Convergent Local Search

S. L. Savage, P. Weiner, M. J. Krone

Research Report #14

This work was partially supported by a grant from the Mobil Foundation to P. Weiner.

March 1973

Convergent Local Search

Abstract

When local search techniques are applied to finite combinatorial problems, the solution obtained may be globally optimal or may instead be only locally optimal relative to the neighborhood structure used. The main result of this work is the determination, for a wide class of problems, of necessary and sufficient conditions under which neighborhood search will converge, that is, obtain the globally optimal solution regardless of starting solution.

I. Introduction

Problems often arise in operations research in which, given a large but finite set of *feasible solutions* S, an extreme point of some real valued cost function on S is sought. This extreme point is known as an *optimal feasible solution*. Algorithms for finding optimal feasible solutions generally fall into one of the two following categories.

a) *constructive* algorithms, which construct a feasible solution in such a manner that it will be optimal upon completion, or

b) *improvement* algorithms, which start with a non-optimal feasible solution and construct from it a sequence of feasible solutions of monotone improving cost.

An obvious advantage of algorithms of the latter type is that they have produced a feasible solution at every stage. Thus if such an algorithm exceeds its limit on computation time, the procedure may be halted and a "good" solution obtained. Examples from the first category are *dynamic programming* (3,12) and *branch and bound* (20). Examples from the second category are *linear programming* (8,10) and the more general technique of *neighborhood search.*

Unlike the other examples above, neighborhood search does not represent a specific technique, but rather a general method which can be applied to almost any discrete optimization problem. The basic idea behind neighborhood search is to pick an initial feasible solution from S, and search a pre-defined neighborhood (a subset of S) associated with this initial solution for an improved solution. If such a solution is found in this neighborhood, then a pre-defined neighborhood associated with it is searched for further improvement. The search terminates when some solution is found to be optimal relative to its associated neighborhood.

Clearly such a technique can be applied to an extremely broad spectrum of problems. But as one might expect, a technique as general as this has some serious limitations. The most obvious of these is that when the search terminates, one has a solution which may be *locally optimal*, that is optimal within its neighborhood, but not necessarily globally optimal.

This drawback notwithstanding, neighborhood search has been used with considerable success of problems for which more formal techniques have failed (2,5,6,16,19,22,23,27). In such applications, it is hoped that the locally optimal solutions produced are globally optimal, or very close to globally optimal in cost a high proportion of the time.

The main result of this work is the determination, for the class of problems defined below, of necessary and sufficient conditions under which neighborhood search will produce a solution which is guaranteed to be globally optimal.

II. The Class of Problems Considered and Neighborhood Search

A broad class of discrete optimization problems can be characterized as follows. The *feasible solutions* are represented as a set of vectors $S = (s_1 \ldots s_m)$ in R^n. Given a vector $x \in R^n$, which is called the *parameter* of the problem, the cost of a feasible solution s_i is the inner product of s_i

with x, or $c(s_i,x) = s_i \cdot x$. That feasible solution with maximum cost for this x is said to be *optimal* with respect to x. It is reasonable to expect that it is possible for any feasible solution to be uniquely optimal with respect to some x. That is, for each s_i there is an $x_i \in R^n$ for which $s_i \cdot x_i > s_k \cdot x_i$ for all $k \neq i$. Indeed, we assume that every feasible solution in S satisfies this condition. Geometrically this means that for each s_i in S there exists a hyperplane with s_i on one side and all remaining points of S on the other. Thus no s_i can be a convex combination of other points in S, implying that S consists of the vertices of some convex polytope. If the hyperplanes defining this polytope are given explicitly, the problem can be phrased as a linear program and solved by the simplex method. There are, however, problems such as the traveling salesman problem, or in general integer programming problems, for which the feasible solutions S are the vertices of a convex polytope for which the defining hyperplanes cannot be determined efficiently.

We refer to this class of problems as *discrete linear optimization* (DLO) problems. An important sub-class of problems consists of those in which the feasible solutions are vectors of 0's and 1's. We refer to such problems as *subset* problems because the feasible solutions can be characterized as subsets of a fixed set of size n. The traveling salesman problem (2,4,5,6, 7,9,12,13,19,20,24,28) and minimal spanning tree problem (17) are examples of subset problems. The optimal binary search tree problem (15) is an example of a DLO problem which is not a subset problem. See (25) for a further discussion of these examples.

