiplying (4) by u^o , (5) by v^o , (6) by w^o and summing yields

(7) $cy = u^o Ay - v^o y + w^o y \leqq s^* v^o + x^* v^o + (d-x^*)w^o = 0$

using (1) and (3). Since $cy \leqq 0$, $y^* = 0$ solves (Q) .

By (2) and Lemma 1, it follows that x^* solves (P) . ∎

Note that the triple (u^o, v^o, w^o) is a feasible solution to the dual of the linear program (CLQ) obtained by deleting the integrality requirements of (Q) , and that $0 = s^* u^o + x^* v^o + (d-x^*)w^o$ is the objective value of the dual of (CLQ) at (u^o, v^o, w^o) . By the duality theory of linear programming, 0 is an upper bound on the optimal objective value of (CLQ) , and thus must also be an upper bound on the optimal objective value of (Q) . Since $y^* = 0$ is feasible for (Q) , it must therefore also be optimal. This line of argument may be thought of as an alternative method of proof for the lemma.

The conditions used in Lemma 2 are the Kuhn-Tucker conditions for linear programming [7]. No use is made of the integrality requirements in (P) and, as may be seen from the alternate proof, the Kuhn-Tucker conditions are not satisfied at the optimum x^* unless the solution to (CLP) , the continuous relaxation of (P) , is also x^* . It is thus desirable to have more general optimality conditions which make use of the integrality requirements.

The following lemma gives a "double relaxation" optimality condition for (Q) that may be specialized to yield a variety of optimality conditions.

Lemma 3: (Double Relaxation Conditions)

Consider the problem

(8) Maximize cy
subject to $y \in F^*$

where c and $y \in R^n$ and let $F_1 \supseteq F^*$, $F_2 \supseteq F^*$, and define $M_1(F_1)$ to be

$$(9) \quad \begin{array}{ll} & \sup\ cy \\ \text{subject to} & y \in F_1, \end{array}$$

and $M_2(F_2)$ to be

$$(10) \quad \begin{array}{ll} & \inf\ cy \\ \text{subject to} & cy > 0 \\ & y \in F_2. \end{array}$$

If $y^* = 0$ is feasible for (8) and $M_1(F_1) < M_2(F_2)$ then $y^* = 0$ is optimal for (8). The condition $M_1(F_1) < M_2(F_2)$ is also a necessary optimality condition if $F_1 = F_2 = F.^*$

Proof:

Suppose $y^* = 0$ is non-optimal for (8). Then there exists a $\bar{y} \in F^*$ such that $c\bar{y} > 0$. Since $\bar{y} \in F_1$ and $\bar{y} \in F_2$, $\bar{y}$ is feasible for both (9) and (10), from which we have $c\bar{y} \le M_1(F_1) < M_2(F_2) \le c\bar{y}$, a contradiction. Thus, $y^* = 0$ must solve (8).

The condition $M_1(F_1) < M_2(F_2)$ is also a necessary optimality condition in the case $F_1 = F_2 = F^*$. For, if $y^* = 0$ is an optimal solution of (8), then $F_1 = F^*$ implies $M_1(F_1) = 0$ and $F_2 = F^*$ implies $M_2(F_2) = +\infty$, since the set over which the inf is taken in (10) is empty. ∎

Note that the ordinary single-relaxation optimality conditions correspond to the special case in which F_1 is such that $M_1(F_1) = 0$ and $F_2 = F_1$ (so that $M_2(F_2) = +\infty$), and that the linear objective function cx could be replaced by a nonlinear objective function $\emptyset(x)$ in the problems (8), (9), and (10). The ability to employ two relaxations F_1 and F_2, one in a minimization problem and one in a maximization problem, provides a degree of flexibility that is unavailable when employing the ordinary relaxation optimality conditions, and this flexibility turns out to be particularly useful for test problem construction.

Lemma 4 below is a special case of Lemma 3 which provides a generalization of Lemma 2. Lemma 4 makes use of a generalization to rationals of the number theoretic concept of greatest common divisor (gcd). Specifically, the generalized greatest common divisor of n rationals, $c_1, c_2, \ldots, c_n$ (assumed not all 0's), denoted as $\text{ggcd}\ (c_1, c_2, \ldots, c_n)$, is defined to be the minimum of $\sum_{j=1}^{n} c_j z_j$ subject to $\sum_{j=1}^{n} c_j z_j > 0$ and z_j integer, $j = 1, 2, \ldots, n$.
It is shown in [3] and [8] that this definition is, in some sense, the dual of the usual definition of the gcd, and that its value is the usual gcd when the arguments c_j are all integers. Reference [3] also shows that the ggcd is well-defined and positive when the arguments c_j are rationals (not all 0), and gives a number of important properties of this function. If the arguments c_j are integers, the ggcd may be efficiently computed by the Euclidean algorithm in at most $5\lfloor \log_{10} c_p \rfloor + n + 3$ iterations where $c_p = \min_{c_i \neq 0} \{|c_i|\}$ as shown in reference [1].

