On Solving Combinatorial Optimization Problems

Jian Yang
Department of Industrial and Manufacturing Engineering
New Jersey Institute of Technology
Newark, NJ 07102
Email: yang@adm.njit.edu

May 2002

Abstract

We present a new viewpoint on how some combinatorial optimization problems are solved. When applying this viewpoint to the NP-equivalent traveling salesman problem (TSP), we naturally arrive to a conjecture that is closely related to the polynomial-time insolvability of TSP, and hence to the $P-NP$ conjecture. Our attempt to prove the conjecture has not been successful so far. However, the byproducts are quite interesting themselves.

Keywords: Combinatorial Optimization Problem; Algorithm; Complexity; NP-Completeness; Traveling Salesman Problem.

1 Introduction

Traditionally, people view a combinatorial optimization problem (COP) as a mapping from a problem instance to one of its optimal solutions among the many feasible solutions. For example, TSP is expressed as a problem seeking the least costly tour when given information about the underlying cities. We feel that this view is not good enough to help us understand why one single algorithm for a particular COP is able to respond to the possibly infinite varieties of instances for that COP and always reach the correct answer.

To our understanding, a feasible solution of a COP is a problem-wide concept instead of an instance-wide concept and the relationship of an instance to its optimal solution is less of "corresponds to" than of "belongs to": For example, for the traveling salesman problem (TSP) with $n+1$ cities, there are $n!$ feasible solutions, namely, the $n!$ sequences of visiting the cities. An instance merely expresses the information on pairwise distances of the cities and it belongs to the set of instances that share the same optimal solution with it. Later on, for a given feasible solution s, we will call the set containing every instance that has s as one of its optimal solutions as the S-set of s.

With the above understanding of a COP, we can give a problem-wide description of an algorithm for the COP. An algorithm corresponds to a history-dependent sequence of divisions in the instance space: It first divides the instance space into two halves, say the left half and the right half. Then, it divides the left half into the left-left half and the left-right half, and divides the right half into the right-left half and the right-right half. And, this process goes on. The algorithm solves the COP when the process goes on both accurately and long enough so that every set resulted from its history of divisions is entirely inside one S-set. In other words, an algorithm for a COP is just a binary tree with every node being a set of instances, with every branching being a division of the set that is the father node into two sets that are the two children nodes, with its root being the entire instance space, and with each of its leaves being entirely inside one of the S-sets. Hence, given an instance, the algorithm executes the sequence of divisions that define the leaf to which the instance belongs and then reports the feasible solution whose S-set contains this leaf as the optimal solution for this instance.

For a variety of COPs, which we will later call RILCOPs, we further realize that each of their S-sets, when being mildly extrapolated, is merely a polyhedron. For these COPs, observation and intuition tempt us to guess that we need to look among only algorithms that divide the instance space by hyper-planes for efficient algorithms for such a COP. among only those which divide the instance space by hyper-planes. Later on, we will call these algorithms SIMPLE. For an RILCOP to be ever solvable by a SIMPLE algorithm in polynomial time, every of its S-set has to satisfy a certain condition. We conjecture that any S-set of TSP, one of the RILCOPs, does not satisfy the condition and therefore TSP cannot be solved by any SIMPLE algorithm in polynomial time. Since TSP is not only NP-hard, but also NP-easy [2], this conjecture serves as a sufficient condition for the well-known and puzzling conjecture $P \neq NP$ when SIMPLE algorithms are all that are needed to solve TSP.

Our effort in cracking the conjecture is not successful. However, along the way, we have found the following results that are previously unreported: Every S-set of TSP has an exponential number of facets; Every S-set of the assignment problem (AP), another RILCOP, has an exponential number of facets as well; (so the number of facets of an RILCOP's typical S-set is not a determining factor for the problem's polynomial-time insolvability); and TSP has only a number of faces whose logarithm is polynomial. Even though our conjecture has not been proved, we believe our viewpoint on COP and how it is solved is potentially a step closer to the final unraveling of the $P - NP$ conjecture.

We organize the remaining paper as follows: In Section 2, we define RILCOPs and SIMPLE algorithms. In Section 3, we elaborate on how a SIMPLE algorithm solves an RILCOP. In Section 4, we gives the conjectures about an S-set of TSP and show some related results. In Section 5, we show a result about an S-set of AP. We conclude the paper in Section 6.

2 RILCOPs and SIMPLE Algorithms

Consider a COP which satisfies:

1) Given a certain size parameter n, its instance space can be represented by a fixed-dimensional space of rational numbers and all its instances share a common finite set of feasible solutions;

2) Under a feasible solution, there is an objective value function of the instance which is linear. And, given an instance, the COP seeks a feasible solution under which the objective value function at the given instance is minimized.

The shortest path problem (SP), the minimal spanning tree problem (MST), TSP, and AP all belong to this type of rational-input linear COPs (RILCOPs).

For instance, TSP of size 4 can have each of its instances be represented by a 12-component rational vector:

$$c \equiv (c_{01}, c_{02}, c_{03}, c_{10}, c_{12}, c_{13}, c_{20}, c_{21}, c_{23}, c_{30}, c_{31}, c_{32})^T.$$

Each instance c has the same six feasible solutions:

$$\pi_0 = (1, 2, 3) = \text{ tour } 0 \to 1 \to 2 \to 3 \to 0,$$

$$\pi_1 = (1,3,2) = \text{ tour } 0 \to 1 \to 3 \to 2 \to 0,$$

$$\pi_2 = (2,1,3) = \text{ tour } 0 \to 2 \to 1 \to 3 \to 0,$$

$$\pi_3 = (2,3,1) = \text{ tour } 0 \to 2 \to 3 \to 1 \to 0,$$

$$\pi_4 = (3,1,2) = \text{ tour } 0 \to 3 \to 1 \to 2 \to 0,$$

$$\pi_5 = (3,2,1) = \text{ tour } 0 \to 3 \to 2 \to 1 \to 0.$$

With a chosen feasible solution π, the objective function $z(c,\pi)$ is linear on c. For example, we have

$$z(c,\pi_0) \equiv c_{01} + c_{12} + c_{23} + c_{30}.$$

For an RILCOP P, we denote its entire instance space (set) by $I_P()$. Given a feasible solution s, we call the set containing every instance that has s as one of its optimal solutions the S-set of s and denote it by $I_P^S(s)$. This set exactly contains all instances that satisfy the linear inequalities which specify the minimization of the objective function with the current s among all feasible solutions. Throughout, it shall not be necessary to distinguish between a feasible solution s and the S-set $I_P^S(s)$ it corresponds to. For our example, the S-set of feasible solution π_0 is the set of all c's that satisfy