Given a DLO problem with solution set S and a parameter x, a common technique for maximizing $c(s,x)$ over S is *neighborhood search*. For every solution $s \in S$, a subset of S is defined to be the *neighborhood* $N(s)$ of s. When such a neighborhood has been defined for each $s \in S$ we say that a *neighborhood structure* N has been defined on S. A sequence of solutions in S is then generated as follows.

s_i, the initial solution, is arbitrary.

s_{i+1} can be any point in $N(s_i)$ such that $c(s_{i+1},x) > c(s_i,x)$.

When for some k, $c(s_k,x) \geqq c(s,x)$ for all $s \in N(s_k)$, s_k is said to be *locally optimal* with respect to the structure N. Note that s_k is not necessarily globally optimal, but the cost of elements of the sequence is strictly increasing.

We have not discussed the procedure by which s_1 is chosen or the order in which the solutions in $N(s_i)$ are searched for the improvement s_{i+1}. In practice, these choices are usually pseudo-random. The algorithm may be repeated on many different random starts, producing in general several different local optima of which the best is chosen as the final solution.

Definition: A neighborhood structure N is *exact* if for any $x \in R^n$ and any $s_i \in S$,

$$c(s_i,x) \geqq c(s,x) \text{ for all } s \in N(s_i) \Rightarrow s_i \text{ is optimal.}$$

That is, a neighborhood structure N is exact if any local optimum with respect to N is a global optimum. It should be clear that a neighborhood search technique can guarantee an optimal solution if and only if the neighborhood structure is exact. Consequently it is of some theoretical importance to be able to ascertain whether or not a neighborhood structure is exact.

In the next section we develop a useful characterization of the optimality of a given feasible solution. We show that the optimality of the feasible solution s_j with respect to the parameter x can be interpreted geometrically as meaning that x lies in a particular convex region R^n. We then show that there is a unique minimal subset of S such that whenever $s \cdot x \leqq s_j \cdot x$ for all solutions s in this subset then x lies in the convex region of interest. We refer to this subset of S as the 0-neighborhood of s_j, or $O(s_i)$. In section IV the minimal exact neighborhood structure for a DLO problem is shown to consist of the 0-neighborhoods.

III. Characterization of Optimality

We begin with a further discussion of what it means for a feasible solution to be optimal. Let the feasible solutions be the finite set S of distinct vectors $(s_1 \ldots s_m)$ in R^n. Suppose that given an arbitrary vector x in R^n, we wish to find an $s \in S$ such that $c(s,x) = s \cdot x$ is maximized over S. We call such an s an optimal feasible solution with respect to x and write $s = \mathrm{OPT}(S,x)$. By definition

(1) $$s_j = \mathrm{OPT}(S,x) \Leftrightarrow s_j \cdot x \geqq s_i \cdot x, \; i = 1 \ldots m.$$

For convenience we restate the above condition as follows. Let V_j be the set of vectors defined by $v_i = s_i - s_j$. An equivalent condition to (1) is then

(2) $$s_j = \mathrm{OPT}(S,x) \Leftrightarrow v \cdot x \leqq 0, \text{ for all } v \in V_j.$$

Notice that for s_j to be a feasible solution there must exist an x_j such that $v \cdot x_j < 0$ for all $v \in V_j$.

We address ourselves now to the problem of determining when the right-hand side of (2) is satisfied. For this expression to be satisfied the m-1 linear forms $(v_i \cdot x,\; i \neq j)$ must be simultaneously non-positive (v_j is by definition the zero vector). We show, using a geometrical argument, that if a certain subset of the above linear forms are non-positive, then they are all non-positive. The following definitions introduce a concept relating the optimality of s_j to a condition on the vector x.

Definition: Let C be a finite set of vectors in R^n. The *polar cone* of C is defined by $C^- = \{x \mid \forall y \in C,\; x \cdot y \leqq 0\}$.

