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 a 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, the domination numbers of the $2-Opt$ and $3-Opt$, 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 number is 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 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 constant r and any constant k such that $k \equiv -1 \ mod \ (n+r)$. The complexity of finding the value of the median tour and of similar problems is 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

March, 2001

1 Introduction

Let $K_n = (V, E)$ be the complete graph on n vertices 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 $H^* \in \mathbb{F}$ such that $c(H^*) \leq c(H)$ for all $H \in \mathbb{F}$. Replacing K_n by a complete digraph $\vec{K}_n$ and asking for the directed tour of minimum cost in the digraph yields an *asymmetric traveling salesman* problem (ATSP). When a statement applies to both ATSP and STSP, we simply use traveling salesman problem (TSP). 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, let H_α be the solution produced by α and let H^* be an optimal solution. Assume that the cost vector c is restricted to the domain $\mathbb{D} \subseteq \mathbb{R}^m$. One measure of the worst-case performance of α on these instances is the *performance ratio*

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

(We assume that there exists $c \in \mathbb{D}$ such that $c(H^*) > 0$. If $c(H^*) \leq 0$ 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 the 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 $\delta \geq 1$ [23]. 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 [2]. The existence of an algorithm with a better performance ratio is an important open question.

Recently Glover and Punnen [3] proposed domination analysis as another approach to measure the quality of a heuristic algorithm α: Let $F_\alpha(c) = \{H \in \mathbb{F} : c(H) \geq c(H_\alpha)\}$. Define the *domination number* of α with respect to the domain $\mathbb{D}$ by

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

When $\mathbb{D} = \mathbb{R}^m$, $dom(\alpha, \mathbb{D})$ is denoted by $dom(\alpha)$. For any heuristic α, the domination number exists and it is at least one for the TSP, as the problem is always feasible. The definition extends directly to any minimization combinatorial problem with feasible set $\mathbb{F}$. If $dom(\alpha) = |\mathbb{F}|$, 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. 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 $\phi(c)$ 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 costs vectors c, the ranking of the solutions in $\mathbb{F}$ is the same with respect to c or to $d = \phi(c)$: $d(H_i) \leq d(H_j)$ 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 [14]) in which in every iteration the search neighborhood and tie breaking rule are independent of the actual value of the edge costs produces the same solution when applied to instances with edge costs c and d as defined above. It follows that the domination number of α remains the same with edge costs c or d. It should be noted that the stability of the domination number under order preserving transformations does not hold for most construction heuristics, an exception being the Patching heuristic [12].

Another advantage of domination analysis is that meaningful upper bounds on the domination number can be obtained even if the cost vector is not restricted. In fact we will show that for a large class of heuristics, the domination number with respect to the domain $\mathbb{R}^m$ is the same as the domination number with respect to the domain $\mathbb{R}^m_\Delta$. 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 [17].

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), denoted by $A(\vec{K}_n)$ (resp. $A(K_n)$), are given by [8,20,21]

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

Sarvanov [21] showed that, for an ATSP with n odd, there are at least $(n-2)!$ tours in $\vec{K}_n$ having objective function value greater than or equal to $A(\vec{K}_n)$. 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 $A(\vec{K}_n)$. 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 $A(\vec{K}_n)$. His proof and result for the case of odd n extends directly to the case for even n, using an Hamiltonian decomposition of $\vec{K}_n$ for even n. The existence of such a decomposition for $n \geq 8$ was proved by Tillson [24]. Recently, Gutin and Yeo [8] 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 $A(\vec{K}_n)$. As a direct consequence, it can be seen that there are $\frac{(n-2)!}{2}$ tours in K_n having objective function value at least $A(K_n)$ [20]. These results can be summarized as follows:

Theorem 1 *For a heuristic algorithm α for the ATSP and $n \neq 6$, if $c(H_\alpha) \leq A(\vec{K}_n)$, then $dom(\alpha) \geq (n-2)!$. In the case of STSP, if $c(H_\alpha) \leq A(K_n)$, then $dom(\alpha) \geq \frac{(n-2)!}{2}$.*