Lemma 4:

Let $x^* \in R_+^n(I,d)$, s^*, $v^o \in R_+^m$, and v^o, $w^o \in R_+^n$. If

(11) $c = A^T u^o - v^o + w^o$

(12) $b = Ax^* + s^*$

(13) $\delta_0 = s^* u^o + x^* v^o + (d-x^*)w^o < \gamma_0$ where $\gamma_0 = \text{ggcd}(c_1, c_2, \ldots, c_n)$

(14) $j \notin I \Rightarrow c_j = 0$ (continuous variables have cost coefficients of 0),

then x^* solves (P).

Proof:

Following the same argument used in Lemma 2, $y^* = 0$ is feasible for (Q), and for any y feasible for (Q) we have

(15) $$cy = u^o Ay - v^o y + w^o y \leq s^* u^o + x^* v^o + (d-x^*)w^o = \delta_0 < \gamma_0$$

using (11) and (13).

Let $F^* \equiv \{y | Ay \leq s^*, -x^* \leq y \leq d - x^*,\ y_j \text{ integer},\ j \in I\}$,

$F_1 \equiv \{y | Ay \leq s^*,\ -x^* \leq y \leq d - x^*\}$,

$F_2 \equiv \{y | y_j \text{ integer},\ j \in I\}$,

so that F^* is the feasible region of (Q), F_1 is the continuous relaxation of (Q), and F_2 is the relaxation of (Q) obtained by discarding all constraints other than integrality. From (15) it follows that $M_1(F_1) = \langle \text{Max. } cy \text{ subject to } y \in F_1 \rangle < \gamma_0$ and from (14) and the definition of the ggcd it follows that

$M_2(F_2) = \langle \text{Min. } cy \text{ subject to } y \in F_2 \text{ and } cy > 0 \rangle$

$= \langle \text{Min. } \sum_{j \in I} c_j y_j \text{ subject to } \sum_{j \in I} c_j y_j > 0 \text{ and } y_j \text{ integer, } j \in I \rangle = \gamma_0$.

Thus, $M_1(F_1) < M_2(F_2)$ and it follows from Lemma 3 that $y^* = 0$ solves (Q), whence by (12) and Lemma 1, x^* solves (P). ∎

Lemma 4 is a generalization of Lemma 2 in that the primal solution pair (x^*, s^*) and the dual solution trio (u^o, v^o, w^o) are required to be complementary in Lemma 2 but in Lemma 4 the quantity δ_0 is allowed to assume a positive value less than γ_0. Equation (13) states that the primal solution pair (x^*, s^*) and the dual solution trio (u^o, v^o, w^o) are "not too far from being complementary". When (13) holds we will say that the solutions are δ_0-quasicomplementary.

Note that the continuous relaxation of (Q) may have an optimal value of δ_0. (This will occur if (u^o, v^o, w^o) is an optimal solution of the dual of the continuous relaxation of (Q).) Thus, the gap between the optimal values of (P) and (CLP) may be as large as δ_0 rather than 0 as in Lemma 2.

Note also that optimality conditions stronger than those of Lemma 4 can be obtained by including more constraints in F_1 and/or F_2. For example, F_2 could be taken to be the set $\{y | -x^* \leq y \leq d - x^*,\ y_i \text{ integer},\ i \in I\}$, in which case γ_0 may be replaced in (13) by the optimal value of

$$\begin{aligned} &\min\ (cx - cx^*) \\ &\text{s.t. } cx > cx^* \\ &\qquad 0 \leq x \leq d^* \\ &\qquad x_i \text{ integer for } i \in I \end{aligned}$$

(Note that this is an integer programming problem, since the continuous variables have cost coefficients of 0, and thus play no role.) Although the inclusion of such additional constraints would, in general, lead to an increase in the value of γ_0 appearing in (13) (i.e., γ_0 would no longer be $\text{ggcd}(c_1, c_2, \ldots, c_n)$, but rather a larger value), this approach would have the practical disadvantage of requiring the solution of an integer or mixed-integer programming problem in order to obtain γ_0 rather than the much simpler calculation of the ggcd. Thus, there is a <u>trade-off</u> between the size of the value of γ_0 appearing in (13) and the effort required to compute γ_0. The optimality conditions that we have elected to use do not require the solution of an integer program to obtain γ_0, but, on the other hand, generate a relatively small γ_0, reducing the likelihood that (13) will be satisfied. In constructing an integer programming algorithm based on Lemma 3, however, refinements in the relaxations F_1 and F_2 would generally be required. (See section 5.)

The following lemma will be used in a further generalization of optimality conditions for (P).