$$z(c,\pi_0) \equiv c_{01} + c_{12} + c_{23} + c_{30} \le z(c,\pi_1) \equiv c_{01} + c_{13} + c_{32} + c_{20},$$

$$z(c,\pi_0) \equiv c_{01} + c_{12} + c_{23} + c_{30} \le z(c,\pi_2) \equiv c_{02} + c_{21} + c_{13} + c_{30},$$

$$z(c,\pi_0) \equiv c_{01} + c_{12} + c_{23} + c_{30} \le z(c,\pi_3) \equiv c_{02} + c_{23} + c_{31} + c_{10},$$

$$z(c,\pi_0) \equiv c_{01} + c_{12} + c_{23} + c_{30} \le z(c,\pi_4) \equiv c_{03} + c_{31} + c_{12} + c_{20},$$

$$z(c,\pi_0) \equiv c_{01} + c_{12} + c_{23} + c_{30} \le z(c,\pi_5) \equiv c_{03} + c_{32} + c_{21} + c_{10}.$$

For TSP, as for other RILCOPs, there does not seem to be any need to go through operations other than the four arithmetic operations involving the instance: ADDITION, SUBTRACTION, MULTIPLICATION by constant rationals, and DIVISION by constant rationals, plus the COMPARISON operation of the type "$\le$" or "$\ge$" between two linear functionals of the instance, and all the OTHER operations that do not involve the instance. For a better understanding of the OTHER operations, note that in the Hungarian method for solving AP, the OTHER operations include solving an entire unweighted maximal matching problem. At a certain stage of the Hungarian method, the 0-1 information indicating which unweighted maximal matching problem is to be solved is indeed determined by the given instance, but the solving itself does not involve doing the first five operations on it. When an algorithm has only the above six operations, we call it a SIMPLE algorithm.

In essence, a SIMPLE algorithm is an algorithm which never breaks an instance by non-arithmetic operations and always does linear operations on it. Dijkstra's algorithms for SP and MST [1], Kruskal's algorithm for the latter problem [5], the Hungarian method for AP [6] are all examples of SIMPLE algorithms.

As a counter example, the linear programming problem (LP) can be viewed as a COP, but not an RILCOP. And, the simplex method is not a SIMPLE algorithm. A standard-form LP with m rows and n columns ($n \geq m$) can be represented by a rational vector with $n + mn + m$ components:

$$e = (c_1, ..., c_n, A_{11}, ..., A_{1n}, ..., A_{m1}, ..., A_{mn}, b_1, ..., b_m),$$

where it is understood that c_j's are the objective coefficients, A_{ij}'s are the constraint coefficients, and b_i's are the right-hand side coefficients. Let $\pi_1, ..., \pi_{n!/(m!(n-m)!)}$ be all the $n!/(m!(n-m)!)$ choices of choosing m columns among n of them and thus, the representations of all possible bases. We let the objective function $z(e, \pi)$ be defined as

$$z(e, \pi) \equiv X^T(e, \pi)c = X_1(e, \pi)c_1 + \cdots + X_n(e, \pi)c_n$$

with $X(e, \pi)$ being the basic feasible solution for basis π under instance e if π offers a feasible basis for e and $+\infty$ otherwise. Then, LP can be viewed as a COP with the objective of finding the π with the least $z(e, \pi)$ for a given e among all $n!/(m!(n-m)!)$ possible choices of π's. But, for a fixed π, $z(e, \pi)$ is not linear in e as $X(e, \pi)$ depends on e. So, LP is not an RILCOP. The simplex method is a non-SIMPLE algorithm either: It has the occasion to do divisions with both operands being components of the instance. For RILCOPs, we have not seen any non-SIMPLE algorithms yet.

3 Decision Trees

Running a SIMPLE algorithm A on an instance of an RILCOP P provokes a sequence of computer operations that is instance-dependent. When run by the algorithm, all instances in the entire instance set $I_P()$ experience the same sequence of operations up to the first(-level) COMPARISON in A. This first COMPARISON on the other hand divides $I_P()$ into two subsets $I_{P,A}(\mathrm{y})$ and $I_{P,A}(\mathrm{n})$, with the former containing all instances in $I_P()$ that answer *yes* to the first COMPARISON and the latter containing all instances that answer *no* to that COMPARISON. All instances in $I_{P,A}(\mathrm{y})$ are in turn divided into $I_{P,A}(\mathrm{yy})$ and $I_{P,A}(\mathrm{yn})$ by the second-level COMPARISON in the algorithm, which is commonly experienced by all instances in $I_{P,A}(\mathrm{y})$, and all instances in $I_{P,A}(\mathrm{n})$ are divided into $I_{P,A}(\mathrm{ny})$ and $I_{P,A}(\mathrm{nn})$ by another second-level COMPARISON in the algorithm, which is commonly experienced by all instances in $I_{P,A}(\mathrm{n})$. Actually, given a y-n sequence θ, when $I_{P,A}(\theta)$ exists, the algorithm must let all instances in $I_{P,A}(\theta)$ experience another common COMPARISON and hence divide $I_{P,A}(\theta)$ into yet another two subsets $I_{P,A}(\theta\mathrm{y})$ and $I_{P,A}(\theta\mathrm{n})$, unless $I_{P,A}(\theta)$ is already entirely inside the S-set of one of P's feasible solutions, say s. When the latter is true, the algorithm will report s as the optimal solution to any instance in $I_{P,A}(\theta)$.

From the above, every SIMPLE algorithm A for an RILCOP P corresponds to a decision tree (DT), say DT$[P, A]$, with every node in this tree being a subset of the entire instance space. The two children of a non-leaf node constitute a partition of their father according to the y-n answer to a COMPARISON that is commonly experienced by every instance in the father. In addition, the root is exactly the entire instance space and each leaf is entirely inside the S-set of one of the RILCOP's feasible solutions. As an example, we give the decision-tree representation of Dijkstra's algorithm (DA) for finding SP from city 1 to cities 2 and 3:

For every instance in $I_{SP}()$: do COMPARISON $c_{12} \le c_{13}$?

  For every instance in $I_{SP,DA}(\mathrm{y})$: do COMPARISON $c_{13} \le c_{12} + c_{23}$?

