Chapter 6

EXPONENTIAL NEIGHBORHOODS AND DOMINATION ANALYSIS FOR THE TSP

Gregory Gutin
Department of Computer Science
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
G.Gutin@rhul.ac.uk

Anders Yeo
Department of Computer Science
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
anders@cs.rhul.ac.uk

Alexei Zverovitch
Department of Computer Science
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
A.Zverovitch@rhul.ac.uk

1.   Introduction, Terminology and Notation

In this section, we overview the main results on the topics of this chapter and give basic terminology and notation used throughout the chapter.

1.1.   Introduction

The purpose of this chapter is to introduce the reader to recently developed concepts and results on exponential (size) neighborhoods and domination analysis for the traveling salesman problem (TSP). Even though these topics are of certain practical relevance, we restrict our-

selves to the theoretical study. The body of computational experiments
with exponential neighborhoods is insufficient yet to carry out mean-
ingful comparisons between new and classical approaches; we refer the
reader to the papers [80, 162] and Chapter 9, where certain compu-
tational experience with exponential neighborhoods is reported. The
reader may consult Chapters 9 and 10 of this book for discussion of the
experimental performance of some heuristics studied in the domination
analysis part of this chapter.

It is worth noting that while the symmetric traveling salesman prob-
lem (STSP) can be considered, in many cases, as a subproblem of the
asymmetric traveling salesman problem (ATSP), sometimes this view is
too simplistic since the ATSP and STSP are defined on different graphs
– complete directed and undirected. Thus, in particular, the number
of tours in ATSP and STSP on n vertices is $(n-1)!$ and $(n-1)!/2$,
respectively. Therefore, while we will mostly consider the ATSP in this
chapter, we will provide a separate treatment of the STSP when needed.
We will use the term TSP when it is not important whether the ATSP
or STSP is under consideration.

Local search heuristics are among the main tools to compute near
optimal tours in large instances of the TSP in relatively short time, see,
e.g., Chapters 8, 9 and 10 of this book. In most cases the neighborhoods
used in the local search algorithms are of polynomial cardinality. One
may ask whether it is possible to have exponential size neighborhoods for
the TSP such that the best tour in such a neighborhood can be computed
in polynomial time. Fortunately, the answer to this question is positive.
(This question is far from being trivial for some generalizations of the
TSP, e.g. Deineko and Woeginger [248] conjecture that for the quadratic
assignment problem there is no exponential neighborhood "searchable"
in polynomial time.)

There are only a few papers on exponential neighborhoods published
before the 1990s: Klyaus [507], Sarvanov and Doroshko [743, 744] and
Gutin [414, 415]. In particular, [744] and [414] independently showed
the existence of $(n/2)!$-size neighborhood for the TSP with n vertices
(n is even). In this neighborhood, the best tour can be computed in
$O(n^3)$ time, i.e., asymptotically in at most the same time as a complete
iteration of 3-Opt, which finds the best tour among only $\Theta(n^3)$ tours.

Punnen [680] showed how to generalize the neighborhood from [414,
744] and Gutin [416] proved that one of Punnen's extensions provides
neighborhoods of size $\Theta(exp(\sqrt{n/2})(n/2)!/n^{1/4})$. We study basic results
on exponential neighborhoods in Section 2. We use the definition of
a neighborhood from [248], the only survey paper on the topic. No-
tice that while Deineko and Woeginger [248], in their own words, "only

scratched the surface" in their survey on the topic, we provide more detailed treatment of some exponential neighborhoods.

In Section 3, following Gutin and Yeo [421], we provide upper bounds on the size of ATSP neighborhood. In particular, we prove that there is no ATSP neighborhood of cardinality at least $\beta(n-k)!$ for any constant $\beta > 0$ and fixed integer k provided NP$\not\subseteq$P/poly. (We provide an informal description of the class P/poly in Section 3.2; for a formal introduction of the topic, see [82].)

While it is natural to study the possible cardinality of neighborhoods, it is clear that the size of a neighborhood is not the only parameter of importance. Indeed, the neighborhood introduced in [414, 744] does not perform well in computational practice. This may be a result of an unfortunate property of the neighborhood: many tours of the TSP are not reachable from each other under the structure imposed by this neighborhood. Carlier and Villon [162] showed that their neighborhood is much better in this respect: each tour can be reached from any other tour in at most logarithmic number (in n) of iterations if the choice of a tour at every iteration is "right". Gutin and Yeo [418] introduced a neighborhood structure, which makes the tours much closer: for every pair of tours T_1, T_5 there are three tours T_2, T_3, T_4 such that every T_i is in the neighborhood of T_{i-1}, $i = 2, 3, 4, 5$. (The neighborhoods in [418] are polynomially searchable.) We study the "closeness" topic in Section 4.

Chapters 9 and 10 consider experimental performance of TSP heuristics. While experimental analysis is of certain importance, it cannot cover all possible families of TSP instances and, in particular, it normally does not cover the most hard ones. Experimental analysis provides little theoretical explanation why certain heuristics are successful while some others are not. This limits our ability to improve on the quality and efficiency of existing algorithms. It also limits our ability to extend approaches successful for the TSP to other combinatorial optimization (CO) problems.

Approximation analysis is a frequently used tool for theoretical evaluation of CO heuristics. Let $\mathcal{H}$ be a heuristic for the TSP, and let $\mathcal{I}_n$ be the set of instances of the TSP of size n. In approximation analysis, we use the approximation ratio $r_{\mathcal{H}}(n) = \max\{f(I)/f^*(I) : I \in \mathcal{I}_n\}$, where $f(I)$ $(f^*(I))$ is the cost of the heuristic (optimal) tour. Unfortunately, in most of cases, estimates for $r_{\mathcal{H}}(n)$ are not constants and provide only a vague picture of quality of heuristics.