There are several polynomial time heuristic algorithms available for the TSP that produce solutions with objective function value at least as good as $A(K_n)$ (or $A(\vec{K}_n)$) [5,8,18,20–22,25] and hence the domination number of each of these heuristics is at least $(n-2)!$ for the ATSP and $(n-2)!/2$ for the STSP, 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 TSP is guaranteed to produce a solution with value at least $A(K_n)$ 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 [3] and its variations [17], and the Lin-Kernighan algorithm [13] are at least $(n-2)!/2$. Section 4 shows that the Node-Shifting heuristic may produce a solution worse than $A(K_n)$ and a data dependent bound on the objective function value of this solution is given. Further, we show that the $3-Opt$ heuristic produce a solution no worse than the average cost of all tours. In section 5 we consider domination analysis of the Christofides [2] and the Double Tree algorithms [12]. 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 a variation 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 $(n-1)! - k$ for any constant k or with domination number $(n-1)! - (\frac{k}{k+1}(n+r))! - 1$ for any constant r and any constant 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 $\lfloor (n-1)!p/k \rfloor$ tours where $p \in \{1, \ldots, k-1$ for any integer constant $k \geq 2$.

An open question is the existence of a polynomial time algorithm for the ATSP with domination number $(n-1)!k$ where $k > 1/2$. Gutin and Yeo [7] showed that if there exists a constant $r > 1$ such that for every sufficiently large k, every k-regular digraph with number of nodes $n < rk$ admits a Hamiltonian decomposition and if 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. Without loss of generality we assume that the nodes of the complete graph are numbered $\{1,\ldots,n\}$ and that all 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 [12]. Consider a Hamiltonian cycle $H=(1,2,\ldots,n,1)$ of K_n. 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 i of K_n, let $\delta_r=\sum_{j=1,r\neq j}^{n} c(r,j)$.

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

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

$$\Delta_i=-(n-3)c(i,i+1)-\sum_{j\in N_i}c(j,j+1)+\delta_i+\delta_{i+1},\ \text{for}\ i=1,2,\ldots,n.$$

Adding these n equations together, we get

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

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

Corollary 3 *The domination number of $2-Opt$ is at least $\frac{(n-2)!}{2}$.*

This follows from Theorems 1 and 2. Note that each edge in H can be exchanged with $(n-3)$ edges and $n(n-3)/2$ distinct 2-exchanges are possible for a given H. Thus, the average cost of a 2-exchange operation from tour H, denoted by $\overline{\Delta}(H)$, is given by $\frac{2\Delta}{n(n-3)}$. Hence

$$\overline{\Delta}(H)=\frac{4(n-1)}{n(n-3)}[A(K_n)-c(H)]\ . \tag{2}$$

Suppose we begin the $2-Opt$ heuristic with a tour H such that $c(H) > A(K_n)$. 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 (2) that this 2-exchange operation reduces the value of $c(H) - A(K_n)$ to $c(H_{ij}) - A(K_n)$, i.e. by a factor of $\frac{4(n-1)}{n(n-3)}$. To count the number of iterations needed to reach a solution with value at least $A(K_n)$, consider the order preserving transformation $\phi(x) = x - \frac{A(K_n)}{n}$ and let $d = \phi(c)$. The sequence of tours $H_1, H_2, \ldots, H_q$ 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 $c(H_0) > A(K_n)$ (implying $d(H_0) > 0$), we have

$$d(H_i) \leq (1 - \frac{4(n-1)}{n(n-3)})\ d(H_{i-1}) \quad \text{for } i = 1, 2, \ldots, q\ . \tag{3}$$

From equation (3), using a result of Grover [5] we get $q = O(n \log(d(H_0)))$, i.e. $q = O(n \log(c(H_0) - A(K_n)))$. We shall now get a strongly polynomial bound for q using the following result of Goemans, cited in [19].

Lemma 4 *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 $i = 1, 2, \ldots, (q-1)$, $0 \leq y_{i+1}c \leq \frac{1}{2}y_i c$, then $q = O(p \log p)$.*

Since

$$(1 - \frac{4(n-1)}{n(n-3)})^n \leq (1 - \frac{4}{n})^n \leq e^{-4}\ ,$$

from equation (3), we see that, after $O(n)$ 2-exchanges, we get a solution with value at most $d(H_0)/2$. Thus, using Lemma 4 with $p = n(n-1)/2$, we get $q = O(n^3 \log n)$. The forgoing discussion can be summarized as

Theorem 5 *For the STSP, the $2-Opt$ algorithm produces a solution with value at most $A(K_n)$ in $O(min\{n^3 \log n, n \log(c(H_0) - A(K_n))\})$ iterations.*

Note that the solution indicated in Theorem 5 need not be locally optimal with respect to the $2-Opt$ neighborhood.