    For every instance in $I_{SP,DA}(\mathrm{yy})$: declare $\{\{1, 2\}, \{1, 3\}\}$ as the optimal solution

    For every instance in $I_{SP,DA}(\mathrm{yn})$: declare $\{\{1, 2\}, \{1, 2, 3\}\}$ as the optimal solution

  For every instance in $I_{SP,DA}(\mathrm{n})$: do COMPARISON $c_{12} \le c_{13} + c_{32}$?

    For every instance in $I_{SP,DA}(\mathrm{ny})$: declare $\{\{1, 2\}, \{1, 3\}\}$ as the optimal solution

    For every instance in $I_{SP,DA}(\mathrm{nn})$: declare $\{\{1, 3, 2\}, \{1, 3\}\}$ as the optimal solution.

In the above, we have used $\{P_{12}, P_{13}\}$ to represent a feasible solution for the 3-city SP, where P_{1i} is the ordered sequence of cities on a path from city 1 to city i.

Since every non-leaf node in DT$[P, A]$ corresponds to a COMPARISON, there is also a tree of COMPARISONs corresponding to DT$[P, A]$, which we denote as CDT$[A]$. This CDT$[A]$ tree is independent of the problem P itself, even though the design of A has the structure of the S-sets of P in mind. In CDT$[A]$, a COMPARISON that is a non-leaf node has two children, the y-child and the n-child. The y-child is the COMPARISON to be conducted on an instance that has just answered *yes* to the father COMPARISON and the n-child is the COMPARISON to be conducted on an instance that has just answered *no* to the father COMPARISON. When we shrink every node in both DT$[P, A]$ and CDT$[A]$ into a point, CDT$[A]$ is made from DT$[P, A]$ by pruning all its leaves. Now, we can say the following: Every SIMPLE algorithm A corresponds to a tree of COMPARISONs CDT$[A]$. When apply CDT$[A]$ to an RILCOP P, we get the tree of polyhedra DT$[P, A]$. We say that A solves P when every leaf in DT$[P, A]$ is entirely inside one of P's S-sets.

If we extrapolate $I_P()$ to the corresponding real space, each of P's S-sets becomes a polyhedron containing all points (instances) satisfying the linear inequalities that define the optimality of s. Moreover, when ties are broken arbitrarily, every COMPARISON, being of the linear inequality type, is equivalent to using a hyper-plane to separate points on its two sides, and every node in DT$[P, A]$ becomes a polyhedron containing all points satisfying the linear inequalities that reflect the y-n answers to all the COMPARISONs commonly experienced by instances in this node.

We can take the intersection of any node in DT$[P, A]$ with a particular S-set $I_P^S(s)$ of P and the resulting new node is still a polyhedron if it is not empty. All the non-empty new nodes resulted from intersecting $I_P^S(s)$ with all nodes in DT$[P, A]$ form a binary tree as well, and we may dub this tree as the sub-DT of the algorithm A on solution s and denote it by sub-DT$[P, A, s]$. This sub-DT, being "embedded" in DT$[P, A]$, registers the complete information on what COMPARISONs are done and what answers are given for all the COMPARISONs on every instance in the S-set. Note that, all the leaves of the DT that are inside this S-set are unchanged and now they are all the leaves of the current sub-DT. Correspondingly, there is also the tree of COMPARISONs sub-CDT$[A, s]$ that specifies all the COMPARISONs in algorithm A whose corresponding hyper-planes intersect with $I_P^S(s)$.

Clearly, the height of CDT$[A]$ poses as a lower bound for the time complexity of A (Of course, when the number of inter-COMPARISON operations is bounded, the height of CDT$[A]$ is the same as the time complexity). For a given feasible solution s, this is even more true for sub-CDT$[P, A, s]$ since it is "embedded" in the former. At the same time, every facet in any leaf of sub-DT$[P, A, s]$ which is not originally in $I_P()$ must be the result of a COMPARISON. Therefore,

Proposition 1 *For an RILCOP P to have a polynomial-time SIMPLE algorithm, there must exist a tree of COMPARISONs, say CDT, with polynomial height, such that when it is applied to each of P's S-set $I_P^S(s)$, every leaf node of the resulting tree of polyhedra, say DT$[s]$, should have only a polynomial number of facets besides the facets already in P's entire instance set $I_P()$.*

4 Techniques

We let $(\mathbf{a}, b)_{E(L,G)}$ stand for the set of $\mathbf{c}$'s satisfying $\mathbf{a}^T\mathbf{c} = (\leq, \geq)b$. We use "$\wedge$"/"$\vee$" to represent set intersection/union, and "$\widetilde{\ }$" for set complement. Given finite index set I and pairs $(\mathbf{a}_i, b_i)$ for $i \in I$, we call $P = \bigwedge_{i\in I}(\mathbf{a}_i, b_i)_L$ an inequality system, or simply, a system. We say that two inequality systems $P^1 = \bigwedge_{i\in I^1}(\mathbf{a}_i^1, b_i^1)_L$ and $P^2 = \bigwedge_{i\in I^2}(\mathbf{a}_i^2, b_i^2)_L$ are separable when there exists pair $(\mathbf{a}, b)$ such that:

s1) $\bigwedge_{i\in I^1}(\mathbf{a}_i^1, b_i^1)_L \subset (\mathbf{a}, b)_L$ and $\bigwedge_{i\in I^2}(\mathbf{a}_i^2, b_i^2)_L \subset (\mathbf{a}, b)_G$;

s2) $(\bigwedge_{i\in I^1}(\mathbf{a}_i^1, b_i^1)_L) \wedge \widetilde{(\mathbf{a}, b)}_G \neq \emptyset$ and $(\bigwedge_{i\in I^2}(\mathbf{a}_i^2, b_i^2)_L) \wedge \widetilde{(\mathbf{a}, b)}_L \neq \emptyset$.

Given pair $(\mathbf{a}, b)$, and separable systems P^1 and P^2, we say that $(\mathbf{a}, b)$ separates P^1 and P^2 when the three satisfy the above s1) and s2).

We say that pair $(\mathbf{a}, b)$ divides system $P = \bigwedge_{i\in I}(\mathbf{a}_i, b_i)_L$ if there are nonempty as well as proper subsets I^L and I^G of I satisfying $I^L \cup I^G = I$, such that:

d1) $(\mathbf{a}, b)_L \wedge (\bigwedge_{i\in I^L}(\mathbf{a}_i, b_i))_L \subset \bigwedge_{i\in I^G}(\mathbf{a}_i, b_i)_L$ and $(\mathbf{a}, b)_G \wedge (\bigwedge_{i\in I^G}(\mathbf{a}_i, b_i))_L \subset \bigwedge_{i\in I^L}(\mathbf{a}_i, b_i)_L$.