Domination analysis provides an alternative to approximation analysis. In domination analysis, we are interested in the number of feasible solutions that are worse or equal in quality to the heuristic one, which is

called the domination number of the heuristic solution. In many cases, domination analysis is very useful. In particular, some heuristics have domination number 1 for the TSP. In other words, those heuristics, in the worst case, produce the unique worst possible solution. At the same time, the approximation ratio is not bounded by any constant. In this case, domination number provides a far better insight into the performance of the heuristics.

Results on domination number of TSP heuristics are considered in Section 5. Domination number was formally introduced in a 1996 version of [680] and [382][1]. Interestingly, the first important results on domination number can be traced back to the 1970s, see Rublineckii [733] and Sarvanov [738]. The *domination number* domn$(\mathcal{H}, \mathcal{I})$ of a TSP heuristic $\mathcal{H}$ for a particular instance $\mathcal{I}$ of the TSP with n vertices is the number of tours in $\mathcal{I}$ which are at least as costly as the tour found by $\mathcal{H}$. The *domination number* domn$(\mathcal{H}, n)$ of $\mathcal{H}$ is the minimum of domn$(\mathcal{H}, \mathcal{I})$ over all instances $\mathcal{I}$ with n vertices. Since the ATSP on n vertices has $(n-1)!$ tours, an algorithm for the ATSP with domination number $(n-1)!$ is exact. The domination number of an exact algorithm for the STSP is $(n-1)!/2$. Similarly, one can define the domination number of heuristics for other CO problems.

Glover and Punnen [382] asked whether there exists a polynomial time STSP heuristic with domination number at least $(n-1)!/p(n)$, where $p(n)$ is a polynomial in n, provided P$\neq$NP. Answering the this question, Gutin and Yeo [423] introduced polynomial time heuristics for the ATSP with domination number at least $(n-2)!$. Two years after [423] was completed, we found out that Rublineckii [733] and Sarvanov [739] answered the above question already in the the 1970s by showing that certain polynomial time heuristics for the STSP and the ATSP are of domination number at least $(n-2)!$ when n is odd and $(n-2)!/2$ when n is even. Punnen, Margot and Kabadi [684] proved that the best improvement[2] versions of some well-known local search heuristics for the TSP after polynomial number of steps produce tours which are not worse than at least $\Omega((n-2)!)$ other tours. Punnen and Kabadi [683] obtained an $O(n^2)$ time heuristic with domination number at least $\sum_{k=1}^{n-2}(k!)$. Gutin and Yeo [420] investigated the existence of polynomial time heuristics with domination number $\Theta((n-1)!)$.

---

[1] Actually, an equivalent concept of domination ratio was introduced, which is the ratio of the domination number and the number of tours.

[2] During every iteration, best improvement local search algorithms compute the best tour in the current neighborhood.

Some heuristics may have a small domination number (thus, indicating that they are not useful, in general). For example, the "anti-greedy" heuristic for the ATSP that starts by choosing an arc of maximum cost and proceeds by choosing the most expensive arc among remaining eligible ones, is of domination number 1 (consider an instance with $c(i, i+1) = 1$ for every $i = 1, 2, ..., n-1$, $c(n, 1) = 1$, and $c(i, j) = 0$ for every $j \neq i + 1$ and $(i, j) \neq (n, 1)$). While the fact that the domination number of the anti-greedy heuristic equals one is quite expected, in Section 5, we prove that the same is true for the greedy and nearest neighbor algorithms for both the ATSP and STSP (these results were obtained by Gutin, Yeo and Zverovich in [424]). Punnen, Margot and Kabadi [684] proved that some other TSP algorithms are of very small domination number. In particular, they showed that the double tree heuristic and some variations of the Christofides heuristic for the STSP are of domination number 1.

In this chapter we discuss approaches and results obtained mostly in the last decade. Despite limited time and effort in the areas of domination analysis and exponential neighborhoods, one can clearly see that significant progress has already been made. Although some of the existing results and approaches have already been used in practice (see Chapter 9 and [80, 162, 377]), it seems that much more research is required before the above mentioned areas can be used to design new high quality heuristics for the TSP and other CO problems. We hope that this chapter will provide motivation for scholars and practitioners to continue studying the domination analysis and exponential neighborhoods for the TSP and other CO problems.

1.2.   Basic Terminology and Notation

Recall that the ATSP is stated as follows. Given a weighted complete digraph $(\overleftrightarrow{K}_n, c)$, find a Hamiltonian cycle in $\overleftrightarrow{K}_n$ of minimum cost. Here the cost function c is a mapping from $A(\overleftrightarrow{K}_n)$ to the set of reals. The cost of an arc (x, y) of $\overleftrightarrow{K}_n$ is $c(x, y)$. It is assumed that $V(\overleftrightarrow{K}_n) = \{1, 2, \ldots, n\}$. The mapping c can be determined by the cost matrix $[c_{ij}]$. The STSP is defined similarly with the only difference that the graph under consideration is complete undirected (denoted by K_n). In this case, the matrix $[c_{ij}]$ is symmetric. Unless it is specified otherwise, n is the number of vertices in the instance of the TSP under consideration.