Definition: We will say that a set of vectors C lies in a *closed* *half-space* if there exists a vector x such that $x \cdot y \leq 0$ for all $y \in C$, and that C lies in an *open half-space* if there exists a vector x such that $x \cdot y < 0$ for all $y \in C$.

We make three remarks.

1: For C^- to be non-empty, C must lie in a closed half-space.

2: Any positive linear combination of vectors in C^- must also be in C^-, hence C^- is convex.

3: Any positive linear combination of vectors in an open half-space must be non-zero.

The concept of polar cones allows us to express (2) conveniently as:

(3) $$s_j = \mathrm{OPT}(S,x) \Leftrightarrow x \in V_j^-.$$

Each v_i is the perpendicular to a hyperplane in R^n. The boundary of V_j^- is composed of some but not necessarily all of these hyperplanes. To ascertain that x lies in V_j^- one need only ascertain that x lies within the bounding hyperplanes of V_j^-. This is equivalent to ascertaining only that those linear forms in x whose associated hyperplanes are actually bounds of V_j^- are non-positive.

To state this formally will require some intermediate results.

Notation: When C is a set of vectors in R^n we write C-v to mean the set C less the vector v if $v \in C$, and C if $v \notin C$.

Definition: Let C be a finite set of distinct vectors in an open half-space of R^n. A vector (or point) v is *interior* to C if v can be expressed as a positive linear combination of points in C-v and is *extreme* on C otherwise. We denote by C* the set of all extreme points of C belonging to C.

Definition: A *generator* of a set of vectors C is a set of vectors U such that any point v in C can be expressed as a positive linear combination of points in U.

Definition: Let u and v be vectors. If there exists no real number b such that $u = bv$ we say that u and v are *non-parallel*. A set of vectors is said to be non-parallel if its members are pairwise non-parallel.

Lemma 1: Let C be a finite set of non-parallel vectors in an open half-space of R^n, then C* is the minimum generator of C belonging to C.

Proof: If a point v is a member of C* then, by definition, any positive linear combination of points in C expressing v must contain v. Hence any generator of C must contain C*. Therefore by showing that C* is a generator of C, we also show that it is the minimum generator of C. We must establish that if $v \in C$, then v is expressible as a positive linear combination of points in C*. There are two cases.

Case 1: v is extreme on C. The result is immediate.

Case 2: v is interior to C. In this case, v is expressible as a positive linear combination of other points in C. Pick such a combination. If it includes only points of C*, we are done. Suppose it includes some point u_k, which is interior to C. Now u_k can be expressed as a positive linear combination of points in $C-u_k$, or

$$u_k = \Sigma a_i u_i, \ a_i > 0, \ u_i \in C, \ i \neq k$$

Case 2a: If v does not appear as any of the u_i's in the above sum, then v is expressible as a positive linear combination of vectors in $C-u_k-v$, and is hence interior to $C-u_k$.

Case 2b: If v appears as some u_j in the above sum, then we have

$y = y + a_j v$, or

$v(1-a_j) = y$

where y is a positive linear combination of vectors in $C-u_k-v$. (The premise that C is a non-parallel set insures that $y \neq 0$.) Now $1-a_j$ must be positive, for if it were not, we would have the positive linear combination $(a_j-1)v + y = 0$. But since C lies in an open half-space this is impossible. Therefore we have

$$v = \frac{1}{1-a_j} \cdot y$$

so again v is expressible as a positive linear combination of vectors in $C-u_k-v$, and is hence interior to $C-u_k$.

We can in this way remove all interior points but v from C, while keeping v interior to the resulting set, which is C^*+v. This implies that v is expressible as a positive linear combination of points in C^*.

QED

Lemma 2: If $C_1 \subseteq C$, then $C^- \subseteq C_1^-$.

Proof: If $x \in C^-$ then $x \cdot v \leqq 0$, for all $v \in C$. In particular, $x \cdot u \leqq 0$ for all $u \in C_1$. Thus $C^- \subseteq C_1^-$.

QED

We now state a well known lemma (26) which is fundamental in proving the main result of the paper. A proof, adapted from (26), may be found in the appendix.

LEMMA 3 (Minkowski-Farkas lemma): Let v be a vector and C be a set of vectors in R^n not containing v. Then the following statements are equivalent.