When the above is true, we also say that $(\mathbf{a}, b)$ divides P into $P^L = (\mathbf{a}, b)_L \wedge (\bigwedge_{i\in I^L}(\mathbf{a}_i, b_i))_L$ and $P^G = (\mathbf{a}, b)_G \wedge (\bigwedge_{i\in I^G}(\mathbf{a}_i, b_i))_L$.

From the proper-and-nonempty-ness of index sets I^L and I^G in the definition of a division, it is easy to see that, as long as pair $(\mathbf{a}, b)$ divides system $P = \bigwedge_{i\in I}(\mathbf{a}_i, b_i)_L$, there will be $i \in I$, such that

$$(\mathbf{a}, b)_L \wedge (\bigwedge_{i'\in I\setminus\{i\}} (\mathbf{a}_{i'}, b_{i'})_L) \subset (\mathbf{a}_i, b_i)_L. \tag{1}$$

Theorem 1 *We will have $\bigwedge_{i\in I}(\mathbf{a}_i, 0)_L \subset (\mathbf{a}, b)_L$, if and only if, $b \geq 0$ and there exist non-negative constants $\{\alpha_i\}_{i\in I}$ such that $\sum_{i\in I} \alpha_i \mathbf{a}_i = \mathbf{a}$.*

Proof: Since the *if* part is obvious, we concentrate on the *only if* part. Suppose $b < 0$. Then vector $\mathbf{0}$ satisfies $\mathbf{a}_i^T\mathbf{0} \leq 0$ and yet $\mathbf{a}^T\mathbf{0} > b$. Suppose $b \geq 0$, but there do not exist nonnegative constants $\{\alpha_i\}_{i\in I}$ satisfying $\sum_{i\in I} \alpha_i \mathbf{a}_i = \mathbf{a}$. Then, by Farkas' lemma, there exists vector $\mathbf{c}$ so that $\mathbf{a}_i^T\mathbf{c} \leq 0$ for any $i \in I$, and yet $\mathbf{a}^T\mathbf{c} > b$. ∎

Later on, we shall denote the set of all permutations of $\{1, ..., n\}$ by $\hat{\Pi}(n)$; and for any permutation $\pi \in \hat{\Pi}(n)$, we shall define the $n(n+1)$-component 0-1 vector $\hat{\mathbf{x}}(\pi)$ as the one whose component positions are labeled as $(01), \cdots, (0n)$, $(10), (12), \cdots, (1n)$, $\cdots \ \cdots$, $(n0), \cdots, (n, n-1)$, and whose components at the $(0\pi(1))$, $(\pi(1)\pi(2))$, $\cdots$, $(\pi(n-1)\pi(n))$, and $(\pi(n)0)$ positions are 1's while components at the remaining $(n+1)(n-1)$ positions are 0's. Then, for any $\pi \in \hat{\Pi}(n)$, we shall define the solution set $\hat{I}(\pi)$ by

$$\hat{I}(\pi) = \bigwedge_{\pi'\in\hat{\Pi}(n)\setminus\{\pi\}} (\bar{\mathbf{x}}(\pi) - \bar{\mathbf{x}}(\pi'), 0)_L.$$

For $n = 3$, the six $\bar{\mathbf{x}}(\pi)$'s are

$$\begin{array}{rcccccccccccccl}
\text{arc index} & & 01 & 02 & 03 & 10 & 12 & 13 & 20 & 21 & 23 & 30 & 31 & 32 & \\
(\bar{\mathbf{x}}(123))^T = & (& 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 &), \\
(\bar{\mathbf{x}}(132))^T = & (& 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 &), \\
(\bar{\mathbf{x}}(213))^T = & (& 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 &), \\
(\bar{\mathbf{x}}(231))^T = & (& 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 &), \\
(\bar{\mathbf{x}}(312))^T = & (& 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 &), \\
(\bar{\mathbf{x}}(321))^T = & (& 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 &).
\end{array} \tag{2}$$

Proposition 2 *For $\pi_1, \pi_2 \in \hat{\Pi}(n)$ and $\pi_1 \neq \pi_2$, $\hat{I}(\pi_1)$ and $\hat{I}(\pi_2)$ are separable and any pair that separates them must be of the form $(a\bar{\mathbf{x}}(\pi_1) - a\bar{\mathbf{x}}(\pi_2), 0)$, where a is a nonzero constant.*

We conjecture that, the linear question (pair) that separates two solution sets $\hat{I}(\pi)$ and $\hat{I}(\pi')$ does not help divide a third solution set $\hat{I}(\pi'')$. Formally, we phrase our conjecture as in the following.

Conjecture 1 *For $\pi_1, \pi_2, \pi_3, \pi_4 \in \hat{\Pi}(n)$ that are all different from each other, we have $(\bar{\mathbf{x}}(\pi_3) - \bar{\mathbf{x}}(\pi_4), 0)_L \wedge (\bigwedge_{\pi \in \hat{\Pi}(n) \setminus \{\pi_1, \pi_2\}} (\bar{\mathbf{x}}(\pi_1) - \bar{\mathbf{x}}(\pi), 0)_L) \not\subset (\bar{\mathbf{x}}(\pi_1) - \bar{\mathbf{x}}(\pi_2), 0)_L$, which, according to Theorem 1, is equivalent to that, for any nonnegative constants α_0 and $\{\alpha_\pi \mid \pi \in \hat{\Pi}(n) \setminus \{\pi_1, \pi_2\}\}$, we have*

$$\alpha_0(\bar{\mathbf{x}}(\pi_3) - \bar{\mathbf{x}}(\pi_4)) + \sum_{\pi \in \hat{\Pi}(n) \setminus \{\pi_1, \pi_2\}} \alpha_\pi (\bar{\mathbf{x}}(\pi_1) - \bar{\mathbf{x}}(\pi)) \neq \bar{\mathbf{x}}(\pi_1) - \bar{\mathbf{x}}(\pi_2).$$

In the above, the pair $(\bar{\mathbf{x}}(\pi_3) - \bar{\mathbf{x}}(\pi_4), 0)$ separates solution sets $\hat{I}(\pi_3)$ and $\hat{I}(\pi_4)$. From the argument around (1), the validity of the conjecture will guarantee the impossibility of dividing $\hat{I}(\pi_1)$ using this pair.