Let $C = x_1 x_2 ... x_k x_1$ be a cycle in $\overleftrightarrow{K}_n$. The operation of *removal* of a vertex x_i $(1 \leq i \leq k)$ results in the cycle $x_1 x_2 ... x_{i-1} x_{i+1} \ldots x_k x_1$ (thus, removal of x_i is not deletion of x_i from C; *deletion* of x_i gives

the path $x_{i+1}x_{i+2}\ldots x_k x_1 x_2 \ldots x_{i-1}$). Let y be a vertex of $\overleftrightarrow{K}_n$ not in C. The operation of *insertion* of y into an arc (x_i, x_{i+1}) results in the cycle $x_1 x_2 \ldots x_i y x_{i+1} \ldots x_k x_1$. The *cost* of the insertion is defined as $c(x_i, y) + c(y, x_{i+1}) - c(x_i, x_{i+1})$. For a set $Z = \{z_1, \ldots, z_s\}$ ($s \leq k$) of vertices not in C, an *insertion* of Z into C results in the tour obtained by inserting the nodes of Z into different arcs of the cycle. In particular, insertion of y into C involves insertion of y into one of the arcs of C.

For a path $P = x_1 x_2 \ldots x_m$ in $(\overleftrightarrow{K}_n, c)$, the *contraction*[3] of P in $(\overleftrightarrow{K}_n, c)$, $(\overleftrightarrow{K}_n /P, c')$, is a complete digraph with vertex set

$$V(\overleftrightarrow{K}_n /P) = V(\overleftrightarrow{K}_n) \cup \{v_P\} - V(P),$$

where $v_P \notin V(\overleftrightarrow{K}_n)$, such that the cost $c'(u, w)$, for $u, w \in V(\overleftrightarrow{K}_n /P)$, is defined by $c(u, x_1)$ if $w = v_P$, $c(x_m, w)$ if $u = v_P$, and $c(u, w)$, otherwise. We can consider an arc $a = (x, y)$ as the path xy of length one; this allows us to look at $\overleftrightarrow{K}_n /a$ as a special case of the above definition. The above definition has an obvious extension to a set of vertex-disjoint paths.

Further definitions on directed and undirected graphs can be found in the corresponding appendix of this book; see also [84].

2.   Exponential Neighborhoods

We adapt the definition of a neighborhood for the TSP due to Deineko and Woeginger [248]. Let P be a set of permutations on n vertices. Then the *neighborhood* (with respect to P) of a tour $T = x_1 x_2 \ldots x_n x_1$ is defined as follows:

$$N_P(T) = \{x_{\pi(1)} x_{\pi(2)} \ldots x_{\pi(n)} x_{\pi(1)} : \pi \in P\}.$$

A *neighborhood structure* consists of neighborhoods for every tour T. The above definition of a neighborhood is quite restrictive[4] but reflects the very important "shifting" property of neighborhoods which distinguishes them from arbitrary sets of tours. Another important property usually imposed on a neighborhood $N(T)$ of a tour T is that the best among tours of $N(T)$ can be computed in time $p(n)$ polynomial in n. This is necessary to guarantee an efficient local search. Neighborhoods

---

[3]The operation is called path-contraction in [84], but since we do not consider any other type of contraction, we will use the shorter name.
[4]In particular, this definition implies that the neighborhood of every tour is of the same cardinality.

satisfying this property are called *polynomially searchable* or, more precisely, *p(n)-searchable.*

In the rest of this section and in Section 4, we only consider the ATSP: the neighborhoods we describe below can be readily adapted to the STSP.

2.1. The Pyramidal Neighborhood

In this subsection, we consider the pyramidal neighborhood introduced by Sarvanov and Doroshko [743]. Let $H = x_1x_2 \ldots x_nx_1$ be a tour. Define the *pyramidal neighborhood* of H, denoted by $PY(x_1, H)$, as follows.

A tour $G = x_{i_1}x_{i_2}x_{i_3}\ldots x_{i_n}x_{i_1}$, with $i_1 = 1$, belongs to $PY(x_1, H)$, if and only if there is an integer k, such that

$$i_1 < i_2 < \ldots < i_k > i_{k+1} > i_{k+2} > \ldots > i_n.$$

Observe that $i_k = n$. Note also that given $\{i_2, \ldots, i_{k-1}\}$ (together with H and x_1) G is uniquely determined, and given G, the set

$$FORW(G, H) = \{i_2, i_3, \ldots, i_{k-1}\}$$

is uniquely determined. The neighborhood structure is not symmetrical as if $H = x_1x_2x_3x_4x_1$ and $G = x_1x_3x_4x_2x_1$, then $G \in PY(x_1, H)$, but $H \notin PY(x_1, G)$. We first prove the well-known fact that the size of $PY(x_1, H)$ is exponential.

Theorem 1 $|PY(x_1, H)| = 2^{n-2}.$

Proof: As mentioned above the tours $G \in PY(x_1, H)$ are uniquely determined by $FORW(G, H)$, which is a subset of $\{2, 3, \ldots, n-1\}$ (of cardinality $n - 2$). Since any subset (including the empty set and the whole set) determines a tour in $PY(x_1, H)$, and there are 2^{n-2} such subsets, we are done. ∎

The fact that the pyramidal neighborhood can be searched in time $O(n^2)$ was proved for the first time by Klyaus [507]; for proofs of this assertion and its extensions, see Section 4 of Chapter 11.