1. v is extreme on C.

2. There exists an x such that $x \cdot u \leqq 0$ for all $u \in C$, and $x \cdot v > 0$.

For a given solution s_j, v_i will denote $s_i - s_j$, and V_j will denote the set $\{v_i, i \neq j\}$. Notice that since we assume the existence of an x_j such that $v \cdot x_j < 0$ for all $v \in V_j$, V_j must lie in an open half-space. We also have

Lemma 4: V_j is non-parallel for any j.

Proof: Let v_i and v_k, $i,k \neq j$ be any two vectors in V_j and assume there exists a real b such that $v_i = bv_k$. We cannot have $b = 0$ or 1, or we would have $s_i = s_j$ or $s_i = s_k$ respectively, and we have assumed distinct solutions. We discuss the three cases: $b < 0$, $0 < b < 1$, and $1 < b$, and show in each case that one of the vectors s_j, s_i, s_k must not be a feasible solution.

Case 1: If $b < 0$ then there can exist no x_j such that $v \cdot x < 0$ for all $v \in V_j$ and hence s_j is not feasible.

Case 2: If $0 < b < 1$ we have

$$s_i - s_j = b(s_k - s_j), \text{ or}$$

$$s_i = bs_k + (1-b)s_j.$$

Now consider the set V_i of vectors of the form $v = s - s_i$. We have

$$v_j = s_j - s_i = s_j - bs_k - (1-b)s_j$$

$$= -b(s_k - s_j)$$

and

$$v_k = s_k - s_i = s_k - bs_k - (1-b)s_j$$

$$= (1-b)(s_k - s_j).$$

If we set $c = \frac{b}{1-b} < 0$, then

$$v_j = cv_k.$$

Now by applying case 1 on the set V_i, we see that s_i cannot be feasible.

Case 3: If $1 < b$ we set $c = \frac{1}{b}$. Then we have $v_k = cv_i$, $0 < c < 1$, and by applying case 2 to the set V_k we see that s_k is not feasible.

QED

The next lemma makes use of the previous lemmas to arrive at a minimal condition under which the right-hand side of (3) is satisfied.

Lemma 5: V_j^* is the unique smallest subset P of V_j such that $P^- = V_j^-$.

Proof: By lemma 1, if v is in V_j, then v is expressible as a positive linear combination of points in V_j^*. Therefore if $x \cdot u \leq 0$ for all $u \in V_j^*$, then $x \cdot v \leq 0$ for all $v \in V_j$. Hence $(V_j^*)^- \subseteq V_j^-$. But from lemma 2, $V_j^- \subseteq (V_j^*)^-$. We may conclude that $(V_j^*)^- = V_j^-$.

Proof of minimality: Let U be a subset of V_j, such that $U^- = V_j^-$, and suppose some element v of V_j^* is not contained in U. Since $v \in V_j^*$, v is extreme on V_j, and therefore also extreme on U. So by lemma 3, there exists an x such that $u \cdot x \leq 0$ for all $u \in U$ and $v \cdot x > 0$. Hence $x \in U^-$ but $x \notin (V_j^*)^-$, contradicting the hypothesis that $U^- = V_j^-$. This proves that if $U^- = V_j^-$ then U must contain V_j^*.

QED

The above results imply that

(4) $$s_j = \mathrm{OPT}(S,x) \Leftrightarrow x \in (V_j^*)^-.$$

This implies that for (1), (2), and (3) to hold, it is sufficient for only those linear forms associated with the extreme points of V_j to be non-positive. Each of these extreme points v_i corresponds to a feasible solution $s_i = v_i + s_j$. The importance of these solutions in exact neighborhood search motivates the next definition.

Definition: We define the 0-neighborhood of s_j, $O(s_j)$, to be the set of feasible solutions $\{s_i | v_i \in V_j^*\}$.

Notice that the line $y = \alpha v_i$, where α is a real number, contains the solutions s_i and s_j. If v_i is extreme on V, the line $y = \alpha v_i$ is extreme on the polytope defined by the vertex set S, and is thus the intersection of n-1 of the defining hyperplanes of this polytope. Therefore the set $O(s_i)$ consists of the *polytopal neighbors* of s_i, that is, just those solutions which would be compared with s_i by the simplex algorithm. (Of course, the simplex algorithm cannot be applied unless the problem is converted to a linear program. As we have mentioned, this can always be done, although not by efficient methods.)