For $n = 3$, $\pi_1 = (123)$, $\pi_2 = (312)$, $\pi_3 = (132)$, and $\pi_4 = (231)$, the conjecture means that knowing

$$\mathbf{c} \in \begin{cases} (\bar{\mathbf{x}}(132) - \bar{\mathbf{x}}(231), 0)_L : & c_{01} + c_{13} + c_{32} + c_{20} \leq c_{02} + c_{23} + c_{31} + c_{10}, \\ (\bar{\mathbf{x}}(123) - \bar{\mathbf{x}}(132), 0)_L : & c_{12} + c_{23} + c_{30} \leq c_{13} + c_{32} + c_{20}, \\ (\bar{\mathbf{x}}(123) - \bar{\mathbf{x}}(213), 0)_L : & c_{01} + c_{12} + c_{23} \leq c_{02} + c_{21} + c_{13}, \\ (\bar{\mathbf{x}}(123) - \bar{\mathbf{x}}(231), 0)_L : & c_{01} + c_{12} + c_{30} \leq c_{02} + c_{31} + c_{10}, \\ (\bar{\mathbf{x}}(123) - \bar{\mathbf{x}}(321), 0)_L : & c_{01} + c_{12} + c_{23} + c_{30} \leq c_{03} + c_{32} + c_{21} + c_{10}, \end{cases} \tag{3}$$

we will not be able to derive that

$$\mathbf{c} \in (\bar{\mathbf{x}}(123) - \bar{\mathbf{x}}(312), 0)_L : \quad c_{01} + c_{23} + c_{30} \leq c_{03} + c_{31} + c_{20}. \tag{4}$$

We also conjecture that, any polyhedral cone that poses as a candidate for a leaf completely inside a solution set must have an exponential number of faces. For such a cone $\hat{C}$ with f faces, we may let $n(n+1)$-dimensional vectors $\mathbf{y}(1), ..., \mathbf{y}(f)$ be its defining vectors, which means that

$$\hat{C} = \bigwedge_{i=1}^{f} (\mathbf{y}(i), 0)_L. \tag{5}$$

By Theorem 1, for $\hat{C} \subset \hat{I}(\pi)$ to happen for some $\pi \in \hat{\Pi}(n)$, there must be $f(n! - 1)$ positive constants $\alpha(\pi', i)$, where $\pi' \in \hat{\Pi}(n) \setminus \{\pi\}$ and $i = 1, ..., f$, such that

$$\bar{\mathbf{x}}(\pi) - \bar{\mathbf{x}}(\pi') = \sum_{i=1}^{f} \alpha(\pi', i) \mathbf{y}(i), \quad \forall \pi' \in \hat{\Pi}(n) \setminus \{\pi\}. \tag{6}$$

For convenience, we use $\bar{\pi}_0(n)$ to denote the identity permutation $(12\cdots n)$. We may present our conjecture formally as in the following:

Conjecture 2 *Suppose for any* $n = 1, 2, ...,$ *there are some* $f(n)$ *number of* $n(n+1)$*-dimensional vectors* $\mathbf{y}(n,1), ..., \mathbf{y}(n, f(n))$ *such that* $\bigwedge_{i=1}^{f(n)}(\mathbf{y}(n,i), 0)_L \neq \emptyset$*, as well as* $f(n)(n!-1)$ *positive constants* $\alpha(\pi', i)$*, where* $\pi' \in \hat{\Pi}(n) \setminus \{\bar{\pi}_0(n)\}$ *and* $i = 1, ..., f(n)$*, such that*

$$\bar{\mathbf{x}}(\bar{\pi}_0(n)) - \bar{\mathbf{x}}(\pi') = \sum_{i=1}^{f(n)} \alpha(\pi', i)\mathbf{y}(n,i), \quad \forall \pi' \in \hat{\Pi}(n) \setminus \{\bar{\pi}_0(n)\}.$$

Then, $f(n)$ *must be exponential. That is, for any* $a > 0$*, we must have* $\limsup_{n\to+\infty} f(n)/n^a = +\infty$*.*

5 The Traveling Salesman Problem

We concentrate on the $(n+1)$-city asymmetric traveling salesman problem. With the understanding that we are always dealing with the $(n+1)$-city problem, we shall omit mentioning n whenever possible. Hence, we denote the problem TSP instead of TSP$[n]$, and so on and so forth. For TSP, every instance is an $n(n+1)$-component distance vector $c \equiv (c_{01}, ..., c_{0n}, ..., c_{n0}, ..., c_{n,n-1})^T$ and $I_{TSP}()$ is the $n(n+1)$-dimensional real space $R^{n(n+1)}$. Every feasible solution of TSP is a permutation of $\{1, ..., n\}$, say $\pi = (\pi(1), ..., \pi(n))$, with $\pi(i)$ standing for the ith city being visited in the corresponding tour after city 0. We denote the set of all permutations of $\{1, ..., n\}$ by Π. Also, we denote the identity permutation of Π by 1_Π. For a given $\pi \in \Pi$, the conventional 0-1 representation of the corresponding solution

$$X(\pi) \equiv (X_{01}(\pi), ..., X_{0n}(\pi), ..., X_{n0}(\pi), ..., X_{n,n-1}(\pi))^T$$

is reached by letting $X_{ij}(\pi) = 1$ when $i = 0, j = \pi(1)$, or $i = \pi(k), j = \pi(k+1)$ for some $k = 1, ..., n-1$, or $i = \pi(n), j = 0$; and letting $X_{ij}(\pi) = 0$ otherwise. The objective function at feasible solution π is

$$z_{TSP}(c, \pi) \equiv X^T(\pi)c = c_{0\pi(1)} + c_{\pi(1)\pi(2)} + \cdots + c_{\pi(n-1)\pi(n)} + c_{\pi(n)0}.$$

The S-set $I_{TSP}^S(\pi)$ is defined by:

$$I_{TSP}^S(\pi) \equiv \{c \in R^{n(n+1)} \mid X^T(\pi)c \leq X^T(\pi')c, \forall \pi' \in \Pi \setminus \{\pi\}\}.$$

Thus, every $I_{TSP}^S(\pi)$ is a polyhedron defined by $n! - 1$ linear inequalities.

