TSP Heuristics: Domination Analysis and Complexity

Abraham Punnen [a,1], Francois Margot [b], Santosh Kabadi [c,2]

[a] *Department of Mathematical Sciences, University of New Brunswick, Saint John, New Brunswick, Canada*

[b] *Department of Mathematics, University of Kentucky, Lexington, Kentucky, USA*

[c] *Faculty of Administration, University of New Brunswick, Fredericton, New Brunswick, Canada*

Abstract

We show that the $2-Opt$ and $3-Opt$ heuristics for the traveling salesman problem (TSP) on the complete graph K_n produce a solution no worse than the average cost of a tour in K_n in a polynomial number of iterations. As a consequence, we get that the domination numbers of the $2-Opt$, $3-Opt$, Carlier-Villon, Shortest Path Ejection Chain, and Lin-Kernighan heuristics are all at least $\frac{(n-2)!}{2}$. The domination number of the Christofides heuristic is shown to be no more than $\lceil \frac{n}{2} \rceil!$, and for the Double Tree heuristic and a variation of the Christofides heuristic the domination numbers are shown to be one (even if the edge costs satisfy the triangle inequality). Further, unless P=NP, no polynomial time approximation algorithm exists for the TSP on the complete digraph $\vec{K}_n$ with domination number at least $(n-1)!-k$ for any constant k or with domination number at least $(n-1)!-(\frac{k}{k+1}(n+r))!-1$ for any non-negative constants r and k such that $(n+r) \equiv 0 \ mod\ (k+1)$. The complexities of finding the median value of costs of all the tours in $\vec{K}_n$ and of similar problems are also studied.

Key words: Domination Analysis, Approximation Algorithms, Traveling Salesman Problem, Computational Complexity

---

Email addresses: `punnen@unbsj.ca` (Abraham Punnen), `fmargot@ms.uky.edu` (Francois Margot), `kabadi@unb.ca` (Santosh Kabadi).

[1] Work completed while visiting University of Kentucky. Work supported by the NSERC grant OPG 0170381

[2] Work supported by the NSERC grant OPG 0008085

April, 2002

1 Introduction

Let $K_n = (V, E)$ be the complete graph on vertex set $\{1, 2, \ldots, n\}$ and let $c(e)$ be a cost associated with $e \in E$. If G is a subgraph of K_n, we denote by $c(G)$ the sum of the costs of the edges in G. Let $\mathbb{F}$ be the family of all Hamiltonian cycles (tours) in K_n. The *symmetric traveling salesman problem* (STSP) is to find an $H_c^* \in \mathbb{F}$ such that $c(H_c^*) \leq c(H)$ for all $H \in \mathbb{F}$. Replacing K_n by the complete digraph $\vec{K}_n$ and defining $\mathbb{F}$ as the family of all the directed tours in $\vec{K}_n$ yields the *asymmetric traveling salesman* problem (ATSP). When a statement applies to both ATSP and STSP, we simply use the term traveling salesman problem (TSP). For a state of the discussion on TSP, we refer to the book [8]. Let $E = \{e_1, e_2, \ldots, e_m\}$. The vector $c = (c(e_1), c(e_2), \ldots, c(e_m))$ of edge costs is called the *cost vector*.

Let α be a heuristic algorithm for the TSP with edge cost vector c, let H_c^α be the solution produced by α and let H_c^* be an optimal solution. Assume that c is restricted to some domain $\mathbb{D} \subseteq \mathbb{R}^m$. One measure of the worst-case performance of α on these instances is the *performance ratio*

$$P_\alpha(\mathbb{D}) = \sup_{c \in \mathbb{D}} \{ \frac{c(H_c^\alpha)}{c(H_c^*)} : c(H_c^*) > 0 \} \ .$$

(We assume that there exists $c \in \mathbb{D}$ such that $c(H_c^*) > 0$. If $c(H_c^*) \leq 0$ for all $c \in \mathbb{D}$ then it makes more sense to consider such problems as maximization problems.)

Clearly $P_\alpha(\mathbb{D}) \geq 1$ and the closer the performance ratio is to one, the better the worst case performance of the algorithm α is. Identifying the exact value of $P_\alpha(\mathbb{D})$ is usually difficult and hence upper bounds on the performance ratio are used instead. A heuristic α is a δ-approximation algorithm with respect to a domain $\mathbb{D}$ if $P_\alpha(\mathbb{D}) \leq \delta$. Unless P=NP, no polynomial time δ-approximation algorithm exists for the TSP with $\mathbb{D} = \mathbb{R}^m$ for any constant $\delta \geq 1$ [26]. If $\mathbb{R}^m_\Delta = \{c \in \mathbb{R}^m :$ the edge-costs $c(e)$ satisfy the triangle inequality$\}$, the Christofides heuristic is a 3/2-approximation algorithm with respect to the domain $\mathbb{R}^m_\Delta$ [2]. The existence of an algorithm with a better performance ratio is an important open question.

Recently Glover and Punnen [5] proposed domination analysis as another approach to measure the quality of a heuristic algorithm α: For a given edge cost vector c and tours $H_1, H_2 \in \mathbb{F}$, we say that H_1 *dominates* H_2 if and only if $c(H_2) \geq c(H_1)$. Let $F_\alpha(c) = \{H \in \mathbb{F} : c(H) \geq c(H_c^\alpha)\}$. Define the *domination number* of α with respect to a domain $\mathbb{D} \subseteq \mathbb{R}^m$ by

$$\mathrm{dom}(\alpha, \mathbb{D}) = \inf_{c \in \mathbb{D}} |F_\alpha(c)| \ .$$

The definition extends directly to a heuristic for any minimization combinatorial problem with a feasible set $\mathbb{F}$. For the TSP on n nodes, when $\mathbb{D} = \mathbb{R}^m$, $\mathrm{dom}(\alpha, \mathbb{D})$ is denoted by $\mathrm{dom}(\alpha, n)$. For any heuristic α for the TSP, the domination number exists and it is at least one, as the problem is always feasible. If $\mathrm{dom}(\alpha, n) = |\mathbb{F}|$, then α is an exact algorithm producing an optimal solution. Thus the goal is to develop heuristic algorithms with domination number close to $|\mathbb{F}|$.

Let $\phi : \mathbb{R} \longrightarrow \mathbb{R}$ be a real valued function. Let $\phi(c)$ be the vector obtained by applying ϕ to each entry in c. The function ϕ is an *order preserving transformation* for the TSP if and only if, for all cost vectors c, the ranking of the solutions in $\mathbb{F}$ is the same with respect to c and $d = \phi(c)$; i.e. $d(H_i) \leq d(H_j)$ if and only if $c(H_i) \leq c(H_j)$. For example $d(i,j) = \mu c(i,j) + a_i + b_j$, where $\mu > 0$ and $a_i, b_i \in \mathbb{R}$, is an order preserving transformation for the TSP. Any solution improvement heuristic (local search [17]), in which in every iteration the search neighborhood and tie breaking rule are independent of the actual values of the edge costs, produces the same solution when applied to instances with edge costs c and d as defined above. This gives us the following theorem.

Theorem 1 *Let α be a heuristic algorithm for the TSP that is invariant under an order preserving transformation. Then* $\mathrm{dom}(\alpha, \mathbb{R}^m) = \mathrm{dom}(\alpha, \mathbb{R}^m_\Delta)$.