IV. Exact Neighborhood Search

We now consider neighborhood search as applied to DLO problems. From the definition of exactness, the neighborhood structure N is exact iff for every $s_j \in S$ and any $x \in R^n$,

$$(5) \qquad s_j \cdot x \geqq s \cdot x \text{ for all } s \in N(s_j) \Leftrightarrow s_j = OPT(S,x).$$

We are now ready to state our main result which characterizes the exactness of neighborhood structures for discrete linear optimization problems.

Theorem 1: The minimal exact neighborhood structure for a DLO problem has neighborhoods $N(s) = O(s)$.

Proof: Let V denote the set of vectors $\{s_i - s \mid s_i \in N(s)\}$. For a neighborhood structure N to be exact we have from (5): for every $s \in S$ and any $x \in R^n$,

$$v_i \cdot x \leqq 0 \text{ for all } v_i \in V \Rightarrow s = OPT(S,x)$$

For each $s_j \in S$ the unique smallest V satisfying the above is V_j^* from lemma 5. This implies that the smallest sets N(s) satisfying (5) must be O(s) for all $s \in S$.

QED

The above result immediately provides necessary and sufficient conditions for neighborhood search to guarantee an optimal solution to a DLO problem.

We next discuss the O-neighborhoods from a more intuitive point of view, and prove a theorem concerning a symmetry among O-neighborhoods. O-neighborhoods possess a property derived in lemma 6 which is useful in characterizing such neighborhoods explicitly for particular DLO problems. It is also possible to arrive at theorem 1 given this property of O-neighborhoods as a definition (25,28).

Lemma 6: $s_k \in O(s_j)$ iff there exists an $x \in R^n$ for which s_j has the second highest cost in S and s_k has the unique highest cost in S.

Proof: $\Leftarrow$ If s_j has the second highest cost in S, and s_k is optimal, then $v_k \cdot x = (s_k - s_j) \cdot x > 0$, with $v_i \cdot x \leqq 0$ for all $v_i \in V_j$, $i \neq k$. Now assume that $s_k \notin O(s_j)$. Then v_k is interior to V_j, or in other words

$$v_k = \Sigma a_i v_i,\ a_i > 0,\ i \neq k.$$

So $v_i \cdot x \leqq 0$ for all $v_i \in V_j$, $i \neq k$ implies $v_k \cdot x \leqq 0$ which is a contradiction.

$\Rightarrow$ If $s_k \in O(s_j)$, then v_k is not interior to V_j, and hence by lemma 3, there exists an $x \in R^n$, such that $v_k \cdot x > 0$, with $v_i \cdot x \leqq 0$ for all $v_i \in V_j$, $i \neq k$. This implies that for this x, $c(s_k,x) > c(s_j,x)$, with $c(s_j,x) \geqq c(s_i,x)$ for $i \neq k$, so s_k has the unique highest cost and s_j has the second highest cost.

QED

One may view a neighborhood structure on a feasible solution set S as a directed graph on a node set associated with S, with an arc pointing from node s_i to s_j iff $s_j \in N(s_i)$. If for all s_i and s_j, $s_j \in N(s_i) \Rightarrow s_i \in N(s_j)$, an undirected graph is clearly adequate to describe the structure. The next theorem shows that the structure imposed by the O-neighborhoods obeys this symmetry.

**Theorem 2*: If $s_k \in O(s_j)$, then $s_j \in O(s_k)$.

---

* See endnote.

Proof: The proof consists of the construction of a parameter x' for which

(6) $$\forall\, i \neq j,k,\ c(s_j,x') > c(s_k,x') > c(s_i,x')$$

implying by lemma 6 that $s_j \in O(s_k)$.