Since TSP is an NP-equivalent problem [2][4], one can prove $P \neq NP$ by showing that no polynomial-time algorithm can solve TSP. Because TSP belongs to those RILCOPs to which SIMPLE algorithms are of extreme importance, proving the weaker property that TSP cannot be solved by any polynomial-time SIMPLE algorithm will already do a great deal to the ultimate proof of the $P - NP$ conjecture. In the following, we give a conjecture that serves as a sufficient condition for this weaker property. Since Proposition 1 gives the necessary condition for an RILCOP to be solvable by any polynomial-time SIMPLE algorithm, our conjecture is basically the inverse of Proposition 1 being applied to $I^S_{TSP}(1_\Pi)$. The apparent symmetry among all $I^S_{TSP}(\pi)$'s makes it sufficient to examine one S-set of TSP if we are to study every of its S-sets individually.

Conjecture 3 *There does not exist a polynomial-height tree of COMPARISONs, say CDT, such that every leaf node in the tree of polyhedra resulted from applying CDT to $I^S_{TSP}(1_\Pi)$ has only a polynomial number of facets.*

We do not mention the facets of $I_{TSP}()$ because there are no such facets.

For convenience, we call a number log-polynomial if its logarithm is polynomial and call a number log-exponential if its logarithm is exponential. For example, both n and 2^n are log-polynomial, while 2^{2^n} is log-exponential. We have the following three facts: 1) As the product of two polynomials are still polynomial, the sum of a log-polynomial number of log-polynomial numbers is still a log-polynomial number. 2) A tree of polynomial height has only a log-polynomial number of leaves. 3) $I^S_{TSP}(1_\Pi)$ has a log-polynomial number of facets itself. From the above three facts, we can deduce that the following is a stronger conjecture than Conjecture 3.

Conjecture 4 *$I^S_{TSP}(1_\Pi)$ cannot be divided into a log-polynomial number of smaller polyhedra with each having only a polynomial number of facets.*

Both of the above conjectures are concerned with the geometric property of the $I^S_{TSP}(1_\Pi)$ polyhedron. The more we learn about this polyhedron, the closer we are from proving or disproving the conjectures. Even though we are not successful in proving or disproving either of the conjectures, we do gain some knowledge about the polyhedron. In the rest of this section, we show two results about the polyhedron and explain their implication to the conjectures.

Theorem 2 *For every $\pi \in \Pi \setminus \{1_\Pi\}$, inequality $X^T(1_\Pi)c \leq X^T(\pi)c$ defines a distinct facet of $I^S_{TSP}(1_\Pi)$. Hence, the polyhedron has $n! - 1$ facets.*

Proof: Given $\pi \in \Pi \setminus \{1_\Pi\}$, we can find an instance c such that

$$X^T(1_\Pi)c \leq X^T(\pi')c \quad \forall \pi' \in \Pi \setminus \{1_\Pi, \pi\},$$

and yet

$$X^T(1_\pi)c > X^T(\pi)c.$$

This c is as follows:

$$c_{01} = c_{12} = \cdots = c_{n-1,n} = c_{n0} = 0,$$

$$c_{ij} = -1 \quad \text{if } X_{ij}(\pi) = 1 \text{ and } X_{ij}(1_\Pi) = 0,$$

and

$$c_{ij} = n+1 \quad \text{if } X_{ij}(\pi) = X_{ij}(1_\Pi) = 0.$$

For $\pi \in \Pi \setminus \{1_\Pi\}$ and $\pi' \in \Pi \setminus \{1_\Pi, \pi\}$, any one of $X(1_\Pi)$, $X(\pi)$, and $X(\pi')$ differs from any other of them by at least two components. We can check that

$$X^T(1_\Pi)c = 0,\ \ X^T(\pi)c < 0,$$

and

$$X^T(\pi')c \geq 0, \quad \forall \pi' \in \Pi \setminus \{1_\Pi, \pi\}.$$

∎

The implication of Theorem 2 is that $I^S_{TSP}(1_\Pi)$ is indeed very complex. However, we will show in the next section that a typical S-set of the polynomial-time solvable AP has also an exponential number of facets. Hence, knowing that $I^S_{TSP}(1_\Pi)$ has an exponential number of facets is only a first step toward finally proving the conjectures.

Theorem 3 *$I^S_{TSP}(1_\Pi)$ has a log-polynomial number of faces.*

Proof: Every face in $I^S_{TSP}(1_\Pi)$ is the intersection of a few of its mutually independent facets. Since $I^S_{TSP}(1_\Pi)$ is only $n(n+1)$-dimensional after all, at one time, the number of mutually independent facets cannot be more than $n(n+1)$. Actually, according to Grotschel and Padberg [3], or according to the simple fact that $\forall \pi \in \Pi$,

$$X_{0n}(\pi) = 1 - X_{01}(\pi) - \cdots - X_{0,n-1}(\pi),$$

$$\cdots\cdots,$$

$$X_{n-1,n}(\pi) = 1 - X_{n-1,0}(\pi) - X_{n-1,n-2}(\pi),$$

$$X_{n0}(\pi) = 1 - X_{10}(\pi) - \cdots - X_{n-1,0}(\pi),$$

$$\cdots\cdots,$$

$$X_{n,n-1}(\pi) = 1 - X_{1,n-1}(\pi) - \cdots - X_{n-2,n-1}(\pi),$$

$$X_{n-1,n-2}(\pi) = n - 1 - X_{01}(\pi) - \cdots - X_{0,n-1}(\pi) - \cdots - X_{n-1,0}(\pi) - \cdots - X_{n-1,n-3}(\pi),$$

the number of mutually independent facets at one time cannot exceed n^2-n-1. Therefore, the number of faces in $I^S_{TSP}(1_\Pi)$ is no more than

$$n!-1+(n!-1)(n!-2)/2+\cdots+(n!-1)!/((n^2-n-1)!(n!-n^2+n)!),$$

which is a log-polynomial number. ∎

We would have proved Conjecture 4 had we shown that $I^S_{TSP}(1_\Pi)$ has a log-exponential number of faces. Here is why: Suppose $I^S_{TSP}(1_\Pi)$ can be divided into a log-polynomial number of small polyhedra with each having only a polynomial number of facets. Then each of these small polyhedra should have only a log-polynomial number of faces. And, the total number of faces all these small polyhedra have is still log-polynomial. On the other hand, we can show in Lemma 1 that, when we partition a polyhedron into two smaller polyhedra, the total number of faces the two smaller polyhedra have is at least the number of faces of the original polyhedron. Hence, if $I^S_{TSP}(1_\Pi)$ has a log-exponential number of faces, the total number of faces of all the small polyhedra will also be log-exponential instead of log-polynomial. Therefore, $I^S_{TSP}(1_\Pi)$ cannot be partitioned in that fashion and Conjecture 4 would have been proved.