Lemma 5:

Suppose x^* is optimal for each of the problems

(16) Maximize $c^{(k)}x$

subject to $x^* \in F^*$, $k = 1,2,\ldots,p$

where x^*, $c^{(k)}$, $k = 1,2,\ldots,p \in R^n$ and let $\lambda_k \geq 0$ be scalars, $k = 1,2,\ldots,p$. Then x^* is optimal for the problem

(17) Maximize cx^*

subject to $x^* \in F^*$ where $c = \sum_{k=1}^{p} \lambda_k c^{(k)}$.

Proof:

Since x^* is optimal for the problems (16), it is feasible for (16) and (17). For any $x \in F^*$, $c^{(k)}x \leq c^{(k)}x^*$, $k = 1,2,\ldots,p$, and since $\lambda_k \geq 0$, $k = 1,2,\ldots,p$, it follows that

$$cx = \sum_{k=1}^{p} \lambda_k c^{(k)}x \leq \sum_{k=1}^{p} \lambda_k c^{(k)}x^* = cx^*$$

establishing the optimality of x^* for (17). ∎

A generalization of Lemma 4 is now obtained by representing c as a nonnegative linear combination of p vectors $c^{(1)}, c^{(2)}, \ldots, c^{(p)}$ such that x^* is optimal for problem (P) with $c = c^{(k)}$, $k = 1,2,\ldots,p$, and applying Lemma 5. The $c^{(k)}$s will be divided into two groups. The set T will denote the set of indices k such that $j \notin I \Rightarrow c_j^{(k)} = 0$, group I will contain those $c^{(k)}$s such

that $k \in T$, and group II will contain those $c^{(k)}$s such that $k \notin T$. Thus, group I contains those $c^{(k)}$s where only integer variables may have nonzero costs $c_j^{(k)}$ and group II contains those $c^{(k)}$s where some continuous variables have nonzero costs $c_j^{(k)}$. For the $c^{(k)}$s in group I, we require that a dual solution trio $(u^{(k)},v^{(k)},w^{(k)})$ be δ_k-quasi-complementary with the primal solution pair (x^*,s^*) where $\delta_k < \gamma_k = \text{ggcd}(c_1^{(k)},c_2^{(k)},\ldots,c_n^{(k)})$ and apply Lemma 4; for the $c^{(k)}$s in group II, we require that a dual solution trio $(u^{(k)},v^{(k)},w^{(k)})$ be complementary with the primal solution pair (x^*,s^*) and apply Lemma 2. The main sufficient optimality criteria for mixed integer programming problems now follows:

(SOC) Theorem 1: (Sufficient Optimality Criteria)

Let $x^* \in R_+^n(I,d)$, $s^* \in R_+^m$,

$u^{(k)} \in R_+^m$, $k = 1,2,\ldots,p$,

$v^{(k)},w^{(k)} \in R_+^n$, $k = 1,2,\ldots,p$,

$\lambda_k \geq 0$, $k = 1,2,\ldots,p$,

$T = \{k \mid j \notin I \Rightarrow c_j^{(k)} = 0\}$.

If

(18) $c^{(k)} = A^T u^{(k)} - v^{(k)} + w^{(k)}$, $k = 1,2,\ldots,p$,
(Dual Feasibility)

(19) $c = \sum_{k=1}^{p} \lambda_k c^{(k)}$,
(Composition)

(20) $b = Ax^* + s^*$,
(Primal Feasibility)

(21) $k \in T \Rightarrow \delta_k = s^*u^{(k)} + x^*v^{(k)} + (d-x^*)w^{(k)} < \gamma_k$
where $\gamma_k = \text{ggcd}(c_1^{(k)},c_2^{(k)},\ldots,c_n^{(k)})$,
(Quasicomplementarity)

(22) $\quad k \in T \Rightarrow s^*u^{(k)} + x^*v^{(k)} + (d-x^*)w^{(k)} = 0$

(Complementarity)

then x^* solves (P) .

Proof:

By (20), $s^* \geq 0$, and $0 \leq x^* \leq d$, x^* is feasible for (P) . If $k \in T$ and $c = c^{(k)}$ then by Lemma 4 x^* solves (P) . If $k \notin T$ and $c = c^{(k)}$ then by Lemma 2 x^* solves (P) .

Hence, by (19) and Lemma 5, x^* solves (P) . ∎

The quantities x^* and s^* will be referred to respectively as the solution vector and the slack vector, $u^{(k)}$, $v^{(k)}$, and $w^{(k)}$ will be referred to as u-, v-, and w- multipliers, the $c^{(k)}$ vectors will be referred to as component cost vectors, the λ_k scalars will be referred to as component weights, the component k such that $k \in T$ will be referred to as integer components, the components k such that $k \notin T$ will be referred to as continuous components, δ_k will be referred to as the index of quasicomplementarity for the k^{th} component and for integer components k, γ_k will be referred to as the critical index for the k^{th} component. In addition, the quantities $x^*v^{(k)} + (d-x^*)w^{(k)}$ and $s^*u^{(k)}$ will be referred to respectively as the solution quasicomplementary index and the slack quasicomplementarity index for the k^{th} component.