Proof. Since $\mathbb{R}^m_\Delta \subset \mathbb{R}^m$ we have

$$\mathrm{dom}(\alpha, \mathbb{R}^m) \leq \mathrm{dom}(\alpha, \mathbb{R}^m_\Delta) \ . \tag{1}$$

Let $c \in \mathbb{R}^m$ and consider the order-preserving transformation $d(i,j) = c(i,j) + M$ where $M \geq 3\max\{|c(i,j)| : (i,j) \in E\}$. We have $d(i,j) + d(j,k) - d(i,k) = c(i,j) + c(j,k) - c(i,k) + M \geq 0$, i.e. $d \in \mathbb{R}^m_\Delta$. It follows that

$$\mathrm{dom}(\alpha, \mathbb{R}^m) \geq \mathrm{dom}(\alpha, \mathbb{R}^m_\Delta) \ . \tag{2}$$

The result follows from (1) and (2). ■

It may be noted that local search algorithms such as $2 - Opt$, $3 - Opt$, etc. are invariant under order preserving transformations. However, stability of the domination number under order preserving transformations does not hold for most construction heuristics, including the well-known Christofides heuristic. An exception to this is the Patching heuristic [15].

Domination analysis also indicates what percentage of the feasible region is 'covered' by the solution produced by an algorithm. This information together with other indicators may be useful in diversifying search paths in local search based metaheuristics [20].

Let us now discuss some notations and basic results used in our analysis. The average cost of all tours in $\vec{K}_n$ (resp. in K_n) with respect to cost vector c, denoted by $A(\vec{K}_n, c)$ (resp. $A(K_n, c)$), is given by [11,23,24]

$$A(\vec{K}_n, c) = \frac{1}{n-1}\, c(\vec{K}_n) \qquad \text{and} \qquad A(K_n, c) = \frac{2}{n-1}\, c(K_n)\ . \tag{3}$$

Sarvanov [24] showed that, for the ATSP with n odd, there are at least $(n-2)!$ tours in $\vec{K}_n$ having objective function value greater than or equal to $A(\vec{K}_n, c)$. He also suggested a computational scheme based on Hamiltonian decomposition of $\vec{K}_n$ to find a solution with objective function value no worse than $A(\vec{K}_n, c)$. When n is even, he showed that there are at least $\frac{(n-2)!}{2}$ tours in $\vec{K}_n$ with objective function value greater than or equal to $A(\vec{K}_n, c)$. His proof and result for the case of odd n extend directly to the case of even n, using a Hamiltonian decomposition of $\vec{K}_n$ for even n. The existence of such a decomposition for $n \geq 8$ was proved by Tillson [27]. Recently, Gutin and Yeo [11] independently showed that for any $n \neq 6$ there are at least $(n-2)!$ tours in $\vec{K}_n$ with objective function value at least $A(\vec{K}_n, c)$. As a direct consequence, it can be seen that there are at least $\frac{(n-2)!}{2}$ tours in K_n having objective function value no less than $A(K_n, c)$. This was proved independently by Rublineckii [23]. In fact, Rublineckii [23] proved a stronger result that there are at least $(n-2)!$ tours in K_n with objective function value no less than $A(K_n, c)$ when n is odd. These results can be summarized as follows:

Theorem 2 *For a heuristic algorithm α for the ATSP and $n \neq 6$, if $c(H_c^\alpha) \leq A(\vec{K}_n, c)$ for every cost vector c, then* $\mathrm{dom}(\alpha, n) \geq (n-2)!$. *In the case of STSP, if $c(H_c^\alpha) \leq A(K_n, c)$, then* $\mathrm{dom}(\alpha, n) \geq \frac{(n-2)!}{2}$ *for even n and* $\mathrm{dom}(\alpha, n) \geq (n-2)!$ *for odd n.*

There are several polynomial time heuristic algorithms available for the TSP that produce solutions with objective function value at least as good as $A(\vec{K}_n, c)$ (or $A(K_n, c)$) [7,11,21,23–25,28] and hence the domination number of each of these heuristics is at least $(n-2)!$ for the ATSP and $(n-2)!/2$ (or $(n-2)!$) for the STSP (depending on n is even or odd), whichever is applicable.

In this paper we study the domination number of the Christofides, Double Tree, $2-Opt$, $3-Opt$ and Node-Shifting heuristics and give some complexity results related to domination analysis. The paper is organized as follows. In section 2 we show that the $2-Opt$ heuristic for STSP is guaranteed to produce a solution with value no more than $A(K_n, c)$ in a polynomial number of iterations and hence the domination number of $2-Opt$ is at least $\frac{(n-2)!}{2}$. In Section 3 we consider domination analysis of Carlier and Villon algorithm [1]. We also observe that the domination numbers of the Shortest Path Generation algorithm of Glover [5] and its variations [20], and the Lin-Kernighan algorithm [16] are at least $(n-2)!/2$. Section 4 shows that the Node-Shifting heuristic may produce a solution worse than $A(K_n, c)$ and a data dependent bound on the objective function value of this solution is given. Further, we show that a partial $3-Opt$ heuristic (and hence the $3-Opt$ heuristic) produces a solution with objective function value no worse than the average cost of all tours. In Section 5 we consider domination analysis of the Christofides algorithm [2] and the Double Tree algorithm [15]. It is shown that the domination number of the Christofides algorithm is at most $(\frac{n}{2})!$ when n is even and at most $(\frac{n+1}{2})!$ when n is odd. Further, we show that the Double Tree algorithm as well as two variations of the Christofides algorithm have domination number one. In section 6 we show that, unless P = NP, no polynomial time algorithm for the ATSP exists with domination number at least $(n-1)!-k$ for any constant k or with domination number at least $(n-1)!-(\frac{k}{k+1}(n+r))!-1$ for any non-negative constants r and k such that $n+r$ is divisible by $k+1$. We also show that, unless P=NP, there is no polynomial time algorithm to compute the value of a tour in $\vec{K}_n$ that dominates exactly $(n-1)!p/k$ tours where $p \in \{1, \ldots, k-1\}$ for any integer constant $k \geq 2$.

Gutin and Yeo [10] considered the possibility of polynomial algorithms with domination number $(n-1)!k$, where k is a constant. In particular, they showed that if there exists a constant $r > 1$ such that for sufficiently large k, every k-regular digraph with number of nodes $n < rk$ admits a Hamiltonian decomposition and such a decomposition can be obtained in polynomial time, then a tour which dominates at least $(n-2)!(n-k)$ tours in $\vec{K}_n$ can be identified in polynomial time. It is mentioned in [10] that Häggkvist has announced (but not yet published) a proof that $r = 2$ works.

Without loss of generality we assume that the nodes of the complete graph are numbered $\{1, \ldots, n\}$ and that all node-labels and subscripts are taken *modulo n*.