For (6) to hold we must have

(7) $$\forall\ i \neq j,k,\ 0 > v_k \cdot x' > v_i\ x'$$

Since $s_k \in O(s_j)$ we can choose an x such that

$$\forall\ i \neq k,\ v_k \cdot x > 0 \geqq v_i \cdot x$$

and by the definition of feasible solution we can choose a vector $y \in R^n$ such that

$$\forall\ i \neq j, v_i \cdot y < 0.$$

Letting $x' = x + ay$, (7) will hold iff

(8) $$v_k \cdot x + a(v_k \cdot y) < 0$$

and

(9) $$\forall\ i \neq j,k,\ v_k \cdot x + a(v_k \cdot y) > v_i \cdot x + a(v_i \cdot y).$$

Clearly (8) is satisfied for

$$a > -\frac{v_k \cdot x}{v_k \cdot y} > 0$$

We now show that there exists an a that simultaneously satisfies (8) and (9). Denote the maximum of $v_i \cdot x$, $i \neq j,k$ by $m \leqq 0$, and the maximum of $v_i \cdot y$, $i \neq j,k$ by $M < 0$. Then a satisfies (9) for

$$v_k \cdot x + a(v_k \cdot y) > m + aM$$

or equivalently

(10) $$v_k \cdot x - m > a(M - v_k \cdot y).$$

Now if $(M - v_k \cdot y) \leqq 0$ any a satisfying (8) also satisfies (10) and we are done. If $(M - v_k \cdot y) > 0$ then (10) implies

$$\frac{v_k \cdot x - m}{M - v_k \cdot y} > a$$

But since $m, M < 0, v_k \cdot x - m > v_k \cdot x$ and $M - v_k \cdot y < -v_k \cdot y$ and hence a can satisfy

$$\frac{v_k \cdot x - m}{M - v_k \cdot y} > a > -\frac{v_k \cdot x}{v_k \cdot y}.$$

QED

Corollary: If $s_k \in O(s_j)$ there exists an $x_1 \in R^n$ which renders s_k uniquely optimal with s_j uniquely second to optimal, and an $x_2 \in R^n$ for which s_k and s_j are tied for optimality.

Proof: Using the values of x and y from the above theorem and choosing a such that $0 < a < -\frac{v \cdot x}{v_k \cdot y}$, we have

$$v_k \cdot x_1 > 0 \text{ and } \forall\, i \neq j,k,\ v_i \cdot x_1 < 0$$

for $x_1 = x + ay$, implying that s_k is uniquely optimal with s_j uniquely second. For $b = -\frac{v \cdot x}{v_k \cdot y}$ and $x_2 = x + by$ we have

$$v_k \cdot x_2 = 0 \text{ and } \forall\, i \neq j,k,\ v_i \cdot x_2 < 0,$$

implying that s_k and s_j are tied for optimality.

QED

VI. Discussion

We feel the above result may play an important role in the synthesis of neighborhood search algorithms. It may also be used in the analysis of the complexity of DLO problems as follows. The last stage of neighborhood search clearly involves exhaustive search of the neighborhood of the local optimum produced. The size of the smallest neighborhood provides us with a lower bound on the run time of the algorithm. Thus if one wishes to develop an exact neighborhood search algorithm for some DLO problem, but finds that the O-neighborhoods are difficult to define explicitly, a lower bound on their size may be enough to show that exact neighborhood search is impractical (25, 28).

When the cardinality of O(s) is uniformly much smaller than the cardinality of S, neighborhood search may be a reasonable method for finding the optimal solution (25). There are problems for which the sets O(s) are so large that exact neighborhood search in inherently inefficient (25,28). There is reason to believe, however (25), that in problems where only a small fraction f of O(s) may be searched, the probability of arriving at a global optimum can be much greater than f.

Appendix

Lemma 3 (Minkowski-Farkas lemma): Let v be a vector and C be a set of vectors in R^n not containing v. Then the following statements are equivalent.

1. v is extreme on C.