(SOC 1) Corollary 1:

Let $x^* \in R^n_+(I,d)$, $s^* \in R^m_+$,

$u^{(k)} \in R^m_+$, $k = 1,2,\ldots,p$,

$v^{(k)},w^{(k)} \in R^n_+$, $k = 1,2,\ldots,p$,

$\mu_k \geq 0$, $k = 1,2,\ldots,p$,

$T = \{k | j \notin I \Rightarrow c_j^{(k)} = 0\}$,

and $c^{(k)}$ be an integer vector, $k = 1,2,\ldots,p$. If

(23) $\quad c^{(k)} = A^Tu^{(k)} - v^{(k)} + w^{(k)}$, $k = 1,2,\ldots,p$,

(24) $\quad c = \sum_{k=1}^{p} \mu_k c^{(k)}$,

(25) $\quad b = Ax^* + s^*$,

(26) $\quad k \in T \Rightarrow \epsilon_k = s^*u^{(k)} + x^*v^{(k)} + (d-x^*)w^{(k)} < 1$,

(27) $\quad k \notin T \Rightarrow s^*u^{(k)} + x^*v^{(k)} + (d-x^*)w^{(k)} = 0$

then x^* solves (P) .

The optimality conditions in (SOC) are more general than the conditions in Lemma 4. Lemma 4 requires that only integer variables may have nonzero costs c_j and that the gap between the optimal objective value of (P) and the optimal objective value of (CLP) be less than $\gamma_0 = \text{ggcd}\ (c_1,c_2,\ldots,c_n)$. However, if c can be expressed as a nonnegative linear combination of component cost vectors $c^{(k)}$ where each problem (P_k) (problem (P) with $c = c^{(k)}$) is such that Lemma 2 or Lemma 4 holds, continuous variables in (P) may have nonzero costs c_j and the gap between the optimal objective value of (P) and the optimal objective value of (CLP) may be considerably larger than γ_0. Such cases are exhibited in the examples of Section 3 where the conditions in (SOC) hold and the conditions in Lemma 4 don't hold.

Although (SOC) is more general than the Kuhn-Tucker conditions and Lemma 4, it is not a necessary condition for x^* to be an optimum of (P) . That is, given an optimum x^* to (P) it may not be possible to find nonnegative scalars $\lambda_1,\lambda_2,\ldots,\lambda_p$, and dual solution trios, $(u^{(k)},v^{(k)},w^{(k)})$, $k = 1,2,\ldots,p$, such that the conditions in (SOC) hold. Such a case is exhibited in [3].

The following corollaries are immediate since bounded variable mixed integer programming problems can always be expressed in the form (P).

(SOC 2) Corollary 2:
Consider the problem

$$\text{(P2)}\quad \begin{array}{lll} \text{Maximize} & cx & \\ \text{subject to} & {}^{\dagger}A_i x \leqq b_i\,, & i \in Q_1 \\ & A_i x \geqq b_i\,, & i \in Q_2 \\ & A_i x = b_i\,, & i \in Q_3 \\ & 0 \leqq x \leqq d & \\ x_i \ \text{integer}, & i \in I \subset N = \{1,2,\ldots n\} & \end{array}$$

where A is an $m \times n$ matrix, $c, d,$ and $x \in R^n$, $b \in R^m$, and $Q_1 \oplus Q_2 \oplus Q_3 = M = \{1,2,\ldots,m\}$.

Let $x^* \in R^n_+(I,d)$, $s^* \in R^m_+$,
$u^{(k)} \in R^m$, $k = 1,2,\ldots,p$,
$v^{(k)}, w^{(k)} \in R^n_+$, $k = 1,2,\ldots,p$,
$\lambda_k \geqq 0$, $k = 1,2,\ldots,p$.
$T = \{k \mid j \notin I \Rightarrow c_j^{(k)} = 0\}$.

If

(28) $$c_j^{(k)} = \sum_{i \in Q_1 \cup Q_3} A_{ij} u_i^{(k)} - \sum_{i \in Q_2} A_{ij} u_i^{(k)} - v_j^{(k)} + w_j^{(k)}, \quad k = 1,2,\ldots,p, \; j = 1,2,\ldots,n,$$

(29) $$c = \sum_{k=1}^{p} \lambda_k c^{(k)},$$

(30) $$b_i = \begin{cases} A_i x^* + s_i^* & \text{if } i \in Q_1, \\ A_i x^* - s_i^* & \text{if } i \in Q_2, \\ A_i x^* & \text{if } i \in Q_3, \end{cases}$$

$^{\dagger}A_i$ denotes the i^{th} row of A.