2 Domination Analysis of 2-Opt

The $2-Opt$ heuristic is a simple and well-known local search algorithm for the STSP [15]. Consider a Hamiltonian cycle H of K_n. Without loss of generality, we assume that $H = (1, 2, \ldots, n, 1)$. A *2-exchange operation* replaces two non-adjacent edges $(i, i+1)$ and $(j, j+1)$ from H by the edges (i, j) and $(i+1, j+1)$ to get a new tour H_{ij}. Let $\Delta_{ij} = c(H_{ij}) - c(H)$. Then

$$\Delta_{ij} = c(i, j) + c(i+1, j+1) - c(i, i+1) - c(j, j+1) \ .$$

For any i in $\{1, \ldots, n\}$, let $N_i = \{1, \ldots, n\} - \{i-1, i, i+1\}$. If $\Delta_{ij} \geq 0$ for all $i = 1, 2, \ldots, n$ and all j in N_i, then H is said to be *locally optimal* for $2-Opt$. For any node r of K_n, let $\delta_r = \sum_{j=1, r\neq j}^{n} c(r, j)$. The following theorem was proved by Rublinekii [23].

Theorem 3 *If H is locally optimal for $2-Opt$ then $c(H) \leq A(K_n, c)$.*

Proof. Let $\Delta_i = \sum_{j \in N_i} \Delta_{ij}$. Then

$$\Delta_i = -(n-2)c(i, i+1) - c(H) + \delta_i + \delta_{i+1}, \text{ for } i = 1, 2, \ldots, n.$$

Adding these n equations together, we get

$$\Delta = \sum_{i=1}^{n} \Delta_i = -(2n-2)c(H) + 4c(K_n) \ . \tag{4}$$

If H is locally optimal for $2-Opt$ then $\Delta_i \geq 0$ for all i and hence $\Delta \geq 0$. Thus (4) yields $c(H) \leq \frac{2c(K_n)}{n-1} = A(K_n, c)$. ■

Corollary 4 *The domination number of $2-Opt$ is at least $\frac{(n-2)!}{2}$ when n is even and $(n-2)!$ when n is odd.*

This follows from theorems 2 and 3.

To reach a tour which is locally optimal with respect to $2-Opt$ neighborhood, one may need an exponential number of iterations [3]. Hence it is natural to ask whether a large domination number can be achieved after a polynomial number of $2-Opt$ iterations. In $2-Opt$, the average cost of a 2-exchange operation from tour H, denoted by $\overline{\Delta}(H)$, is given by $\frac{\Delta}{n(n-3)}$. Hence, using (3) and (4),

$$\overline{\Delta}(H) = \frac{2(n-1)}{n(n-3)} \left[A(K_n, c) - c(H)\right] \ . \tag{5}$$

Suppose we begin the $2-Opt$ heuristic with a tour H such that $c(H) > A(K_n, c)$. Then $\overline{\Delta}(H) < 0$ and, for the optimal choice of non-adjacent edges $(i, i+1)$ and $(j, j+1)$ for the 2-exchange operation, $\Delta_{ij} \leq \overline{\Delta}(H) < 0$. It follows from equation (5) that this 2-exchange operation reduces the value of $c(H) - A(K_n, c)$ to $c(H_{ij}) - A(K_n, c)$, i.e. by a factor of at least $\frac{2(n-1)}{n(n-3)}$. To count the number of iterations needed to reach a solution with value at most $A(K_n, c)$, consider the order preserving transformation $\phi(x) = x - \frac{A(K_n,c)}{n}$ and let $d = \phi(c)$. The sequence of tours $H_1, H_2, \ldots, H_q$, where $c(H_q) \leq A(K_n, c)$ and $c(H_i) > A(K_n, c)$ for $i < q$, produced by the $2-Opt$ heuristic for the cost vectors c and d are the same provided that they start from the same initial tour H_0. For cost vector d, the average cost of all possible tours is 0 and therefore, for any starting tour H_0 with $c(H_0) > A(K_n, c)$ (implying $d(H_0) > 0$), we have

$$d(H_i) \leq \left(1 - \frac{2(n-1)}{n(n-3)}\right) \ d(H_{i-1}) \ \text{ for } i = 1, 2, \ldots, q \ . \tag{6}$$

¿From equation (6), using a result of Grover [7] we get $q = O(n \log(d(H_0)))$, i.e. $q = O(n \log(c(H_0) - A(K_n, c)))$. We shall now get a strongly polynomial bound for q using the following result of Goemans, cited in [22].

Lemma 5 *Let $c = (c_1, c_2, \ldots, c_p)$ be a real vector and let $y_1, y_2, \ldots, y_q$ be vectors in $\{-1, 0, 1\}^p$. If, for all $i = 1, 2, \ldots, (q-1)$, $0 \leq y_{i+1}c^T \leq \frac{1}{2}y_i c^T$, then $q = O(p \log p)$.*

Since

$$\left(1 - \frac{2(n-1)}{n(n-3)}\right)^n \leq \left(1 - \frac{2}{n}\right)^n \leq e^{-2} ,$$

from equation (6), we see that, after $O(n)$ 2-exchanges, we get a solution with value at most $d(H_0)/2$. Thus, using Lemma 5 with $p = n(n-1)/2$, we get $q = O(n^3 \log n)$. The foregoing discussion can be summarized as

Theorem 6 *For the STSP, the $2 - Opt$ algorithm produces a solution with value at most $A(K_n, c)$ in $O(min\{n^3 \log n, n \log(c(H_0) - A(K_n, c))\})$ iterations, where H_0 is the starting solution.*

Note that the solution indicated in Theorem 6 need not be locally optimal with respect to the $2 - Opt$ neighborhood. It may also be noted that it is not necessary to make optimal choice of edges to be exchanged in each iteration to achieve this bound. Any pair $(i, i+1), (j, j+1)$ such that $\Delta_{ij} \leq \overline{\Delta}(H)$ is good enough. One could even permit a constant number of first improvement $2 - Opt$ moves between two such iterations and still get the same complexity bound.

3 Carlier-Villon Algorithm

Let the nodes of K_n be labelled as $1, 2, \ldots, n$. A tour $(\pi(1), \pi(2), \ldots, \pi(n), \pi(1))$ of K_n is *pyramidal with respect to this node ordering* if there exists an index $1 \leq k \leq n$ such that $\pi(k) = n$ and

$$\pi(1) < \pi(2) < \ldots < \pi(k) > \pi(k+1) > \ldots > \pi(n) .$$

There are 2^{n-3} pyramidal tours in K_n with respect to a given node labelling and the best pyramidal tour can be obtained in $O(n^2)$ operations [14].

Let $H = (u_1, u_2, \ldots, u_n, u_1)$ be an arbitrary tour of K_n. The *Carlier-Villon neighborhood* of H, CV(H), is defined as follows [1]: Choose any node u_i of H and relabel the nodes $u_i, u_{i+1}, \ldots, u_n, u_1, u_2, \ldots u_{i-1}$ respectively as $1, 2, \ldots n$. Let F_i be the class of all tours of K_n that are pyramidal with respect to this new labelling of the nodes. Then $CV(H) = \cup_{i=1}^{n} F_i$. The best member in $CV(H)$ can be identified in $O(n^3)$ operations by repeated application of the algorithm for computing the best pyramidal tour. A local search algorithm selecting the best solution in $CV(H)$ is called the CV-algorithm [1]. Let $2-Opt(H)$ denote the $2-Opt$ neighborhood of H.

A tour that is locally optimal with respect to the CV neighborhood is also locally optimal with respect to the $2-Opt$ neighborhood, as mentioned in [12]. The following lemma yields this result.