2. There exists an x such that $x \cdot u \leqq 0$ for all $u \in C$, and $x \cdot v > 0$.

Proof: $(2 \Rightarrow 1)$ We prove the contrapositive. Thus we assume v to be interior to C and show that 2 cannot hold. If v is interior to C, then

$$v = \Sigma a_i u_i, \text{ with } a_i \geqq 0, \text{ and } u_i \in C.$$

Thus

$$x \cdot v = \Sigma a_i (x \cdot u_i)$$

which implies that there exists no x such that

$$x \cdot u \leqq 0 \text{ for all } u \in C, \text{ and } x \cdot v > 0.$$

$(1 \Rightarrow 2)$

Case 1: Condition 1 holds, and the subspace spanned by the elements of C does not contain v. Then v can be expressed as

$$v = \Sigma b_i u_i + v', \; u_i \in C, \; v' \neq 0,$$

where the b_i's are real coefficients, and

$$v' \cdot u = 0, \text{ for all } u \in C.$$

Then for $x = v'$, 2 is satisfied.

Case 2: Condition 1 holds, and the subspace spanned by the elements of C does contain v. The proof is by induction on the cardinality of C. For $|C| = 1$, C is a single vector u. By hypothesis v is in the subspace spanned by u, but is not interior to u. Thus

$$v = -au, \ a > 0$$

and 2 is satisfied for $x = -u$.

Now assume that $(1 \Rightarrow 2)$ for $|C| = k-1$. Let condition 1 hold for

$$C = (u_1, \ldots u_k);$$

then it certainly holds for

$$C_1 = (u_1, \ldots u_{k-1}).$$

By assumption since $|C_1| = k-1$, 2 holds for C_1. That is, there exists an x_1 such that

$$x_1 \cdot v > 0, \ x_1 \cdot u_i \leqq 0, \ 1 \leqq i \leqq k-1.$$

Now if $x_1 \cdot u_k \leqq 0$, then x_1 satisfies 2 for C as well, and we are done

If on the other hand $x_1 \cdot u_k > 0$, let

$$u'_i = u_i + b_i u_k, \text{ where } b_i = -\frac{x_1 \cdot u_i}{x_1 \cdot u_k} \geqq 0, \ 1 \leqq i \leqq k-1$$

and

$$v' = v + cu_k, \text{ where } c = -\frac{x_1 \cdot v}{x_1 \cdot u_k} < 0, \ 1 \leqq i \leqq k-1.$$

Suppose v' were interior to $C' = \{u'_1, \ldots u'_{k-1}\}$. Then we would have

$$v' = \Sigma a_i u'_i, \ a_i \geqq 0, \ i \leqq 1 \leqq k-1.$$

Then substituting for the u'_i's and v', and collecting terms, the above equation becomes

$$v = \sum_{1 \leqq i \leqq k-1} a_i u_i + \Big(\sum_{1 \leqq i \leqq k-1} b_i a_i - c\Big) u_k.$$

Letting a_k denote $\Sigma b_i a_i - c$, and noting that this is a positive quantity, we have

$$v = \Sigma a_i u_i, \; a_i \geqq 0, \; 1 \leqq i \leqq k$$

or equivalently, that v is interior to C, which contradicts the hypothesis. Thus v' is not interior to C', and since $|C'| = k\text{-}1$, there exists an x' such that

$$x' \cdot v' > 0, \; x' \cdot u'_i \leqq 0, \; 1 \leqq i \leqq k\text{-}1.$$

Now set

$$x = x' - \frac{x' \cdot u_k}{x_1 \cdot u_k} x_1.$$

We have for $1 \leqq i \leqq k\text{-}1$

$$\begin{aligned} x \cdot u_i &= x' \cdot u_i - \frac{x' \cdot u_k}{x_1 \cdot u_k} x_1 \cdot u_i \\ &= x' \cdot (u_i - \frac{x_1 \cdot u_i}{x_1 \cdot u_k} u_k) \\ &= x' \cdot u'_i \leqq 0. \end{aligned}$$

We also have

$$\begin{aligned} x \cdot u_k &= x' \cdot u_k - \frac{x' \cdot u_k}{x_1 \cdot u_k} x_1 \cdot u_k \\ &= 0 \end{aligned}$$

and

$$\begin{aligned} x \cdot v &= x' \cdot v - \frac{x' \cdot u_k}{x_1 \cdot u_k} x_1 \cdot v \\ &= x' \cdot (v - \frac{x_1 \cdot v}{x_1 \cdot u_k} u_k) \\ &= x' \cdot v'_i > 0. \end{aligned}$$

Thus x satisfies 2 for C.