(31) $$i \in Q_1 \cup Q_2 \Rightarrow u_i^{(k)} \geqq 0,$$

(32) $$k \in T \Rightarrow \delta_k = \sum_{i \in Q_1 \cup Q_2} s_i^* u_i^{(k)} + x^* v^{(k)} + (d - x^*) w^{(k)} < \gamma_k$$

where $\gamma_k = \text{ggcd}(c_1^{(k)}, c_2^{(k)}, \ldots, c_n^{(k)})$,

(33) $$k \notin T \Rightarrow \sum_{j \in Q_1 \cup Q_2} s_i^* u_i^{(k)} + x^* v^{(k)} + (d - x^*) w^{(k)} = 0,$$

then x^* solves (P2). Note that for the equality constraints, the corresponding $u_i^{(k)}$ multipliers are unrestricted in sign.

(SOC 3) Corollary 3:
Suppose (P3) is the same problem as (P2) except that the objective is to minimize. If all of the assumptions of (SOC 2) hold except that the algebraic signs in (28) are reversed then x^* solves (P3).

(SOC 4) Corollary 4:
Suppose (P4) is the same problem as (P2) except that there are no explicit upper bounds on the variables (i.e., no d vector). If all of the assumptions of (SOC 2) hold except that the $w^{(k)}$ vectors along with the terms in (28), (32), and (33) containing $w^{(k)}$ are omitted, then x^* solves (P4).

3. EXAMPLES

Example 1:

In this example, the vectors d, x^*, s^*, $u^{(k)}$, $v^{(k)}$, $w^{(k)}$, the scalars λ_k and γ_k, and the fourth row of A were specified a priori; the remainder of A, b, and c were then selected in such a manner that (SOC) would hold.

$$\begin{aligned}
\text{Maximize} \quad & 33x_1 + 5x_2 + 48x_3 + 20x_4 + 20x_5 \\
\text{subject to} \quad & 10x_1 + 3x_2 + 7x_3 + 4x_4 + 2x_5 \leqq 52 \\
& 8x_1 + 7x_2 + 12x_3 + 6x_4 + 10x_5 \leqq 103 \\
& 4x_1 + 0x_2 + 15x_3 + 14x_4 + 8x_5 \leqq 116 \\
& x_1 + x_2 + x_3 + x_4 + x_5 = 10
\end{aligned}$$

$0 \leqq (x_1, x_2, x_3, x_4, x_5) \leqq (5, 10, 7, 9, 10)$

x_1, x_2, x_3 integer.

Solution: Maximum objective value = 325,

$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*) = (0, 1, 5, 1.5, 2.5)$ and

$(s_1^*, s_2^*, s_3^*, s_4^*) = (3, 2, 0, 0)$.

We have in the notation of (SOC 2),

$$I = \{1, 2, 3\}, \quad Q_1 = \{1, 2, 3\}, \quad Q_2 = \phi, \quad Q_3 = \{4\} .$$

Let $\lambda_1 = \lambda_2 = \lambda_3 = 1$,

$$\begin{aligned}
u^{(1)} &= (2, 1, 0, -13), & v^{(1)} &= (0, 0, 0, 1, 1), & w^{(1)} &= (0, 0, 0, 0, 0), \\
u^{(2)} &= (1, 2, 1, -30), & v^{(2)} &= (0, 2, 1, 0, 0), & w^{(2)} &= (0, 0, 0, 0, 0), \\
u^{(3)} &= (0, 0, 0, 20), & v^{(3)} &= (0, 0, 0, 0, 0), & w^{(3)} &= (0, 0, 0, 0, 0).
\end{aligned}$$

Note that the $u_4^{(k)}$ are unrestricted in sign since the fourth constraint is an equality. Computation using (28) yields

$$\begin{aligned}
c^{(1)} &= (13, 0, 13, 0, 0), & \gamma_1 &= 13, \\
c^{(2)} &= (0, -15, 15, 0, 0), & \gamma_2 &= 15, \\
c^{(3)} &= (20, 20, 20, 20, 20), & \gamma_3 &= 20,
\end{aligned}$$

and
$$\sum_{k=1}^{3} \lambda_k c^{(k)} = (33, 5, 48, 20, 20) = c$$

whence (29) holds. Computation using (30) yields

$$b = (52, 103, 116, 10).$$

Since $Q_1 \cup Q_2 = \{1,2,3\}$, (31) holds. For this example, $T = \{1,2\}$ and (32) holds since

$$\delta_1 = \sum_{i=1}^{3} s_i^* u_i^{(1)} + x^* v^{(1)} + (d-x^*) w^{(1)} = 12 < 13 = \gamma_1 ,$$

$$\delta_2 = \sum_{i=1}^{3} s_i^* u_i^{(2)} + x^* v^{(2)} + (d-x^*) w^{(2)} = 14 < 15 = \gamma_2 .$$