Lemma 7 $2-Opt(H) \subseteq CV(H)$.

Proof. Let $H = (u_1, u_2, \ldots, u_n, u_1)$. We show that $\hat{H} \in 2-Opt(H)$ implies $\hat{H} \in CV(H)$. Choose two arbitrary non-adjacent edges (u_r, u_{r+1}) and (u_s, u_{s+1}) of H. Without loss of generality, let us assume that $r < s$. Let $\hat{H}$ be the tour obtained by a 2-exchange operation involving these two edges. Thus

$$\hat{H} = (u_1, u_2, \ldots, u_r, u_s, u_{s-1}, \ldots, u_{r+1}, u_{s+1}, u_{s+2}, u_n, u_1) \ .$$

It can be verified that $\hat{H} \in F_{s+1}$ and hence $\hat{H} \in CV(H)$. ■

Theorem 8 *The CV-algorithm produces a solution to the STSP with value no more than $A(K_n, c)$ in $O(min\{n^3 \log n, n \log(c(H_0) - A(K_n, c))\})$ iterations and the domination number of this algorithm is at least $\frac{(n-2)!}{2}$ for even n and $(n-2)!$ for odd n.*

The proof of this theorem follows from Lemma 7 and theorems 3 and 6. It may be noted that the previously best known domination number of the CV-algorithm for STSP was $n2^{n-3}$ [1]. As in the case of Theorem 6, the solution indicated in Theorem 8 need not be locally optimal with respect to the CV(H) neighborhood. The worst case complexity of this algorithm is not known to be polynomial.

Note that the CV-algorithm may be used on directed graphs too. However, the argument used in proving Theorem 8 is not valid for directed graphs.

Gutin and Yeo [9] showed that any polynomial time heuristic α for the STSP can be modified to get a polynomial time heuristic α^* for the ATSP such that $\text{dom}(\alpha^*, n) \geq \text{dom}(\alpha, n)$: Given an instance of ATSP with input graph $\vec{K}_n$ and arc cost vector c construct an instance of STSP on K_n with edge costs $d(i,j) = (c(i,j) + c(j,i))/2$. Let H_d^α be the solution produced by α for this instance. Of the two tours in $\vec{K}_n$ corresponding to H_d^α, one in forward and the other in backward direction, choose the one with lesser cost. Choosing α as the CV-algorithm for STSP, we get an algorithm for ATSP with domination number at least $\frac{(n-2)!}{2}$.

Ejection chain algorithms is another class of important powerful heuristics for the TSP. Two of the most important members of this class of heuristics are the Lin-Kernighan algorithm [16] and the shortest path ejection chain algorithms [6,20]. Several variations of these algorithms are known. In many variations of these algorithms a local optimum solution is also a local optimum with respect to the $2-Opt$ neighborhood. For all such variations of the Lin-Kernighan algorithm and the Shortest Path Ejection Chain algorithm it can be shown that the domination numbers are at least $\frac{(n-2)!}{2}$.

4 Node-shifting and 3-Opt heuristics

In this section we investigate the values of local optima with respect to neighborhoods that are subsets of the well known $3-Opt$ neighborhood [15]. Let us first consider the node-shifting neighborhood. Let H be a tour in $\vec{K}_n$. Without loss of generality we assume that $H = (1, 2, \ldots, n, 1)$. Eject a node i and an edge $(j, j+1), j \neq i, i-1$, from H and introduce the edges $(i-1, i+1)$, (j, i) and $(i, j+1)$ to form the tour H_{ij}. We call this operation *shifting node i between nodes j and $j+1$* and we say that H_{ij} is obtained from H by *node-shifting*. The collection of all tours that can be obtained from H by node-shifting is called the *node-shifting neighborhood* of H. The node-shifting neighborhood is defined for both STSP and ATSP. Unlike the $2-Opt$ neighborhood, the objective function value of a locally optimal solution corresponding to the node-shifting neighborhood could be worse than $A(\vec{K}_n, c)$ (or $A(K_n, c)$) as illustrated by the following example.

Consider a complete graph on 10 nodes. (See Figure 1. The missing edges have cost zero.) All edges of the tour $H = (1, 2, 3, \ldots, 10, 1)$ have cost 1, all edges of

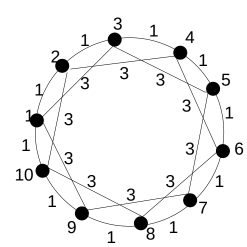

Fig. 1.

the subtours $(1, 3, 5, 7, 9, 1)$ and $(2, 4, 6, 8, 10, 2)$ have cost 3 and all other edges have cost zero. The average cost of all tours is $80/9$ and the cost of H is 10. It can be verified that H is locally optimal with respect to the node-shifting neighborhood.

The local search algorithm using the node-shifting neighborhood is called the *Node-Shifting algorithm*. Let us now give a data dependent bound on the objective function value of the solution produced by the Node-Shifting algorithm. We define the *mate* of the tour H, denoted by $\hat{H}$, as follows. If n

is odd then $\hat{H}$ is the tour $(1, 3, \ldots n-2, n, 2, \ldots, n-1, 1)$ and if n is even then $\hat{H}$ is the collection of two subtours $(1, 3, \ldots, n-1, 1)$ and $(2, 4, \ldots, n, 2)$. Let $\Lambda_{ij} = c(H_{ij}) - c(H)$. The tour H is said to be locally optimal with respect to the node-shifting neighborhood if $\Lambda_{ij} \geq 0$ for all $i = 1, 2 \ldots, n$ and $j = 1, 2, \ldots, n,\ j \neq i, i-1$. For each node r of $\vec{K}_n$ let $\vec{\delta}_r = \sum_{j \neq r} c(r, j)$ and $\tilde{\delta}_r = \sum_{j \neq r} c(j, r)$.

Theorem 9 *Let H be a locally optimal solution with respect to the node-shifting neighborhood. Then for the ATSP, $c(H) \leq \frac{(n-2)}{3n-4} c(\hat{H}) + \frac{2(n-1)}{3n-4} A(\vec{K}_n, c)$ and for the STSP, $c(H) \leq \frac{(n-2)}{3n-4} c(\hat{H}) + \frac{2(n-1)}{3n-4} A(K_n, c)$.*

Proof. It can be verified that $\Lambda_{ij} = -c(i-1, i) - c(i, i+1) + c(i-1, i+1) + c(j, i) + c(i, j+1) - c(j, j+1)$. Adding all these Λ_{ij} values for $j = 1, 2, \ldots, n$, $j \neq i-1, i$ and denoting the sum as Λ_i we get

$$\Lambda_i = (n-2)(-c(i-1, i) - c(i, i+1) + c(i-1, i+1)) + \vec{\delta}_i + \tilde{\delta}_i - c(H)\ .$$

Thus

$$\Lambda = \sum_{i=1}^{n} \Lambda_i = -(3n-4)c(H) + (n-2)c(\hat{H}) + 2c(\vec{K}_n)\ .$$

If H is locally optimal, with respect to the node-shifting neighborhood, then $\Lambda \geq 0$. Thus we have

$$c(H) \leq \frac{(n-2)c(\hat{H}) + 2c(\vec{K}_n)}{3n-4}\ . \tag{7}$$

The result now follows from the formula for $A(\vec{K}_n, c)$ indicated earlier. The case of the STSP can be proved in a similar way. ■