Acknowledgement

We are grateful to Martin H. Schultz for many interesting discussions, and for pointing out that every DLO problem can in principle be converted to a linear programming problem.

Endnote

Theorem 2 is equivalent to the statement that if s_j and s_k are extreme points of a convex polytope such that s_k is a polytopal neighbor of s_j, then s_j is also a polytopal neighbor of s_k. The truth of this statement is evident from geometrical considerations. We have provided an algebraic, constructive proof to Theorem 2, which is also used in the proof of the following corollary.

References

1. N. Agin, Optimum Seeking with Branch and Bound, Management Science 13 (4), 1966.

2. F. Bock, An Algorithm for Solving "Traveling-Salesman" and Related Network Optimization Problems, presented at the 14th National Meeting of ORSA, 1958.

3. R. Bellman, Dynamic Programming, Princeton University Press, 1957.

4. M. Bellmore and G. Nemhauser, The Traveling Salesman Problem: A Survey, Operations Research 16 (3):538-558, 1968.

5. N. Burkov and S. Lovetskii, Methods for the Solution of Extremal Problems of Combinatorial Type, Aut Remot RR:1785-1806, 1968.

6. G. Croes, A Method for Solving Traveling-Salesman Problems, Operations Research 6 (6):792-812, 1958.

7. G. Dantzig, R. Fulkerson, and S. Johnson, Solution of a Large-Scale Traveling-Salesman Problem, Operations Research 2 (4):393-410, 1954.

8. G. Dantzig, Maximization of a Linear Function of Variables Subject to Linear Constraints, Cowles Commission Monograph 13, Wiley and Sons: 339-347, 1951.

9. M. Flood, The Traveling-Salesman Problem, Operations Research 4 (1): 61-75, 1956.

10. S. Gass, An Illustrated Guide to Linear Programming, McGraw-Hill, 1970.

11. B. Grunbaum, Convex Polytopes, Interscience Publishers, 1967.

12. M. Held and R. Karp, A Dynamic Programming Approach to Sequencing Problems, J. Soc. Indust. Appl. Math. 10:196-209, 1962.

13. R. Karg and G. Thompson, A Heuristic Approach to Solving Traveling Salesman Problems, Management Science 10:225-248, 1964.

14. M. G. Kendall, A Course in the Geometry of n Dimensions, Hafner Publishing, 1961.

15. D. Knuth, The Art of Computer Programming 1:399-405, Addison Wesley, 1968.

16. M. Krone, Heuristic Programming Applied to Scheduling Problems, Princeton University PhD Thesis, September 1970.

17. J. Kruskal, On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem, Proc. American Math. Society 7 (2):48-50, 1956.

18. H. Kuhn and A. Tucker, Linear Inequalities and Related Systems, Princeton University Press, 1956.

19. S. Lin, Computer Solutions of the Traveling Salesman Problem, Bell System Tech Journal 44:2245-2269, 1965.

20. J. Little, K. Murty, D. Sweeney, C. Karel, An Algorithm for the Traveling Salesman Problem, Operations Research 13:972-989, 1963.

21. C. L. Liu, Introduction to Combinatorial Mathematics, McGraw-Hill, 1968.

22. O. Ore, Theory of Graphs, American Mathematical Society, 1962.

23. S. Reiter and G. Sherman, Discrete Optimizing, J. Soc. Indust. Appl. Math. 13 (3):864-889, 1965.

24. S. Roberts and B. Flores, An Engineering Approach to the Traveling Salesman Problem, Management Science 13 (3), 1966.

25. S. L. Savage, The Solution of Discrete Linear Optimization Problems by Neighborhood Search Techniques, Yale University PhD Thesis, March 1973.

26. M. Simonnard, Linear Programming, Prentice-Hall, 1966.

27. K. Steiglitz, P. Weiner, D. Kleitman, The Design of Minimal Cost Survivable Networks, Technical Memorandum TM-105, National Resource Analysis Center, Systems Evaluation Division, 1969.

28. P. Weiner, S. L. Savage, A. Bagchi, Neighborhood Search Algorithms for Finding Optimal Traveling Salesman Tours Must Be Inefficient, Proceedings of the Fifth Annual ACM Symposium on Theory of Computing, April 1973.