Let $\phi : \mathbb{R} \longrightarrow \mathbb{R}$ be a mapping and let $d = \phi(c)$ for an arbitrary cost vector c. A heuristic algorithm α for the TSP is said to be *stable* with respect to ϕ (ϕ-stable for short) if and only if $c(H_\alpha) = d(H_\alpha)$. All improvement heuristics, such as the $2-Opt$ and $3-Opt$ heuristics, in which the neighborhood and tie-breaking rules are independent of edge costs are ϕ-stable for any order-preserving transformation ϕ. However, the Christofides heuristic is not ϕ-stable even for the transformation $\phi(c_{ij}) = \mu c_{ij} + a_i + b_j$ where $\mu > 0$ and $a_i, b_j \in \mathbb{R}$.

Theorem 6 *If α is a ϕ-stable heuristic for the TSP with $\phi(c(i,j)) = c(i,j) + M$ for $M \geq 3max\{|c(i,j)| : (i,j) \in E\}$, then $dom(\alpha, \mathbb{R}^m) = dom(\alpha, \mathbb{R}^m_\Delta)$.*

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

$$dom(\alpha, \mathbb{R}^m) \leq dom(\alpha, \mathbb{R}^m_\Delta) \ . \tag{4}$$

Let $c \in \mathbb{R}^m$. For $d = \phi(c)$, 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. $d \in \mathbb{R}^m_\Delta$. Since the transformation ϕ is order-preserving, we have

$$dom(\alpha, \mathbb{R}^m) \geq dom(\alpha, \mathbb{R}^m_\Delta) \ . \tag{5}$$

The result follows from (4) and (5). ■

Since $2 - Opt$ is ϕ-stable when ϕ is defined as in Theorem 6, it follows that imposing the triangle inequality on the cost vector will not improve its domination number.

3 Carlier-Villon Algorithm

Let the nodes of K_n be labeled 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-2} pyramidal tours in K_n with respect to a given node labeling and the best pyramidal tour can be obtained in $O(n^2)$ operations [11].

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: 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 labeling 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 [1]. 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 [9]. 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 $A(K_n)$ and the domination number of this algorithm is at least $\frac{(n-2)!}{2}$.*

The proof of this theorem follows from Lemma 7 and theorems 2 and 5. It may be noted that the previously best known domination number of the CV-algorithm was $n2^{n-2}$ [1]. The worst case complexity of this algorithm is not known to be polynomial. However, we have the following.

Theorem 9 *The CV-algorithm produces a solution with value no worse than $A(K_n)$ in polynomial time.*

The proof of Theorem 9 follows from Lemma 7 and Theorem 5. As in the case of Theorem 5, the solution indicated in Theorem 9 need not be locally optimal with respect to the CV(H) neighborhood.

Note that the CV-algorithm may be used on directed graphs too. However, the arguments used in proving theorems 8 and 9 are not valid for directed graphs.

Gutin and Yeo [6] 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 $dom(\alpha^*) \geq dom(\alpha)$: 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_α be the solution produced by α for this instance. Of the two tours in $\vec{K}_n$ corresponding to H_α, one in forward and other in backward direction, choose the one with lesser cost. As a consequence of this result and Theorem 6 we have,

Theorem 10 *The domination number of the CV-algorithm for the ATSP is at least $\frac{(n-2)!}{2}$.*

Using arguments similar to that in the proofs of Lemma 7 and Theorem 8, it can be shown that the domination numbers of Lin-Kernighan algorithm [13] and the Shortest Path Ejection Chain algorithm [4,17] are at least $\frac{(n-2)!}{2}$.

4 Node-shifting and 3-Opt heuristics

In this section we investigate the value of local optima with respect to neighborhoods that are subsets of the well known $3-Opt$ neighborhood [12]. 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 $A(\vec{K}_n)$ (or $A(K_n)$) 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

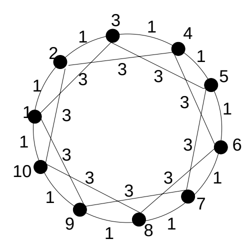

Fig. 1.