Corollary 10 *If $c(\hat{H}) \leq c(H)$ then, for the STSP, $c(H) \leq A(K_n, c)$ and for the ATSP, $c(H) \leq A(\vec{K}_n, c)$.*

Corollary 11 *If the edge costs satisfy the triangle inequality, then $c(H) \leq \frac{2(n-1)}{n} A(\vec{K}_n, c)$ for the ATSP and $c(H) \leq \frac{2(n-1)}{n} A(K_n, c)$ for the STSP.*

Proof. If the edge costs satisfy the triangle inequality, $c(\hat{H}) \leq 2c(H)$ and the result follows from (7). ■

We shall now show that, by considering additional $3-Opt$ exchanges, a solution can be obtained with value at most $A(K_n, c)$. Let $(i, i+1), (j, j+1), (u, u+1)$ be a triplet of distinct edges in $H = (1, 2, \ldots, n, 1)$ that do not form a path of length 3. Consider the following types of $3-opt$ exchanges:

Type 1: For $j = i+1$, let the new tour obtained by this exchange be the tour H_{iu} obtained by shifting node i between u and $u+1$.

Type 2: Suppose that edges $(i,i+1),(j,j+1),(u,u+1)$ are pairwise non-adjacent and that $i<j<k$. Replace the edges $(i,i+1),(j,j+1),(u,u+1)$ by the edges

Type 2(a): $(i,j+1),(j,u),(i+1,u+1)$ to get a new tour H^1_{iju}.

Type 2(b): $(i,u),(i+1,j+1),(j,u+1)$ to get a new tour H^2_{iju}.

Type 2(c): $(i,j),(i+1,u),(j+1,u+1)$ to get a new tour H^3_{iju}.

Now we consider the *partial 3-Opt neighborhood* corresponding to type 1, type 2(a), 2(b), and 2(c) exchanges. Recall that, for the tour H_{ij} obtained using node-shifting operations, we have defined

$$\begin{aligned}\Lambda_{iu} = c(H_{iu}) - c(H) = &-c(i-1,i) - c(i,i+1) + c(i-1,i+1) \\ &+c(u,i) + c(i,u+1) - c(u,u+1)\ .\end{aligned} \tag{8}$$

Define

$$\begin{aligned}\Lambda^1_{iju} = c(H^1_{iju}) - c(H) = &\ c(i,j+1) + c(u,j) + c(i+1,u+1) \\ &-c(i,i+1) - c(j,j+1) - c(u,u+1)\end{aligned} \tag{9}$$

$$\begin{aligned}\Lambda^2_{iju} = c(H^2_{iju}) - c(H) = &\ c(i,u) + c(j+1,i+1) + c(j,u+1) \\ &-c(i,i+1) - c(j,j+1) - c(u,u+1)\end{aligned} \tag{10}$$

$$\begin{aligned}\Lambda^3_{iju} = c(H^3_{iju}) - c(H) = &\ c(i,j) + c(i+1,u) + c(j+1,u+1) \\ &-c(i,i+1) - c(j,j+1) - c(u,u+1)\ .\end{aligned} \tag{11}$$

Theorem 12 *If H is locally optimal with respect to the partial $3-$Opt neighborhood, then $c(H) \le A(K_n,c)$.*

Proof. Consider a Hamiltonian cycle H in K_n. Without loss of generality, we assume that $H=(1,2,\ldots,n,1)$. Adding all the inequalities (8) (over all non-adjacent i and u), and (9), (10), (11) (over all $1 \le i < j < u \le n$), and using symmetry, we get,

$$\Delta = \sum \Lambda_{iu} + \sum \Lambda^k_{iju} \tag{12}$$

$$= -r_1 c(H) + r_2 c(S_1) + \sum_{e \in S_2} r(e)c(e), \tag{13}$$

for some r_1, r_2 and $\{r(e) : e \in S_2\}$ where, $S_1 = \{(i,i+2) : i \in \{1,2,\ldots,n\}\}$ and $S_2 = E - (S_1 \cup H)$. An edge e in H occurs in $3(n-4)$ exchanges of type 1 and $\frac{(n-4)(n-5)}{2}$ exchanges of each of the types 2(a), 2(b) and 2(c). Hence, $r_1 = \frac{3(n-3)(n-4)}{2}$.

Let $e \in S_1$ with $e = (a, b)$ and assume w.l.o.g. that $b \equiv a + 2 \ (mod\ n)$. Edge e is involved in $(n-2)$ exchanges of type 1: $(n-4)$ of them with $i = a+1$, one with $i = a, u = b$ and one with $i = b, u = a-1$. It is also involved in $2(n-5)$ exchanges of type 2: e can possibly be only edge (j, u) or $(i+1, u+1)$ for 2(a) exchanges, edge (i, u) or $(i+1, j+1)$ for 2(b) exchanges, and edge (i, j) or $(j+1, u+1)$ for 2(c) exchanges. If $1 \leq a < b \leq n$, observe that the cases where (a, b) is (i, j) or (j, u) always account for $(n-5)$ choices of i, j, u, as well as the cases $(i+1, j+1)$ or $(j+1, u+1)$. Since the remaining two cases are excluded, e is indeed in $2(n-5)$ exchanges as claimed. If $1 \leq b < a \leq n$, then (a, b) is either (u, i) or $(u+1, i+1)$, accounting for $2(n-5)$ exchanges as claimed. Thus $r_2 = 2(n-5) + (n-2) = 3(n-4)$.

Now consider any edge $e = (a, a+k)$ in S_2 for some $a \in V$ and $2 < k < n-2$. Let us count the number of type 2 exchanges in which edge e is involved. Since the edges $\{(i, i+1), (j, j+1), (u, u+1)\}$ are pairwise non-adjacent in H, the set $\{i, j, u\}$ should contain precisely one of $a-1$ and a and it should contain precisely one of $a+k-1$ and $a+k$. Then e will be involved in precisely one of the corresponding type 2(a), 2(b), 2(c) exchanges. Now consider the following four cases. Case 1: $\{a-1, a+k-1\} \subset \{i, j, u\}$, case 2: $\{a, a+k\} \subset \{i, j, u\}$, case 3: $\{a, a+k-1\} \subset \{i, j, u\}$ and case 4: $\{a-1, a+k\} \subset \{i, j, u\}$. In each of cases 1 and 2, we have $n-6$ choices for the third edge. In case 3, we have $n-k-2$ choices for the third edge, while in case 4, we have $k-2$ choices for the third edge. Thus the total number of type 2 exchanges in which e is involved is $3n-16$. Also edge e is involved in four type 1 exchanges, (i) $i = a, u = a+k-1$, (ii) $i = a, u = a+k$, (iii) $i = a+k, u = a-1$, and (iv) $i = a+k, u = a$. Thus $r(e) = 3n - 16 + 4 = 3(n-4)$. Since edge e was chosen arbitrarily in S_2, $r(e) = 3(n-4)$ for all e in S_2. By substituting the values of r_1, r_2 and $\{r(e) : e \in S_2\}$ in equation (13) we get,

$$\Delta = 3(n-4)c(K_n) - \frac{3(n-4)(n-1)}{2}c(H) \tag{14}$$

$$= \frac{3}{2}(n-4)(n-1)\left[A(K_n, c) - c(H)\right] \tag{15}$$

Now, if H is locally optimal with respect to the partial $3-Opt$ neighborhood, then $\Lambda_{iu} \geq 0$ for all non-adjacent i and u, and $\Lambda_{iju}^k \geq 0$ for all $1 \leq i < j < u \leq n$ and $k = 1, 2, 3$. Hence,

$$\Delta = \frac{3}{2}(n-4)(n-1)\left[A(K_n, c) - c(H)\right] \geq 0.$$

This implies that $c(H) \leq A(K_n, c)$. ■

Corollary 13 *If H is locally optimal for $3-Opt$, then $c(H) \leq A(K_n, c)$. Further the domination number of $3-Opt$ heuristic is at least $(n-2)!/2$.*

We now show that the partial $3-Opt$ heuristic produces tour with value no more than $A(K_n, c)$ in polynomial time.