Theorem 2 *We can find an optimal tour in $PY(x_1, H)$ in $O(n^2)$ time.*

Since every tour in $PY(x_1, H)$, when $H = x_1x_2\ldots x_nx_1$, either uses the arc x_1x_2 or the arc x_2x_1 (and either $x_{n-1}x_n$ or x_nx_{n-1}), the algorithm of Theorem 2 will not produce a good tour if these arcs are

expensive. One way of avoiding this problem is to consider the neighborhood $PCV(H) = \cup_{j=1}^{n} PY(j, H)$ instead (PCV stands for *pyramidal Carlier-Villon* as Carlier and Villon [162] introduced this neighborhood). Clearly, by Theorem 2, we can find an optimal tour in $PCV(H)$ in $O(n^3)$ time, by just running the algorithm of Theorem 2 n times.

It is not difficult to show that, for example, the well-known $2 - Opt$ neighborhood is a subset of PCV (i.e., $2 - Opt(H) \subset PCV(H)$) for the STSP. Deineko and Woeginger [248] proved that PCV covers at least 75 % of tours in 3-Opt. For some experimental results using PCV we refer the reader to [162].

2.2.   The Assign Neighborhood and Its Variations

For the special case of $|Z| = \lfloor n/2 \rfloor$ (see the definition of Z below), this neighborhood was introduced in [414, 744]. Punnen [680] introduced the general definition of this neighborhood as well as its further extension (see the last paragraphs of this subsection).

Let $T = x_1 x_2 ... x_n x_1$ be a tour and let $Z = \{x_{i_1}, x_{i_2}, ..., x_{i_s}\}$ be a set of non-adjacent vertices of T, i.e., $2 \leq |i_k - i_r| \leq n - 2$ for all $1 \leq k < r \leq s$. The *assign neighborhood* of T with respect to Z, $N(T, Z)$, consists of the tours that can be obtained from T by removal of the vertices in Z one by one followed by an insertion of Z into the cycle derived after the removal. (Recall that, by the definition of insertion of several vertices into a cycle C in Subsection 1.2, the vertices of Z are inserted into different arcs of C.) For example,

$$N(x_1 x_2 x_3 x_4 x_5 x_1, \{x_1, x_3\}) =$$
$$\{x_2 x_i x_4 x_j x_5 x_2, x_2 x_i x_4 x_5 x_j x_2, x_2 x_4 x_i x_5 x_j x_2 : \{i, j\} = \{1, 3\}\}.$$

Theorem 3 *[416, 680] The neighborhood $N(T, Z)$ is $O(n^3)$-searchable.*

Proof: Let $C = y_1 y_2 \ldots y_{n-s} y_1$ be the cycle obtained from T after removal of Z and let $Z = \{z_1, z_2, \ldots, z_s\}$. By the definition of insertion, we have $n - s \geq s$. Let ϕ be an injective mapping from Z to $Y = \{y_1, y_2, \ldots, y_{n-s}\}$. (The requirement that ϕ is injective means that $\phi(z_i) \neq \phi(z_j)$ if $i \neq j$.) If we insert some z_i into an arc (y_j, y_{j+1}), then the weight of C will be increased by $c(y_j, z_i) + c(z_i, y_{j+1}) - c(y_j, y_{j+1})$. Therefore, if we insert every z_i, $i = 1, 2, \ldots, s$, into $(y_{\phi(i)}, y_{\phi(i)+1})$, the weight of C will be increased by

$$g(\phi) = \sum_{i=1}^{s} c(y_{\phi(i)}, z_i) + c(z_i, y_{\phi(i)+1}) - c(y_{\phi(i)}, y_{\phi(i)+1}).$$

Clearly, to find a tour of $N(T, Z)$ of minimum weight, it suffices to minimize $g(\phi)$ on the set of all injections ϕ from Z to Y. This can be done using the following weighted complete bipartite graph B. The partite sets of B are Z and Y. The weight of an edge $z_i y_j$ is set to be $c(y_j, z_i) + c(z_i, y_{j+1}) - c(y_j, y_{j+1})$.

By the definition of B, every maximum matching M of B corresponds to an injection ϕ_M from Z to Y. Moreover, the weights of M and ϕ_M coincide. A minimum weight maximum matching in B can be found by solving the assignment problem. Therefore, in $O(n^3)$ time, we can find the best tour in $N(T, Z)$. ∎

Let $\text{ins}(n, s)$ be the number of tours in $N(T, Z)$, $s = |Z|$, and let $n \geq 5$. Since there are $k = n - s$ ways to insert x_1 in C, $k - 1$ ways to insert x_2 in C when x_1 has been inserted, etc., we obtain that $\text{ins}(n, s) = (n - s)(n - s - 1)...(n - 2s + 1)$. It is natural to ask what is the largest possible size of the assign neighborhood for the ATSP with n vertices. This question is answered in the following theorem. For a real r, $[r]_0$ ($[r]_1$, resp.) is the maximum integer (semi-integer, respectively) that does not exceed r (a semi-integer is a number of the form $p/2$, where p is an odd integer); for an integer m, $\sigma(m) = m \bmod 2$.

Theorem 4 *[416] For a fixed $n \geq 5$, the maximum size of the assign neighborhood equals*

$$\text{maxins}(n) = \frac{(n/2 + p_0)!}{(2p_0)!},$$

where $p_0 = \left[\sqrt{\frac{1}{8}(n + \frac{9}{8})} + \frac{3}{8}\right]_{\sigma(n)}$.