Finally, (33) holds since

$$\sum_{i=1}^{3} s_i^* u_i^{(3)} + x^* v^{(3)} + (d-x^*) w^{(3)} = 0 .$$

Therefore, the conditions of (SOC 2) are satisfied. For this example, a solution to the continuous relaxation of the problem is $x^0 \cong (0.82142, 0.00000, 4.62502, 1.15177, 3.40178)$ with an objective value of ≈ 340.179. Thus, the gap between the optimal objective value of the problem and the optimal objective value of its continuous relaxation is 15.179. Note that Lemma 4 cannot furnish a proof of the optimality of x^* in this case.

Example 2:

The following class of problems is discussed in reference [6] as an example of a class of difficult test problems:

$$\begin{array}{ll} \text{Maximize} & -x_1 \\ \text{subject to} & 2px_1 - qx_2 = p \\ & x_1, x_2 \geqq 0 \ , \quad \text{integer.} \end{array}$$

where $(p, q) \geqq (1,3)$, integer, and $\gcd(2p,q) = 1$.

Solution: Maximum objective value $= -\frac{1}{2}(q+1)$, $(x_1^*, x_2^*) = (\frac{1}{2}(q+1), p)$, and $s_1^* = 0$.

Since no upper bounds are given, assume $(x_1, x_2) \leqq (p+q, p+q)$. We have

$$I = \{1,2\}, \quad Q_1 = Q_2 = \phi, \; Q_3 = \{1\} \ .$$

Since $\gcd(2p,q) = 1$, there exists integers μ_1, μ_2 such that $2p\mu_1 + q\mu_2 = 1$ [10].

Let $\lambda_1 = \lambda_2 = 1$,

$$u^{(1)} = (\mu_1), \quad v^{(1)} = (1,0), \quad w^{(1)} = (0,0),$$
$$u^{(2)} = (-\mu_1), \quad v^{(2)} = (0,0), \quad w^{(2)} = (0,0) \ .$$

Computation using (28) yields

$$c^{(1)} = (2\mu_1 p - 1, -\mu_1 q) = (-\mu_2 q, -\mu_1 q), \quad \gamma_1 \geqq q \ ,$$
$$c^{(2)} = (-2\mu_1 p, \mu_1 q), \quad \gamma_2 \geqq \min(\mu_1, 1) \ ,$$

and $c^{(1)} + c^{(2)} = (-1,0) = c$ whence (29) holds. Since $b = 2px_1^* - qx_2^*$, (30) holds. With one equality constraint, (31) holds trivially. For this example, T in (32) $= \{1,2\}$, and

$$\delta_1 = s^* u^{(1)} + x^* v^{(1)} + (d - x^*) w^{(1)} = \frac{1}{2}(q+1) < q \leqq \gamma_1 \ ,$$
$$\delta_2 = s^* v^{(2)} + x^* v^{(2)} + (d - x^*) w^{(2)} = 0 < \min(\mu_1, 1) \leqq \gamma_2 \ ,$$

whence (32) holds and (33) holds vacuously.

Therefore, the conditions of (SOC 2) are satisfied.

Jeroslow and Kortanek [6] show that the solutions to this class of problems require an arbitrarily large number of cuts using the Gomory algorithm as p and q become large. What is even more interesting is that the solution to the continuous relaxation of the problem is always $(x_1^0, x_2^0) = (\frac{1}{2}, 0)$ with an objective value of $-\frac{1}{2}$, regardless of p and q . Thus, as q becomes large, so does the differential between the optimal objective value of the problem and the optimal objective value of its continuous relaxation. The gap is $\frac{q}{2}$ and $\gamma_0 = \text{ggcd}(c_1, c_2) = 1$.

Example 3:

The following example is a plant location problem. Material is shipped from supply points to demand points along routes connecting the supply points to the demand points. Each supply point has associated with it a capacity limiting the amount of material which can be shipped out of it and a fixed cost incurred if any material is shipped out of it. Each demand point has an associated demand which is the amount of material required to be shipped into it. Each route has associated with it a variable cost incurred for each unit of material shipped along it. The problem is to minimize the total cost satisfying the capacity restrictions and the demand requirements.

The sufficient optimality conditions derived in Section 2 have been used to develop a test problem generator for such plant location problems (see Section 4). The generator was used to construct the following problem, with 3 supply points, 3 demand points, and 7 routes. (This small problem was constructed for illustrative purposes; much larger problems have been constructed with the generator.) The data for the problem is indicated in Figure 1.