Theorem 14 *The partial* $3-Opt$ *heuristic produces a tour with cost no more than* $A(K_n, c)$ *in* $O(\min\{n^3\log(n), n\log(c(H)-A(K_n, c))\})$ *iterations, where* H *is the starting solution.*

Proof. The total number of exchanges of types $1, 2(a), 2(b)$ and $2(c)$ are obviously

$$\frac{nr_1}{3} = \frac{n(n-3)(n-4)}{2}.$$

Hence the average cost of a partial $3-Opt$ heuristic is given by,

$$\overline{\Delta} = \frac{2\Delta}{n(n-3)(n-4)} \tag{16}$$

$$= \frac{3(n-1)}{n(n-3)}\left[A(K_n, c) - c(H)\right] \tag{17}$$

The proof now follows along the same lines as the proof of Theorem 6, by replacing equation (5) with equation (17). ■

As an immediate consequence of the above theorem, we have the following complexity result for the $3-Opt$ heuristic.

Corollary 15 *The* $3-Opt$ *heuristic produces a tour with cost no more than* $A(K_n, c)$ *in* $O(\min\{n^3\log(n), n\log(c(H)-A(K_n, c))\})$ *iterations, where* H *is the starting solution.*

5 Christofides and Double Tree Heuristics

The Christofides algorithm [2] is a well known $\frac{3}{2}$-approximation algorithm for the STSP when the edge costs satisfy the triangle inequality. The algorithm constructs a minimum cost spanning tree T of K_n and a minimum cost perfect matching M of the nodes of odd degree in T. The graph $B = T \cup M$ is thus Eulerian and connected. Select an ordering of the edges in B to produce an Eulerian tour of B. Then, starting at an arbitrary node of B, traverse the Eulerian tour, introducing shortcuts to skip already visited nodes, to obtain a Hamiltonian tour H. The quality of H depends on the selected perfect matching, on the starting node of the Eulerian tour and on the Eulerian tour itself. Among all the possible tours that can be generated in this way, identifying the best tour is an NP-hard problem [18].

Theorem 16 *The domination number of Christofides heuristic is at most $(\frac{n}{2})!$ for even n and at most $(\frac{n+1}{2})!$ for odd n, even if the edge costs satisfy the triangle inequality.*

Proof. To prove this theorem, we only need to construct an instance of the STSP for which Christofides algorithm produces a tour that dominates only the claimed number of tours. Consider the case of n even. Let $V = \{1, 2, \ldots, n\}$ be the node set of the complete graph K_n. Let the cost of each of the edges in $M^* = \{(3, 4), (5, 6), \ldots, (n-1, n)\}$ be two. All other edges of K_n have cost one. In this case, the star at node 1 is a minimum spanning tree T. Let $M = \{(2, 3), (4, 5), \ldots, (n-2, n-1), (n, 1)\}$. Observe that M is a minimum cost perfect matching of the nodes of odd degree in T. Consider the Eulerian tour

$$n, 1, 2, 3, 1, 4, 5, 1, \ldots, (n-1), 1, n$$

in $B = T \cup M$. The short-cutting phase of Christofides algorithm produces the tour

$$(n, 1, 2, 3, 4, \ldots, n-1, n)\ .$$

This is one of the worst tours in K_n and there are exactly $(\frac{n}{2})!$ tours in K_n having this cost. A similar example yields the result for n odd. ■

Let us consider a variation of the Christofides algorithm designed to improve its performance when applied to an instance where the edge costs do not satisfy the triangle inequality. We do not know who introduced this variation and we call it *modified Christofides algorithm.* The algorithm can be described as follows.

Step 1: Find a minimum cost spanning tree T in K_n.

Step 2: Find a minimum cost perfect matching M in the subgraph $G(T)$ of K_n induced by the odd degree vertices of T with the cost of edge (i, j) in $G(T)$ equal to the cost of the shortest ij-path in K_n.

Step 3: For each edge $(i, j) \in M$, let $P(i, j)$ be a shortest ij-path in K_n. Let $M^* = \cup_{(i,j)\in M} P(i, j)$.

Step 4: Consider the Eulerian graph $T \cup M^*$. Using shortcuts, as in the Christofides algorithm, produce a tour in K_n.

Theorem 17 *The domination number of the modified Christofides algorithm is one.*

Proof. We give an example where the modified Christofides algorithm produces the worst tour and this tour is the only one with that value. Consider the complete graph $K_n = (V, E)$ where $V = \{1, 2, \ldots, n\}$. Assume that n is even. Let the cost of each edge incident to node 1 be zero and the costs of the edges $\{(3, 4), (4, 5), \ldots, (n-2, n-1), (n-1, n), (n, 2)\}$ be 10. Assign a cost of one to the remaining edges of K_n. The minimum cost spanning tree T in this case will be the star at node 1. Note that every node has odd degree in T and that the shortest ij-path, for $i \neq j$, is a two edges path passing through node 1 with a cost of 0. Hence, $(1,2),(3,4),\ldots,(n-1,n)$ is a minimum cost perfect matching in $G(T)$ and the resulting connected, Eulerian graph (as obtained in Step 4 of the algorithm) is precisely the double tree obtained from the star at node 1. Consider the Eulerian traversal $(2,1,3,1,4,1,5,1,\ldots,n-1,1,n,1,2)$. The short-cutting phase of the modified Christofides algorithm produces the tour $(2,1,3,4,5,\ldots,n-1,n,2)$. It can be verified that it is the unique worst tour in K_n and hence the domination number the modified Christofides algorithm is one. The case where n is odd is similar. ■

In the modified Christofides algorithm discussed earlier, if Step 2 is replaced by

Step 2: Find a minimum cost perfect matching M in the subgraph $G(T)$ of K_n induced by the odd degree vertices of T

we get yet another version of the Christofides algorithm. Under triangle inequality, this version also guarantees a 3/2-approximate solution for the TSP. However it is possible to show that the domination number of this variation of Christofides algorithm is one even if the edge costs satisfy the triangle inequality.

Let us now discuss the domination number of the Double Tree algorithm [15]. This heuristic first computes a minimum spanning tree of K_n, duplicates its edges to form an Eulerian graph and then, as the Christofides algorithm, follows an Eulerian tour and uses shortcuts to skip already visited nodes to produce a tour in K_n. As in the case of Christofides algorithm, it is easy to show that finding the best double-tree tour (over all possible Eulerian tours) is NP-hard.