Proof: Assume first that n is even. Consider $f(p) = \text{ins}(n, n/2 - p)$, where p is a non-negative integer smaller than $n/2$. For $p \geq 1$, the difference $\Delta f(p) = f(p) - f(p-1) = b(-2p(2p-1) + (n/2+p)) = bq(p)/2$, where $q(p) = -8p^2 + 6p + n$, $b = (n/2 + p - 1)(n/2 + p - 2) \cdots (2p + 1)$. Clearly, $sign(\Delta f(p)) = sign(q(p))$. Therefore, $f(p)$ increases when $q(p) > 0$, and $f(p)$ decreases when $q(p) < 0$. For $p \geq 1$, $q(p)$ decreases and has a positive root $r = \sqrt{\frac{1}{8}(n + \frac{9}{8})} + \frac{3}{8}$. Thus, $f(p)$, where $p \in \{1, \ldots, n/2\}$ is maximum for $p = [r]_0$.

Analogously, when n is odd, we obtain that $f(p)$ is maximum for $p = [r]_1$. ∎

The following asymptotic formula provides us with an estimate on how large $\text{maxins}(n)$ is. Note that, for $2m \leq n \leq 2m+1$, $\text{ins}(n, m) = [\frac{n+1}{2}]_0!$.

Theorem 5 *[416] We have* $\text{maxins}(n) = \Theta\left(\dfrac{e^{\sqrt{n/2}}[\frac{n+1}{2}]_0!}{n^{\frac{1}{4}+[\frac{1}{2}]_\sigma(n)}}\right).$

The value of $\text{maxins}(n)$ is the maximum known size of a neighborhood searchable in time $O(n^3)$. We can combine several neighborhoods $N(T, Z)$ of T for various sets Z and construct a polynomially searchable neighborhood of size $\Theta(e^{\sqrt{n/2}}[\frac{n+1}{2}]!n^k)$ for every natural number k [416]. Do there exist larger polynomially searchable neighborhoods? Some stronger question is raised in Section 6.

For large values of n, the time $O(n^3)$ appears to be too high to be used in local search algorithms. Thus, the following result is of interest (observe that $(n-1)! = 2^{\Theta(n \log n)}$):

Theorem 6 *[416] 1. For every β, $0 < \beta \leq 2$, there is an $O(n^{1+\beta})$-algorithm for finding the best among $2^{\Theta(n \log n)}$ tours.*
  2. For every positive integer r there exists an $O(r^5 n)$-time algorithm for constructing the best among $\Omega(r^n)$ tours.

Corollary 12 in Subsection 3.1 of this chapter implies that the first part of this theorem cannot be, in some sense, improved.

Punnen [680] suggested an extension of the assign neighborhood. There we allow one to remove paths rather than vertices and insert them back. The rationale behind this extension is to preserve "good" parts of the current tour T. For example, one can use the following strategy: the cheaper an arc a in T the larger the probability to preserve a. The reader can easily add his/her own details to this approach. In practice this more general approach seems to be more promising.

A small number of computational experiments on a fairly straightforward implementation of a local search heuristic using Punnen's extension of the assign neighborhood were performed by Gutin, Punnen and Zverovich (unpublished). In general, the results appeared to be too modest in comparison to those of the state-of-the-art heuristics. To improve the results, one should probably combine Punnen's extension of the assign neighborhood with some "classical" neighborhoods.

2.3.   The Balas-Simonetti Neighborhood

The following neighborhood, was introduced by Balas [66] and studied computationally by Balas and Simonetti in [80]. Although this neighborhood has been defined for both ATSP and STSP, we consider here only the ATSP case. Let k be any integer with $2 \leq k \leq n$, let

$H = x_1 x_2 \ldots x_n x_1$ be a tour, and define the neighborhood of H, denoted by $BS_k(x_1, H)$, as follows (see [80]).

A tour $G = x_{\pi(1)} x_{\pi(2)} x_{\pi(3)} \ldots x_{\pi(n)} x_{\pi(1)}$ (with $\pi(1) = 1$) belongs to $BS_k(x_1, H)$ if and only if for all integers i and j with $j \geq i + k$ we have $\pi(i) < \pi(j)$.

In other words if a vertex, x_j, lies k or more places after a vertex x_i in H, then x_j must lie after x_i in G (when one walks along the tour, starting at x_1). Furthermore, the inequality $j \geq i + k$ is not taken modulo n, which is why the vertex x_1 has a special function in the above definition. The above neighborhood is not symmetric, as seen by the following example with $n = 5$ and $k = 3$. Let $H = x_1 x_2 x_3 x_4 x_5 x_1$, $G = x_1 x_4 x_2 x_5 x_3 x_1$, and note that $G \in BS_k(x_1, H)$, but $H \notin BS_k(x_1, G)$ as x_3 does not come after x_4 in H.

In the proof of Theorem 8, we will illustrate how to find an optimal solution in $BS_k(x_1, H)$ in $O(nk^2 2^k)$ time, by reducing the problem to a shortest path problem in an auxiliary digraph, G^*, with at most $nk(k + 1)2^{k-2}$ arcs. Note that for a fixed k this implies a linear algorithm, which turns out to be quite effective in practice [80].

To the best of our knowledge, the following theorem that provides bounds for the size of $BS_k(x_1, H)$ is a new result.