Lemma 1 *When polyhedron H is partitioned by a hyper-plane into two polyhedra H_L and H_G, the total number of faces of the two smaller polyhedra is at least the number of faces H has.*

Proof: Suppose one of H's faces f is partitioned by the hyper-plane into two parts f_L and f_G (one of them is possibly $\emptyset$) with $f_L = f \cap H_L$ and $f_G = f \cap H_G$. Still, f_L is a face of H_L and f_G is a face of H_G. If not for the fact that two different faces f and f' in H may yield the same f_L and f'_L or the same f_G and f'_G, we would have proved the lemma here. Now, without loss of generality, we consider the case where $f' \neq f$ and yet $f'_L = f_L$. Again, without loss of generality, we consider the two cases where $Dim(f') \geq Dim(f)+1$ and $Dim(f') = Dim(f)$, respectively. In the first case,

$$Dim(f'_G) = Dim(f') \geq Dim(f_L)+1.$$

So two different faces in H result in at least two different faces in H_L and H_G. In the second case, if

$$Dim(f_L) = Dim(f'_L) = Dim(f) = Dim(f'),$$

we must have $f = f'$, since no two different faces in the same polyhedron can have the same dimension and still one is in another. But f and f' are different, we must have

$$Dim(f_L) = Dim(f'_L) \leq Dim(f)-1 = Dim(f')-1,$$

which ensures that

$$f_G = f, \quad f'_G = f'.$$

Again, since f and f' are different faces of H, f_G and f'_G are also different faces of H_G. ∎

6 The Assignment Problem

The assignment problem with n pairs of assignments, AP[n] (we shall avoid mentioning n whenever possible later on), has R^{n^2} as its entire instance set $I_{AP}()$ and each of its instances is an n^2-component cost vector $c \equiv (c_{11}, ..., c_{1n}, ..., c_{n1}, ..., c_{nn})^T$. Each of AP's feasible solution is a permutation π of $\Pi[n]$. For a given π, the usual solution vector for AP is as follows:

$$Y(\pi) \equiv (Y_{11}(\pi), ..., Y_{1n}(\pi), ..., Y_{n1}(\pi), ..., Y_{nn}(\pi))^T$$

where $Y_{ij}(\pi) = 1$ when $\pi(i) = j$ and $Y_{ij}(\pi) = 0$ otherwise. The objective $z_{AP}(c, \pi)$ is defined as

$$z_{AP}(c, \pi) \equiv Y^T(\pi)c = c_{1\pi(1)} + \cdots + c_{n\pi(n)}.$$

The S-set $I^S_{AP}(\pi)$ is defined as

$$I^S_{AP}(\pi) \equiv \{c \in R^{n^2} \mid Y^T(\pi)c \le Y^T(\pi'), \forall \pi' \in \Pi[n] \setminus \{\pi\}\}.$$

Due to the apparent symmetry, we need only to study $I^S_{AP}(1_{\Pi[n]})$ to understand each S-set of AP individually. Unlike TSP, we can show that some of the inequalities in the definition of $I^S_{AP}(1_{\Pi[n]})$ can actually be derived from others. But still, the polyhedron has an exponential number of facets.

We note that every permutation π in $\Pi[n]$ can be partitioned into some R cyclic permutations $\sigma_1[n_1], ..., \sigma_R[n_R]$, where $n = n_1 + \cdots + n_R$. We denote this fact by $\pi \equiv \sigma_1[n_1] \cdots \sigma_R[n_R]$. Under this notation, we should understand that π also determines for every $r = 1, ..., R$, which n_r numbers in $\{1, ..., n\}$ are to be cyclically permuted by $\sigma_r[n_r]$ and among the $(n_r - 1)!$ possibilities, what cyclic permutation $\sigma_r[n_r]$ exactly is. Now, we can confirm that the fact

$$Y^T(1_{\Pi[n]})c \le Y^T(\sigma_1[n_1]1_{\Pi[n-n_1]})c$$

$$\text{AND } Y^T(1_{\Pi[n]})c \le Y^T(1_{\Pi[n_1]}\sigma_2[n_2]1_{\Pi[n-n_1-n_2]})c$$

$$\text{AND } \cdots\cdots$$

$$\text{AND } Y^T(1_{\Pi[n]})c \le Y^T(1_{\Pi[n-n_R]}\sigma_R[n_R])c$$

infer the fact

$$Y^T(1_{\Pi[n]})c \le Y^T(\sigma_1[n_1]\cdots\sigma_R[n_R])c \equiv Y^T(\pi)c.$$

Thus, only inequalities of the type $Y^T(1_{\Pi[n]})c \le Y^T(\sigma[m]1_{\Pi[n-m]})c$ for $m = 2, ..., n$ are candidates for $I_{AP}^S(1_{\Pi[n]})$'s facets.

To precisely describe a permutation π of the type $\sigma[m]1_{\Pi[n-m]}$, we need to know which m elements of $\{1, ..., n\}$ is cyclically permuted by $\sigma[m]$ and what the exact cyclic permutation it is. To this end, define permutation δ on $\{1, ..., n\}$ so that for $i = 1, ..., m$, $\delta(i)$ is the ith smallest member of $\{1, ..., n\}$ that should be permuted by $\sigma[m]$ and for $i = m + 1, ..., n$, $\delta(i)$ is the $(i - m)$th smallest member of $\{1, ..., n\}$ that should not be in $\sigma[m]$. Also, define permutation σ on $\{1, ..., n\}$ which acts on $\{1, ..., m\}$ as π acts on $\{\delta(1), ..., \delta(m)\}$ and acts on $\{m + 1, ..., n\}$ as an identity permutation. Now, we have $\pi = \delta\sigma\delta^{-1}$. Let $\Delta[m, n]$ be the set of all $n!/(m!(n-m)!)$ permutations that could be δ and let $\Sigma[m, n]$ be the set of all $(m-1)!$ permutations that could be σ if π is of the type $\sigma[m]1_{\Pi[n-m]}$. Then,

$$I_{AP}(1_{\Pi[n]}) = \{c \in R^{n^2} \mid Y^T(1_{\Pi[n]})c \le Y^T(\delta\sigma\delta^{-1})c,$$

$$\forall m = 2, ..., n, \delta \in \Delta[m, n], \sigma \in \Sigma[m, n]\}.$$

$I_{AP}(1_{\Pi[n]})$ should be invariant under the relabeling of cities. That is, if we define

$$\pi \times c \equiv (c_{\pi^{-1}(1)\pi^{-1}(1)}, ..., c_{\pi^{-1}(1)\pi^{-1}(n)}, ..., c_{\pi^{-1}(n)\pi^{-1}(1)}, ..., c_{\pi^{-1}(n)\pi^{-1}(n)})^T,$$

then

$$I_{AP}(1_{\Pi[n]}) = \pi \times I_{AP}(1_{\Pi[n]}) \equiv \{\pi \times c \mid c \in I_{AP}(1_{\Pi[n]}).$$