Theorem 18 *The domination number of the Double Tree algorithm is one even if the edge costs satisfy the triangle inequality.*

Proof. We shall construct an instance of the STSP where the Double Tree algorithm produces the worst tour and where this worst tour is unique. Consider the complete graph $K_n = (V,E)$ where $V = \{1,2,\ldots,n\}$. Let the cost of each edge incident to node 1 be one and the cost of the edges in $M = \{(3,4),(4,5),\ldots,(n-2,n-1),(n-1,n),(n,2)\}$ be two. All other edges of K_n are of cost one. In this case the star at node 1 is a minimum spanning tree. Consider the Eulerian tour $(2,1,3,1,4,1,5,1,6,1,7,\ldots,n-1,1,n,1,2)$ in the double tree obtained from T. The short-cutting phase of the Double Tree algorithm produces the tour $(2,1,3,4,5,\ldots,n-1,n,2)$. It can be verified that this is the unique worst tour in the graph and hence, the domination number of the Double Tree algorithm is one. ■

There are other heuristic algorithms for TSP known to have domination number equal to one. For details see Gutin, Yeo and Zverovich [13].

6 Complexity and Domination Analysis

An important open question in domination analysis is to identify the largest possible domination number for a polynomial time heuristic for the TSP. Although this question remains open, we prove some upper bounds under the assumption $P \neq NP$. In this section we consider primarily the ATSP and analogous results can be easily derived for the STSP.

Theorem 19 *Unless $P = NP$, no polynomial time algorithm for ATSP has domination number $(n-1)! - k$ for any constant k.*

Proof. Let α be a polynomial time algorithm for the ATSP having domination number $(n-1)! - k$ for some constant k. We show that α can be used to find a minimum cost Hamiltonian uv-path in a complete directed graph D for $u, v \in V(D)$ with arc cost vector d, a well-known NP-hard problem [4] that we refer to as MHP(u, v). Construct a corresponding instance of the ATSP with cost vector c as follows. Let $D^{'}$ be a complete digraph with $k^{'}$ vertices, such that $k'! > k$. Join each node of $D^{'}$ to each node of $D - \{u, v\}$ by two opposite arcs e, e' with cost $c_e = c_{e'} = M$, where $M > \sum_{e \in D} |d_e|$. For each node $x \in D^{'}$, let $c(u, x) = M$ and $c(x, v) = M$. Arcs joining any two nodes of D keep their original cost. All remaining arcs of the complete directed graph on the node set $V(D) \cup V(D')$ have cost zero. Let D^* be the resulting directed graph. Let $|V(D) \cup V(D')| = N$.

If P is a minimum cost Hamiltonian uv-path in D with respect to cost vector d, then there are at least $k^{'}!$ Hamiltonian cycles in D^* with cost equal to $d(P)$, obtained by extending P with the nodes in D' in all possible order. It can be verified that each of these $k^{'}!$ Hamiltonian cycles in D^* corresponds to an optimal solution to the ATSP on D^*. Furthermore, each optimal solution to the ATSP on D^* is of this form and from any one of these optimal solutions, an optimal Hamiltonian uv-path in D can be recovered. Since, $k^{'}! > k$, any tour in D^* dominating $(N-1)! - k$ tours must be one of these optimal tours in D^* and α serves as a polynomial time algorithm for Hamiltonian uv-path problem, which is impossible under the assumption $P \neq NP$. ■

Theorem 20 *Unless $P = NP$, there is no polynomial time approximation algorithm for the ATSP with domination number $(n-1)! - (\frac{k}{k+1}(n+r))! - 1$ for any non negative constants r and k with $(n+r) \equiv 0 \bmod (k+1)$.*

Proof. The proof of this theorem is similar to that of the previous theorem. The only difference is that we choose the number of nodes in D' to be $k(n+r)$. Note that there are $(k(n+r))!$ optimal tours in D^* and each one of them corresponds to an optimal solution to the Hamiltonian uv-path problem on D.

Let α be a polynomial time algorithm for the ATSP on $\vec{K}_n$ having domination number $(n-1)! - (\frac{k}{k+1}(n+r))! - 1$ for some nonnegative constants k and r satisfying the condition of the theorem. If α is applied on D^*, (note that D^* has $N = n + k(n+r)$ nodes), the resulting solution, say H will be no worse than $(N-1)! - (\frac{k}{k+1}(N+r))! - 1$ alternative tours. Now,

$$\begin{aligned}(N-1)! - (\frac{k}{k+1}(N+r))! &= (n+k(n+r)-1)! - (\frac{k}{k+1}(n+k(n+r)+r))! \\ &= (n+k(n+r)-1)! - (\frac{k}{k+1}(k+1)(n+r)))! \\ &= (n+k(n+r)-1)! - (k(n+r))!\end{aligned}$$

Thus H must be one of the $(k(n+r))!$ optimal hamiltonian cycles in D^*. From H an optimal Hamiltonian uv-path in D can be recovered. The result now follows. ■

By choosing appropriately the number of nodes in D' in the proofs of theorems 19 and 20, several related complexity results on domination analysis can be obtained. In particular, it can be shown that unless P=NP, there is no polynomial time heuristic algorithm for ATSP on $\zeta(n)$ nodes with domination number $(\zeta(n)-1)! - (\zeta(n)-n)! - 1$ for any integer valued polynomial function ζ, defined on positive integers, such that $\zeta(n) \geq n$.

We have seen that the average value of all tours in a graph can be obtained by evaluating a simple formula in $O(n^2)$ operations. One might wonder if this is also the case for the median cost of all the tours. We now show that unless $P = NP$ there is no polynomial time algorithm that computes the median of all the tour costs in $\vec{K}_n$. In fact we prove a more general result.

Theorem 21 *Unless $P = NP$, no polynomial time algorithm can compute the objective function value of a tour which dominates exactly $(n-1)!\ p/k$ tours where $p \in \{1, \ldots, k-1\}$ for any integer constant $k \geq 1$.*

Proof. Let α be a polynomial time algorithm to compute the objective function value of a tour which dominates exactly $\lfloor (n-1)!p/k \rfloor$ tours. We show that α can be used to solve MHP(u,v) in a complete directed graph $\vec{K}_n$, where n is divisible by k for a given integer k. We refer to this problem as *minimum (u,v)-Hamiltonian path problem with parameter k* (MHP(u,v,k) for short). This problem is NP-hard, as any instance of MHP(u,v) on the digraph $\vec{K}_n$

can be transformed into an instance of MHP(u,v,k) by adding a wl-dipath P of $k-(n\ mod\ k)$ new nodes (all arcs in P and arc (v,w) have cost 0, all other new arcs in the new complete directed graph have cost larger than $\sum_{e\in K_n}|c_e|$) and asking for a minimum cost Hamiltonian ul-dipath in the new complete directed graph.

From an instance of MHP(u,v,k), we construct a complete directed graph $\vec{K}_{n+1}$ as follows. Let S be a subset of the node set of $\vec{K}_n$ such that $u\notin S, v\notin S$ and $|S|=n(k-p)/k$. Introduce a new node z and join v to z and z to u by arcs of cost zero. Let $M>\sum_{e\in\vec{K}_n}|c_e|$ and let $M'>(n+1)M$. Join each node of S to z by arcs of cost $-M'$ and join z to each node of $\vec{K}_n$ except u by arcs of cost M. Also join each node not in $S\cup v$ to z by edges of cost M. All arcs of the constructed $\vec{K}_{n+1}$ corresponding to the original $\vec{K}_n$ keep their original costs (see Figure 2).