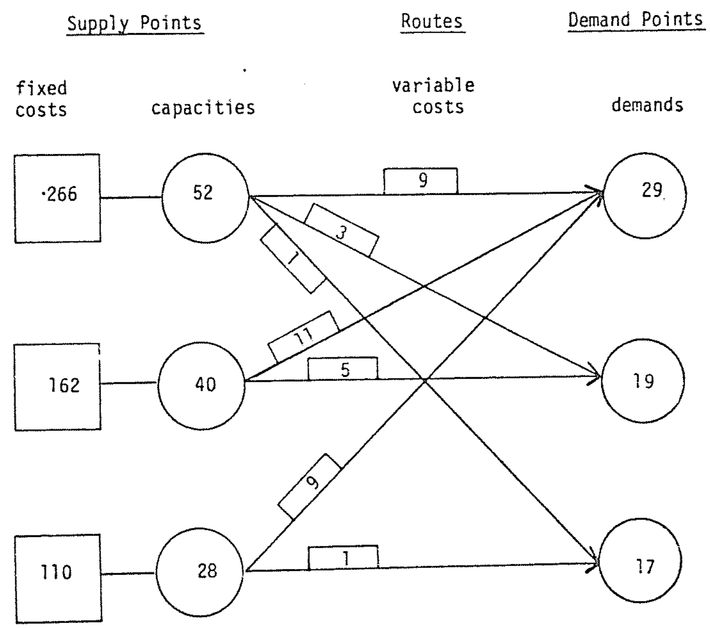

Figure 1.

A Plant Location Problem Constructed by the Test Problem Generator

The mixed integer programming formulation of the problem is

Minimize

$$9x_1 + 3x_2 + 1x_3 + 11x_4 + 5x_5 + 9x_6 + 1x_7 + 266x_8 + 162x_9 + 110x_{10}$$

subject to

$$\begin{array}{rrrrrrrrrrl}
-x_1 & -x_2 & -x_3 & & & & & +52x_8 & & & \geq 0 \\
 & & & -x_4 & -x_5 & & & & +40x_9 & & \geq 0 \\
 & & & & & -x_6 & -x_7 & & & +28x_{10} & \geq 0 \\
x_1 & & & +x_4 & & +x_6 & & & & & = 29 \\
 & x_2 & & & +x_5 & & & & & & = 19 \\
 & & x_3 & & & & +x_7 & & & & = 17
\end{array}$$

$x_1, x_2, x_3, x_4, x_5, x_6, x_7 \geq 0$, $x_8, x_9, x_{10} = 0$ or 1.

Here, $x_1, x_2, \ldots, x_7$ correspond to the amounts of material shipped along the routes, and x_8, x_9, x_{10} are 0-1 variables where 0 designates a closed supply point (no material shipped out of it) and 1 designates an open supply point (fixed cost incurred).

Solution: Minimum objective value = 681, with

$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*, x_7^*, x_8^*, x_9^*, x_{10}^*) =$

$(0, 0, 0, 18, 19, 11, 17, 0, 1, 1)$ and optimal slacks

$(s_1^*, s_2^*, s_3^*, s_4^*, s_5^*, s_6^*) =$

$(0, 3, 0, 0, 0, 0)$.

Thus, supply points 2 and 3 should be open (numbering them consecutively) and an optimal shipping schedule is given by the following table:

Route	Supply Point	Demand Point	Amount Shipped
1	1	1	0
2	1	2	0
3	1	3	0
4	2	1	18
5	2	2	19
6	3	1	11
7	3	3	17

We have

$I = \{1,2,\ldots,7\}$, $Q_1 = \emptyset$, $Q_2 = \{4,5,6\}$, $Q_3 = \{1,2,3\}$.

Let $\lambda_1 = 54/46$, $\lambda_2 = 1$,

$u^{(1)} = (3,3,3,3,3,3)$,

$v^{(1)} = (0,0,0,0,0,0,0,0,18,0)$

$w^{(1)} = (0,0,0,0,0,0,0,18,0,38)$

$u^{(2)} = (2,0,2,11,5,3)$

$v^{(2)} = w^{(2)} = 0$.

Note that $w_j^{(k)}$ must be 0 for $j \leq 6$ because there are no explicit upper bounds on x_j for $j \leq 6$.