Theorem 7 *For $n \geq k(k + 1)$, $(\frac{k}{e})^{n-1} < |BS_k(x_1, H)| \leq k^{n-1}$. Furthermore, we have that $|BS_2(x_1, H)| = Fib(n)$, where $Fib(n)$ is the nth Fibonacci number.*

Proof: We will start by proving that $(\frac{k}{e})^n \leq |BS_k(x_1, H)|$. First assume that $n = ik + 1$, where i is an integer, and without loss of generality let $H = x_1 x_2 \ldots x_n x_1$. Define $\mathcal{F}$ to be the set of all tours of the form $x_{\pi(1)} x_{\pi(2)} x_{\pi(3)} \ldots x_{\pi(n)} x_{\pi(1)}$ with $\pi(1) = 1$ and $\{\pi(jk + 2), \pi(jk + 3), \ldots, \pi(jk + k + 1)\} = \{jk + 2, jk + 3, \ldots, jk + k + 1\}$, for all $j = 0, 1, \ldots, i - 1$. This means that we only allow tours that permute the first k vertices (not including x_1), the next k vertices, etc, but not vertices between these sets. Clearly the number of tours in $\mathcal{F}$ is $(k!)^i$ and $\mathcal{F} \subset BS_k(x_1, H)$. Using Stirling's formula we get $(k!)^i > (\sqrt{2\pi k} k^k e^{-k})^{(n-1)/k} > (k/e)^{n-1}$. This proves the case when $n - 1$ is divisible by k.

If $n = ik + j$, where $1 < j \leq k$, then we proceed as follows. We still have i sets of size k we may permute, but now we have $j - 1$ vertices left over (we do not count x_1). We choose the i sets as follows: $\{x_2, x_3, \ldots, x_{k+1}\}$ is the first set, x_{k+2} is a left-over vertex,

$$\{x_{k+3}, x_{k+4}, \ldots, x_{2k+2}\}$$

is the second set, x_{2k+3} is a left-over vertex, etc. After all $j - 1$ left-over vertices have been used, the sets will not have any vertices between them. Since $n \geq k(k+1)$ this can be done.

We can obtain a tour in $BS_k(x_1, H)$ by permuting the sets and then retaining every left-over vertex or inserting it in one of the places available. Since we have, on average, at least k possibilities for every left-over vertex, we obtain that

$$|BS_k(x_1, H)| > (k/e)^{ik} k^j > (k/e)^{n-1}.$$

We will now prove that $|BS_k(x_1, H)| \leq k^{n-1}$. We will build the tour starting at x_1, and show that we have at most k choices at each position. Assume that we have built a partial tour (i.e. a path), $x_1 x_{\pi(2)} x_{\pi(3)} \ldots x_{\pi(i)}$, and let l be the smallest index not used yet (i.e., $l = \min(\{1, 2, \ldots, n\} - \{1, \pi(2), \pi(3), \ldots, \pi(i)\})$). Clearly we can only place x_j in the $(i+1)$th position if $l \leq j \leq l + k - 1$. Therefore we get that $|BS_k(x_1, H)| \leq k^{n-1}$.

Finally we prove that $|BS_2(x_1, H)| = Fib(n)$ by induction. Note that $BS_2(x_1, H)$ contains all tours where we only have swapped positions of neighbors in H (and x_1 stays fixed). Observe that $|BS_2(x_1, H)| = Fib(n)$ holds for $n = 2$ and $n = 3$, and assume that it holds for $n - 1$ and $n - 2$, $n \geq 4$. Let $H = x_1 x_2 \ldots x_n x_1$, and note that by the induction hypothesis there are $Fib(n - 2)$ tours starting with $x_1 x_3 x_2$, and that there are $Fib(n - 1)$ tours starting with $x_1 x_2$. Since there are no other possibilities we get that $|BS_2(x_1, H)| = Fib(n-2) + Fib(n-1) = Fib(n)$
∎

Note that $Fib(n)$ is approximately $0.7236 \times 1.618^{n-1}$.

Theorem 8 *[66] We can find an optimum in $BS_k(x_1, H)$ in $O(nk^2 2^k)$ time.*

Proof: We transform the problem to a minimum cost path problem, in an auxiliary digraph $D_k(x_1, H)$, which we will simply denote by D_k. The vertices of D_k are tuples (i, j, S^-, S^+), such that there exists some tour $R = x_{\pi(1)} x_{\pi(2)} \ldots x_{\pi(n)} x_{\pi(1)} \in BS_k(x_1, H)$ ($\pi(1) = 1$) such that the following holds:

1. $\pi(i) = j$;

2. $S^- = \{\pi(1), \pi(2), \ldots, \pi(i-1)\} \cap \{i, i+1, \ldots, n\}$;

3. $S^+ = \{\pi(i), \pi(i+1), \ldots, \pi(n)\} \cap \{1, 2, \ldots, i-1\}$.

We furthermore say that the tuple (i, j, S^-, S^+) is compatible with the tour R. Note that $|S^-| = |S^+| (= i - 1 - |\{\pi(1), \pi(2), \ldots, \pi(i-1)\} \cap$

$\{1, 2, \ldots, i - 1\}|)$. An arc (x, y) is in D_k if $x = (i, j_x, S_x^-, S_x^+)$ and $y = (i+1, j_y, S_y^-, S_y^+)$, and there exists some tour $R = \pi(1)\pi(2) \ldots \pi(n)\pi(1)$, for which both x and y are compatible. Furthermore, if this is the case, then $S_y^- = S_x^- \cup (\{j_x\} \cap \{i+1, i+2, \ldots, n\}) - \{i\}$ and $S_y^+ = S_x^+ \cup (\{i\} \cap (V - S_x^-)) - \{j_x\}$, where $V = \{1, 2, \ldots, n\}$. Since the last two formulas can be proved similarly, we will show only the second one. It is straightforward to see that $S_y^+ = S_x^+ \cup (\{\pi(i), \pi(i+1), \ldots, \pi(n)\} \cap \{i\}) - \{j_x\}$. However,

$$
\begin{aligned}
\{\pi(i), \pi(i+1), \ldots, \pi(n)\} \cap \{i\} &= \\
(\{\pi(i), \pi(i+1), \ldots, \pi(n)\} \cup \{1, 2, \ldots, i-1\}) \cap \{i\} &= \\
(V - S_x^-) \cap \{i\}.
\end{aligned}
$$

We will use the fact that S_y^- and S_y^+ are totally determined by S_x^-, S_x^+, i and j_x several times below.