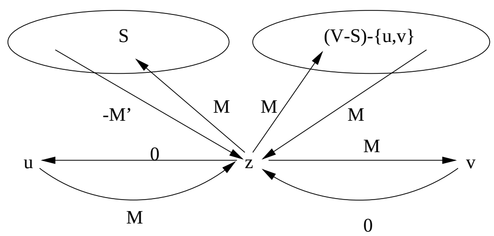

Fig. 2. Weights for arcs incident to z

Observe that an optimal Hamiltonian uv-dipath P with cost $c(P)$ in $\vec{K}_n$ can be extended to a tour in $\vec{K}_{n+1}$ by adding arcs (v,z) and (z,u) to it. Moreover, the only tours of $\vec{K}_{n+1}$ with a cost lower than $c(P)$ are tours using an arc (s,z) for some $s\in S$. There are exactly $|S|\cdot(n-1)!=n!(k-p)/k$ such tours. Thus $n!p/k$ tours of this $\vec{K}_{n+1}$ will have length greater than or equal to $c(P)$ and α could be used to find a solution to MHP(u,v,k). Since MHP(u,v,k) is NP-hard, the result follows. ■

Note that the above theorem does not rule out the possibility of a polynomial time algorithm with domination number at least $(n-1)!p/k$ for the ATSP.

7 Conclusion

In this paper we obtained the domination number of several popular heuristics for the TSP improving the best known domination numbers. We also gave upper bounds (unless P=NP) on the domination number of any polynomial time heuristic for the TSP. It is an open question to find a least upper bound on the domination number valid for all polynomial time heuristics for the TSP.

References

[1] Carlier J., Villon P., "A New Heuristic for the Traveling Salesman Problem", *RAIRO-Operations Research* 24 (1990), 245-253.

[2] Christofides N., "Worst-case Analysis of a New Heuristic for the Traveling Salesman Problem", Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA, (1976).

[3] Fischer S.T., "A note on the complexity of local search problems", *Information Processing Letters* 53 (1995), 69-75.

[4] M.R. Garey, D.S. Johnson, *Computers and Intractability, a Guide to the Theory of NP-completeness*, Freeman, 1979.

[5] Glover F., Punnen A.P., "The Traveling Salesman Problem: New Solvable Cases and Linkages with the Development of Approximation Algorithms", *Journal of the Operational Research Society* 48 (1997), 502-510.

[6] Glover F., "Ejection Chains, Reference Structures, and Alternating Path Algorithms for the Traveling Salesman Problem", Research Report, University of Colorado-Boulder, Graduate School of Business, (1992). {A short version appeared in *Discrete Applied Mathematics* 65 (1996), 223-253.}

[7] Grover L., Local Search and the Local Structure of NP-complete Problems, *Operations Research Letters* 12 (1992) 235-243.

[8] Gutin G. and Punnen A.P., *The traveling salesman problem and its variations, Kluwer Academic Publishers*, Boston, 2002.

[9] Gutin G., Yeo A., "TSP Heuristics with Large Domination Number", TR-12, Department of Mathematics and Statistics, Brunel University, UK, (1998).

[10] Gutin G., Yeo A., "TSP Tour Domination and Hamilton Cycle Decomposition of Regular Digraphs", *Operations Research Letters,* 28 (2001) 107-111.

[11] Gutin G., Yeo A., "Polynomial Algorithms for the TSP and the QAP with a Factorial Domination Number", *Discrete Applied Mathematics,* 119 (2002) 107-116.

[12] Gutin G., Yeo A., Zverovich A., "Exponential neighborhoods and domination analysis for the TSP", in *Traveling Salesman Problem and its Variations*, Gutin G and Punnen A (editors), Kluwer Academic Publishers, 2002.

[13] Gutin G., Yeo A., Zverovich A., Traveling salesman should not be greedy: domination analysis of greedy type heuristics for the TSP, To appear in *Discrete Applied Mathematics,* 117 (2002) 81-86.

[14] Klyaus P. S.,"The Structure of the Optimal Solution of Certain Classes of Travelling Salesman Problem", *Vestsi Akad. Nauk BSSR, Phys. Math. Sci., Minsk*, 1 (1976) 95-98.

[15] Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G., Shmoys D.B., *The Traveling Salesman Problem*, Wiley, New York, (1985).

[16] Lin S., Kernighan B.W., "An Effective Heuristic Algorithm for the Traveling Salesman Problem", *Operations Research* 21 (1973), 972-989.

[17] Papadimitriou C.H., Steiglitz K., *Combinatorial Optimization: Algorithms and Complexity*, Prentice Hall, NJ, 1982.

[18] Papadimitriou C.H., Vazirani U., "On two Geometric Problems Related to the Traveling Salesman Problem", *Journal of Algorithms* 5 (1984), 231-246.

[19] Punnen A.P., Kabadi S.N., "Domination Analysis of Some Heuristics for the Traveling Salesman Problem", *Discrete applied mathematics,* 119 (2002) 117-128.

[20] Punnen A.P., Glover F., "Implementing Ejection Chains with Combinatorial Leverage for the TSP", Research report, University of Colorado-Boulder, (1996).

[21] Punnen A.P., "The Traveling Salesman Problem: New Polynomial Approximation Algorithms and Domination Analysis", *Journal of Information and Optimization* 22 (2001) 191-206.

[22] Radzik T., "Parametric Flows, Weighted Means of Cuts and Fractional Combinatorial Optimization", 351-383, in *Complexity in Numerical Optimization*, Pardalos P.M. (editor), World Scientific, 1993.

[23] Rublineckii V.I., " Estimates of the Accuracy of Procedures in the Traveling Salesman Problem", (Russian), *Numerical Mathematics and Computer Technology* No. IV, (1973), 18-23.

[24] Sarvanov V.I., "The Mean Value of the Functional of the Assignment Problem", (Russian), *Vesc. Acad. Navuk BSSR* Ser. Fiz. -mat. Navuk 143, No.2 (1976), 111-114.

[25] Sarvanov V.I., Doroshko N.N., "Approximate Solution of the Traveling Salesman Problem by a Local Algorithm with Scanning Neighborhoods of Factorial Cardinality in Cubic Time", (Russian), *Software: Algorithms and Programs*, Mathematics Institute of the Belorussia Academy of Science, Minsk, No. 31 (1981) 11-13.

[26] Sahni S., Gonzalez T., "P-complete Approximation Problem", *Journal of ACM* 23 (1976), 555-565.

[27] Tillson T.W., "Hamiltonian Decomposition of K^*_{2m}, $2m \geq 8$", *Journal of Combinatorial Theory*, Ser. B, 29 (1980), 68-74.

[28] Vizing V.G., "Values of the Target Functional in a Priority Problem that are Majorized by the Mean Value", (Russian), *Kibernetika* (Kiev) (1973), 76-78.

[29] Yeo A., "Large Exponential Neighborhoods for the Traveling Salesman Problem", Preprint no. 47, Department of Maths and CS, Odense University, (1997).