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 8.888 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 $(1, 3, \ldots n-2, n, 2, \ldots, n-1, u_1)$ 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 $\vec{\delta}_r = \sum_{j=1}^{n} c(r, j)$ and $\tilde{\delta}_r = \sum_{j=1}^{n} c(j, r)$.

Theorem 11 *Let H be a locally optimal solution with respect to the node-*

shifting neighborhood. Then for the ATSP, $c(H) \leq \frac{(n-2)}{3n-4}c(\hat{H}) + \frac{2(n-1)}{3n-4}A(\vec{K}_n)$ *and for the STSP,* $c(H) \leq \frac{(n-2)}{3n-4}c(\hat{H}) + \frac{2(n-1)}{3n-4}A(K_n)$*.*

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

$$c(H) \leq \frac{(n-2)c(\hat{H}) + 2c(\vec{K}_n)}{3n-4} \ . \tag{6}$$

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

Corollary 12 *If* $c(\hat{H}) \leq c(H)$ *then, for the STSP,* $c(H) \leq A(K_n)$ *and for the ATSP,* $c(H) \leq A(\vec{K}_n)$*.*

Corollary 13 *If the edge costs satisfy the triangle inequality, then* $c(H) \leq \frac{2(n-1)}{n}A(\vec{K}_n)$ *for the ATSP and* $c(H) \leq \frac{2(n-1)}{n}A(K_n)$ *for the STSP.*

Proof. If the edge costs satisfy the triangle inequality, $c(\hat{H}) \leq 2c(H)$ and the result follows from (6). ■

We shall now show that, by considering additional $3-Opt$ exchanges, a solution can be obtained with value at most $A(K_n)$. 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

$$\begin{aligned}\Lambda_{iu} = c(H_{iu}) - c(H) = &-c(i-1,i) - c(i,i+1) + c(i-1,i+1) \\ &+c(u,i) + c(i,u+1) - c(u,u+1) \ .\end{aligned} \tag{7}$$

Define

$$\begin{aligned}\Lambda^1_{iju} = c(H^1_{iju}) - c(H) = &\ c(i,j+1) + c(u,j) + c(i+1,u+1) \\ &-c(i,i+1) - c(j,j+1) - c(u,u+1)\end{aligned} \tag{8}$$

$$\begin{aligned}\Lambda^2_{iju} = c(H^2_{iju}) - c(H) = &\ c(i,u) + c(j+1,i+1) + c(j,u+1) \\ &-c(i,i+1) - c(j,j+1) - c(u,u+1)\end{aligned} \tag{9}$$

$$\begin{aligned}\Lambda^3_{iju} = c(H^3_{iju}) - c(H) = &\ c(i,j) + c(i+1,u) + c(j+1,u+1) \\ &-c(i,i+1) - c(j,j+1) - c(u,u+1) \ .\end{aligned} \tag{10}$$

Theorem 14 *If H is locally optimal with respect to the partial $3-Opt$ neighborhood, then $c(H) \leq A(K_n)$.*

Proof. If H is locally optimal with respect to the partial $3-Opt$ neighborhood, then, for all $1 \leq i < j < u \leq n$, $\Lambda_{iu} \geq 0$ and $\Lambda^k_{iju} \geq 0$ for $k = 1, 2, 3$. Adding all the inequalities (7), (8), (9), (10), we get, by symmetry,

$$-rc(H) + r_1 c(S_1) + \sum_{e \in S_2} r(e)c(e) \geq 0 \ , \tag{11}$$

for some r, r_1 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, $r = \frac{3(n-3)(n-4)}{2}$.

Let $e \in S_1$ with $e = (a, b)$ and assume w.l.o.g. that $b = a + 2 \ (mod\ n)$. 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 $r_1 = 2(n-5) + (n-2) = 3(n-4)$.

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, r_1 and $\{r(e) : e \in S_2\}$ in inequality (11) we get,

$$3(n-4)c(K_n) - \frac{3(n-4)(n-1)}{2}c(H) \geq 0 \ .$$

This implies that $c(H) \leq \frac{2c(K_n)}{(n-1)} = A(K_n)$. ■

Corollary 15 *If H is locally optimal for $3-Opt$, then $c(H) \leq A(K_n)$. Further the domination number of $3-Opt$ heuristic is at least $(n-2)!/2$*

The following theorem can be proved in essentially the same way as the similar theorem corresponding to the $2-Opt$ heuristic.