We will now show that there is a one-to-one correspondence between tours in $BS_k(x_1, H)$ and paths from $(1, 1, \emptyset, \emptyset)$ to $(n + 1, 1, \emptyset, \emptyset)$ in D_k. For an example, see Figure 6.1. Clearly any tour in $BS_k(x_1, H)$ has a corresponding path from $(1, 1, \emptyset, \emptyset)$ to $(n + 1, 1, \emptyset, \emptyset)$ in D_k, so now let P be a path from $(1, 1, \emptyset, \emptyset)$ to $(n + 1, 1, \emptyset, \emptyset)$ in D_k. Let $P = (1, 1, \emptyset, \emptyset)(2, \pi(2), S_2^-, S_2^+) \ldots (n, \pi(n), S_n^-, S_n^+)(n + 1, 1, \emptyset, \emptyset)$. Now we show that $Q = x_1 x_{\pi(2)} x_{\pi(3)} \ldots x_{\pi(n)} x_1$ is a tour in $BS_k(x_1, H)$.

Note that if R is a tour compatible with $(i, \pi(i), S_i^-, S_i^+)$, then one can uniquely determine the first i elements in R (but not their order), as they are the ones with the following indices, $S_i^- \cup (\{1, 2, \ldots, i-1\} - S_i^+) \cup \pi(i)$. We will now show by induction that $(i, \pi(i), S_i^-, S_i^+)$ is compatible with Q and $1, \pi(2), \pi(3), \ldots, \pi(i)$ are distinct. Clearly this is true for $i = 2$. So assume that it is true for $i - 1$ ($i \geq 3$). As $1, \pi(2), \pi(3) \ldots, \pi(i - 1)$ are uniquely determined by $(i - 1, \pi(i - 1), S_{i-1}^-, S_{i-1}^+)$, and there is an arc from $(i - 1, \pi(i - 1), S_{i-1}^-, S_{i-1}^+)$ to $(i, \pi(i), S_i^-, S_i^+)$ we must have that $\pi(i)$ is distinct from $1, \pi(2), \pi(3) \ldots, \pi(i - 1)$ (as there is a tour that is compatible with both $(i - 1, \pi(i - 1), S_{i-1}^-, S_{i-1}^+)$ and $(i, \pi(i), S_i^-, S_i^+)$). Furthermore S_i^- and S_i^+ are totally determined by S_{i-1}^-, S_{i-1}^+, $i - 1$ and $\pi(i - 1)$, so therefore $(i, \pi(i), S_i^-, S_i^+)$ is compatible with Q. This completes the inductive proof. It now follows that all $(1, 1, \emptyset, \emptyset), (2, \pi(2), S_2^-, S_2^+) \ldots (n + 1, 1, \emptyset, \emptyset)$ are compatible with Q and Q is a tour. Therefore Q is a tour in $BS_k(x_1, H)$.

Now by setting the cost of the arc xy, where $x = (i, j_x, S_x^-, S_x^+)$ and $y = (i+1, j_y, S_y^-, S_y^+)$, to the cost of the arc $x_{j_x} x_{j_y}$ the cost of a path from $(1, 1, \emptyset, \emptyset)$ to $(n + 1, 1, \emptyset, \emptyset)$ is equal to the cost of the corresponding tour given by this path. For an example, see the path illustrated by the thick rectangles in Figure 6.1, which corresponds to the tour $x_1 x_3 x_4 x_2 x_6 x_5 x_1$.
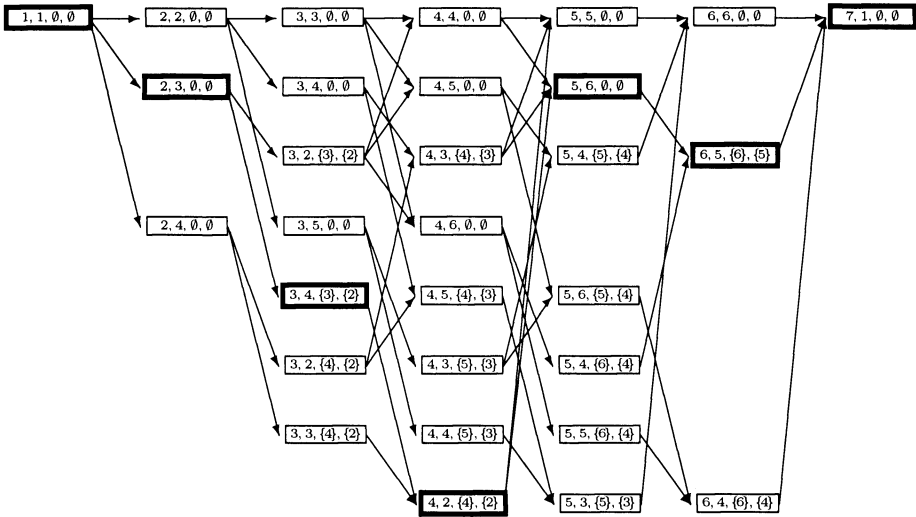
Figure 6.1. Example when $n = 6$ and $k = 3$. The path connecting the thick nodes correspond to the tour $x_1 x_3 x_4 x_2 x_6 x_5 x_1$.