To verify this, we can check that

$$Y^T(\pi\pi_1\pi^{-1})(\pi \times c) = c_{\pi^{-1}(1)\pi_1\pi^{-1}(1)} + \cdots + c_{\pi^{-1}(n)\pi_1\pi^{-1}(n)}$$

$$= c_{1\pi_1(1)} + \cdots + c_{n\pi_1(n)} = Y^T(\pi_1)c.$$

And, since $\pi 1_{\Pi[n]} = 1_{\Pi[n]}\pi$, we have

$$Y^T(1_{\Pi[n]})(\pi \times c) = Y^T(1_{\Pi[n]})c.$$

Hence, the c's in $I_{AP}^S(1_{\Pi[n]})$ satisfies the same set of inequalities as the c's in $\pi \times I_{AP}^S(1_{\Pi[n]})$.

Moreover, the city-relabeling operation $\pi\times$ offers a symmetry of inequalities within $I_{AP}^S(1_{\Pi[n]})$. Define $1_{\Sigma[m,n]}$ to be a permutation on $\{1, ..., n\}$ such that $1_{\Sigma[m,n]}(1) = 2$, ..., $1_{\Sigma[m,n]}(m-1) = m$, and $1_{\Sigma[m,n]}(m) = 1$, and that $1_{\Sigma[m,n]}(k) = k, \forall k = m + 1, ..., n$. Given $\delta \in \Delta[m, n]$ and $\sigma \in \Sigma[m, n]$, we can always define permutation π such that $\pi(1) = \delta(1)$,

$\pi(2) = \delta\sigma(1), ...,$ and $\pi(m) = \delta\sigma^{m-1}(1)$, and that $\pi(k) = \delta(k)$ for $k = m+1, ..., n$. Then, we have $\pi 1_{\Sigma[m,n]} = \delta\sigma\delta^{-1}\pi$, and therefore

$$Y^T(\delta\sigma\delta^{-1})(\pi \times c) = Y^T(1_{\Sigma[m,n]})c.$$

Hence, if the inequality associated with $1_{\Sigma[m,n]}$ defines a facet for $I^S_{AP}(1_{\Pi[n]})$, then $\forall\delta \in \Delta[m,n]$ and $\forall\sigma \in \Sigma[m,n]$, the inequality associated with $\delta\sigma\delta^{-1}$ defines a facet for the polyhedron as well.

Theorem 4 *Every inequality of the type*

$$z_{AP}(c, 1_{\Pi[n]}) \leq z_{AP}(c, \delta\sigma\delta^{-1})$$

for $m = 2, ..., n, \delta \in \Delta[m,n], \sigma \in \Sigma[m,n]$ *is a facet of the* S*-set* $I_{AP}(1_{\Pi[n]})$. *The total number of facets is* $\sum_{m=2}^{n} n!/(m(n-m)!)$.

Proof: Given $m = 2, ..., n$, we are to prove that inequality

$$Y^T(1_{\Pi[n]})c \leq Y^T(1_{\Sigma[m,n]})c$$

cannot be deduced from all other $I_{AP}(1_{\Pi[n]})$-defining inequalities associated with $m' \neq m$. We study the following instance c:

$$c_{11} = \cdots = c_{nn} = 0,$$

$$c_{12} = c_{23} = \cdots = c_{m-1,m} = c_{m1} = -1,$$

and

$$c_{ij} = m, \quad \text{for all other } i, j.$$

In any cycle of length $m' \neq m$, at most $m-1$ of the $c_{12}, c_{23}, ..., c_{m-1,m}, c_{m1}$, and at least one c_{ij} of the third type in the above, are incurred. So all the $I_{AP}(1_{\Pi[n]})$-defining inequalities with $m' \neq m$ are satisfied for this instance. But we certainly have

$$Y^T(1_{\Pi[n]})c > Y^T(1_{\Sigma[m,n]})c.$$

∎

7 Concluding Remarks

In this paper, we have provided our viewpoint on how a COP is solved. With this viewpoint, we have identified a more special group of COPs as the RILCOPs, proposed the concept of the S-sets, and shown that an S-set of a RILCOP is a polyhedron. In addition, we have defined a group of algorithms as SIMPLE algorithms. We feel that the most natural algorithm to solve a RILCOP should be a SIMPLE algorithm. With the understanding that a SIMPLE algorithm merely divides the instance space of an RILCOP into small polyhedra in a certain fashion so that the final resulting polyhedra are all entirely inside some S-sets, we have found one necessary condition for a SIMPLE algorithm to be able to solve an RILCOP in polynomial time.

TSP, one of the RILCOPs, is an NP-equivalent problem, that is, the easiest NP-hard problem. That TSP cannot be solved in polynomial time by any SIMPLE algorithm will serve as a strong evidence for $P \neq NP$. Using the above necessary condition, we have conjectured two sufficient conditions for TSP not being solvable by any SIMPLE algorithm in polynomial time. The conjectures are all concerned with a polyhedron which is one of the S-sets of TSP, and thus have geometric flavors. So far, we have not succeeded in proving these conjectures. Nevertheless, we did achieve some understanding about that polyhedron and a similar polyhedron defined for AP.

Acknowledgements

The first author is supported, in part, by New Jersey Institute of Technology under Grant No. 4-21830 and the National Center for Transportation and Industrial Productivity under Grant No. 9-92518.

References

[1] Dijkstra, E.W. 1959. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, **1**, pp. 269-271.

[2] Garey, M.R. and D.S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman & Company, New York.

[3] Grotschel, M. and M.W. Padberg. 1985. Polyhedral Theory. In E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys (eds.), *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, pp. 251-306. John Wiley & Sons, Chichester.

[4] Karp, R.M. 1979. Reducibility Among Combinatorial Problems. In R.E. Miller and J.W. Thatcher (eds.), *Complexity of Computer Computations*, pp. 85-103. Plenum Press, New York.

[5] Kruskal, J.B.,Jr. 1956. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proceedings of American Mathematical Society*, **7**, pp. 48-50.

[6] Kuhn, H.W. 1955. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, **2**, pp. 83-97.