Theorem 16 *The $3-Opt$ heuristic produces a tour with cost no more than $A(K_n)$ in $O(\min\{n^3\log(n), n\log(c(H)-A(K_n))\})$ 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 TSP when the edge costs satisfy the triangle inequality. The algorithm constructs a minimum 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 an Hamiltonian tour C. The quality of C 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 [15].

Theorem 17 *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 TSP for which Christofides algorithm produces a tour that dominates only the claimed number of tours. Consider the case 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 produce 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 spanning tree 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 when the cost of edge ij in $G(T)$ becomes the cost of the shortest ij-path in K_n.

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

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

Theorem 18 *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 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 resultant 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 our 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, if the costs of edges in $G(T)$ are set to their original values, 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 [12]. 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 19 *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 TSP 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. ∎

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 $P \neq NP$.

Theorem 20 *Unless $P = NP$, there is no polynomial time approximation algorithm for ATSP with domination number $(n-1)! - k$ for any constant k.*

Proof. Assume that α is a polynomial time algorithm for the TSP with domination number $(n-1)! - k$ for some constant k. We show that this algorithm can be used to find a minimum cost Hamiltonian uv-path in a complete directed graph D for $u, v \in V(D)$ with edge cost vector d. Construct a corresponding instance of the TSP 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 edges e, e' with cost $c_e = c_{e'} = M$, where $M > \sum_{e \in D} |d_e|$. Edges joining any two nodes of D get their original cost except edge (u, v) with $c_{uv} = M$. All remaining edges of the complete directed graph on the node set $V(D) \cup V(D')$ have cost zero. Let D^* be the resulting directed graph.

If P is a minimum cost Hamiltonian uv-path in D with cost $d(P)$, 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 TSP on D^*. Furthermore, each optimal solution to the TSP 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 21 *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 constant r and any constant k with $k + 1 \equiv 0 \ mod(n + r)$.*

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. ■

By choosing appropriately the number of nodes in $D^{'}$ in the proofs of theorems 20 and 21, several related complexity results on domination analysis can be obtained.

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

Proof. Assume that there is a polynomial time algorithm α to compute the objective function value of a tour which dominates exactly $\lfloor (n-1)!p/k \rfloor$ tours. We show that this algorithm can be used to compute the value of a minimum cost Hamiltonian uv-path 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 value (u, v)-Hamiltonian path problem (MVHPP(u, v, k)).* It is not difficult to show that MVHPP(u, v, k) is NP-hard. From an instance of MVHPP(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 $M > 0$ be a number larger than the absolute value of the cost of any Hamiltonian dipath in the original graph (with free extremities) 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 $S \cup u$ to z by edges of cost zero. All edges of the constructed $\vec{K}_{n+1}$ corresponding to the original K_n keep their cost in $\vec{K}_n$.

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 edges vz and zu to it. Moreover, the only tours of $\vec{K}_{n+1}$ with a cost lower than $c(P)$ are tours using an edge sz 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 $c(P)$. Thus α can be used to find a solution to MVHPP(u,v,k). Since MVHPP(u, v, k) is NP-hard, the result follows. ∎

It may be noted that the above theorem does not rule out the possibility of a polynomial time algorithm with domination number at least $\lfloor (n-1)!p/k \rfloor$ 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 heuristic 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] 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.

[4] 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.}

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

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

[7] Gutin G., Yeo A., "TSP Tour Domination and Hamilton Cycle Decomposition of Regular Digraphs", To appear in Operations Research Letters.

[8] Gutin G., Yeo A., "Polynomial Algorithms for the TSP and the QAP with a Factorial Domination Number", To appear in *Discrete Applied Mathematics*.

[9] Gutin G., Yeo A., Zverovich A., "Polynomial Restriction Approach for the ATSP and STSP", in *Traveling Salesman Problem and its Variations*, Gutin G and Punnen A (editors), Kluwer Academic Publishers, (To appear).

[10] Kabadi S.N., Punnen A.P., "Traveling Salesman Problem with Two Distinct Tour Values", Working Paper 95-026, Faculty of Administration, University of New Brunswick, (1995).

[11] 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.

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

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

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

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

[16] Punnen A.P., Kabadi S.N., "Domination Analysis of Some Heuristics for the Traveling Salesman Problem", Revised for *Discrete applied mathematics.*

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

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

[19] 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.

[20] 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.

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

[22] 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.

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

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

[25] 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.

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