Computation using (28) (with the algebraic signs reversed for a minimization problem) yields

$c^{(1)} = (0,0,0,0,0,0,0,138,138,46)$, $\gamma_1 = 46$,

$c^{(2)} = (9,3,1,11,5,9,1,104,0,56)$, $\gamma_2 = 1$,

and $\sum_{k=1}^{2} \lambda_k c^{(k)} = (9,3,1,11,5,9,1,266,162,110) = c$

whence (29) holds. Computation using (30) yields $b = (0,0,0,29,19,17)$. Since $u^{(1)}, u^{(2)} \geq 0$, (31) holds. For this example $T = \{1\}$ and (32) holds since

$$\delta_1 = \sum_{i=1}^{6} S_i^* u_1^{(1)} + \sum_{j=1}^{10} x_j^* v_j^{(1)} + \sum_{j=8}^{10} (1-x_j^*) w_j^{(1)} = 45 < 46 = \gamma_1.$$

Finally, (33) holds since

$$\sum_{i=1}^{6} S_i^* u_1^{(2)} + \sum_{j=1}^{10} x_j^* v_j^{(2)} + \sum_{j=8}^{10} (1-x_j^*) w_j^{(2)} = 0.$$

Therefore, the sufficient optimality conditions are satisfied. For this example, a solution to the continuous relaxation of the problem is

$x^0 \approx (1, 19, 17, 0, 0, 28, 0, .711538, 0, 1)$

with an optimal objective value of ≈ 634.269. The gap between the optimal objective value of the problem and the optimal objective value of its continuous relaxation is 46.731.

4. Applications to Test Problem Generation

The sufficient optimality conditions derived in Section 2 have been used to construct integer and mixed integer programs arising from three classes of problems with physical interpretations. These are generalized capital budgeting, plant location, and generalized transportation problems.

Generalized capital budgeting problems are pure integer programming problems with upper-bounded variables and nonnegative data. Reference [2] describes a procedure for the generation of generalized capital budgeting problems with known optimal solutions. Computational experience on problems generated by the procedure is also given in [2].

Plant location and generalized transportation problems are two classes of network problems which may be formulated as mixed integer programming problems. Both of these classes of problems have sparse constraint matrices. Reports are in preparation which will describe procedures for the generation of classes of problems of these types with known optimal solutions.

All of these procedures for constructing test problems of the form (P) with a known optimal solution x^* make use of (SOC). Some of the data comprising (P) are generated randomly according to parameter values specified by the user and the remainder of the data is generated in such a manner that (SOC) will be satisfied. The procedures have been coded in FORTRAN and can generate test problems and solutions according to a small number of parameter values specified by the user.

5. Directions for Further Research

As shown by an example in [3], the conditions of (SOC) are not necessary optimality conditions for all problems of the form (P). This suggests characterizing the class of problems that can be constructed using (SOC), and thereby determining a class of problems for which the condition of (SOC) are also necessary for optimality (i.e., they would be necessary in the sense that they must hold at some optimal solution). Alternatively, a class of mixed integer programs for which the (SOC) conditions were not necessary optimality conditions might be identified, and it might be possible to show that the problems in this class were, in some sense, a "difficult" set of mixed-integer programs.

Another area for further research lies in constructing an integer programming algorithm which makes use of the double relaxation conditions of Lemma 3. More specifically, Lemma 3 could be used in conjunction with the branch-and-bound algorithm by appropriately relating the sets F_1 and F_2 to the relaxations used in branch-and-bound.

Finally, the use of the dual variables u, v, and w of the (SOC) conditions for sensitivity analysis is being studied. For example, if x^* solves (P) and (SOC) holds, the u- multipliers can be used to determine a class of problems with varying right hand sides b such that x^* solves each of the problems in that class. Additionally, the relationship of these dual variables to the dual prices discussed by Gomory and Baumol [5] is under study.

References

1. G. H. Bradley, Algorithm and Bound for the Greatest Common Divisor of n Integers, Communications of the ACM, Volume 13, July, 1970, pp. 433-436.

2. J. M. Fleisher, Construction of Generalized Capital Budgeting Test Problems with Known Optimal Solutions, Computer Sciences Technical Report No. 260, University of Wisconsin, Madison, 1975.

3. J. M. Fleisher and R. R. Meyer, A New Class of Sufficient Optimality Conditions for Integer Programming, Computer Sciences Technical Report No. 248, University of Wisconsin, Madison, April, 1975.

4. R. S. Garfinkel and G. L. Nemhauser, Integer Programming, John Wiley & Sons, N. Y., 1972.

5. R. Gomory and W. Baumol, Integer Programming and Pricing, Econometrica, Volume 28, pages 521-550, 1960.

6. R. G. Jeroslow and K. O. Kortanek, On an Algorithm of Gomory, Siam Journal of Applied Math., Volume 21, July, 1971, pages 55-59.

7. O. L. Mangasarian, Nonlinear Programming, McGraw-Hill Book Co., N. Y., 1969.

8. R. R. Meyer and J. M. Fleisher, Strong Duality for a Class of Integer Programs, Computer Sciences Technical Report No. 256, University of Wisconsin, Madison, May, 1975.

9. J. B. Rosen and S. Suzuki, Construction of Nonlinear Programming Test Problems. Communications of the ACM, Volume 8, February, 1965, page 113.

10. H. M. Stark, An Introduction to Number Theory, Markham Publishing Co., Chicago, 1970.