In [66] Balas proved that the number of vertices in D_k of the form (i, j_x, S_x^-, S_x^+) is equal to $(k + 1)2^{k-2}$, for any given i, with $k + 1 \leq i \leq n - k + 1$. It is easy to see that for every remaining i there are at most $(k + 1)2^{k-2}$ such vertices. This implies that the total number of vertices in D_k is at most $n(k + 1)2^{k-2}$.

We will now prove that the out-degree of any vertex in D_k is at most k. Let $x = (i, j_x, S_x^-, S_x^+)$ be some vertex in D_k, and let $y = (i+1, l, S_y^-, S_y^+)$ be an out-neighbor of x. As mentioned above, S_y^- and S_y^+ are totally determined by S_x^-, S_x^+, i and j_x. Let $p = \min\{S_y^+ \cup \{i + 1\}\}$, (or $\min\{S_x^+ \cup \{i, i+1\} - j_x\}$, which is equivalent), and note that p is the smallest index, such that x_p is not used in the path $x_{\pi(1)} x_{\pi(2)} \cdots x_{\pi(i)}$. Now it is not difficult to see that $p \leq l \leq p + k - 1$. Therefore, the out-degree of x is at most k.

This implies that the number of arcs in D_k is bounded by $nk(k + 1)2^{k-2}$. So finding a cheapest path of length n in D_k can be done in $O(nk(k + 1)2^{k-2}) = O(nk^2 2^k)$ time. Since any tour in $BS_k(x_1, H)$ corresponds to a path of length n in D_k and any path of length n in D_k corresponds to a tour in $BS_k(x_1, H)$, we are done. ∎

The above algorithm can be generalized, such that the constant k depends on the position on the tour. That is, given a set of integers $\{k(i) : i = 1, 2, \ldots, n\}$, in the definition of the neighborhood, we have

that, if $j \geq i + k(i)$ then $\pi(i) < \pi(j)$. This generalization is not too difficult to implement and a description of this can be found in [80].

Furthermore, using the above generalizations the algorithm can be extended to time window problems as well as the time target problems. We refer the reader to [80] for more details.

The algorithm of this subsection has been tested in [80], and seems to work well as a local search algorithm. One may compare the algorithm in this subsection with k-Opt as they both perform local changes. However, the neighborhood described here has exponential size, and can be searched in linear time (for constant k), whereas k-Opt has polynomial size neighborhoods, and non-linear running time (in n). The algorithms of this subsection seem to perform particularly well on TSP instances that model actual cities and distances between cities. One reason for this could be that cities tend to cluster in metropolitan areas. For a more detailed discussion of this topic we refer the reader to [80].

Finally we note that the digraphs D_k, described in the proof of Theorem 8, can be computed using the values n and k, independently of the input $(\overleftrightarrow{K}_n, c)$. Then it remains to add the costs, when the input becomes known. This preprocessing may, in many cases, save considerable time as actually constructing the digraphs D_k is more time consuming than computing the shortest path in D_k.

3.   Upper Bounds for Neighborhood Size

The aim of this section is to provide upper bounds for ATSP neighborhood sizes. In Subsection 3.1 we prove upper bounds depending on the time to search the neighborhood. In Subsection 3.2 we obtain an upper bound of the size of polynomially searchable neighborhoods.

3.1.   General Upper Bounds

This subsection is based on [421]. The next theorem provides an upper bound on the size of an ATSP neighborhood depending on the time to search the neighborhood. It is realistic to assume that the search algorithm spends at least one unit of time on every arc that it considers.

Theorem 9 *Let N_n be an ATSP neighborhood that can be searched in time $t(n)$. Then $|N_n| \leq \max_{1 \leq n' \leq n}(t(n)/n')^{n'}$.*

Proof: Let $D = (\overleftrightarrow{K}_n, c)$ be an instance of the ATSP and let H be the tour that our search algorithm returns, when run on D. Let E denote the set of arcs in D, which the search algorithm actually examine; observe that $|E| \leq t(n)$ by the assumption above. Let the arcs of $A(H) - E$ have high enough cost and the arcs in $A(D) - E - A(H)$ have low enough

cost, such that all tours in N_n must use all arcs in $A(H) - E$ and no arc in $A(D) - E - A(H)$. This can be done as H has the lowest cost of all tours in N_n. Now let D' be the digraph obtained by contracting the arcs in $A(H) - E$ and deleting the arcs not in E, and let n' be the number of vertices in D'. Note that every tour in N_n corresponds to a tour in D' and, thus, the number of tours in D' is an upper bound on $|N_n|$. In a tour of D', there are at most $d^+(i)$ possibilities for the successor of a vertex i, where $d^+(i)$ is the out-degree of i in D'. Hence we obtain that

$$|N_n| \leq \prod_{i=1}^{n'} d^+(i) \leq \left(\frac{1}{n'} \sum_{i=1}^{n'} d^+(i) \right)^{n'} \leq \left(\frac{t(n)}{n'} \right)^{n'},$$

where we applied the arithmetic-geometric mean inequality.   ∎

Corollary 10 *Let N_n be an ATSP neighborhood that can be searched in time $t(n)$. Then $|N_n| \leq \max\{e^{t(n)/e}, (t(n)/n)^n\}$, where e is the basis of natural logarithms.*

Proof: Let $U(n) = \max_{1 \leq n' \leq n}(t(n)/n')^{n'}$. By differentiating $f(n') = (t(n)/n')^{n'}$ with respect to n' we can readily obtain that $f(n')$ increases for $1 \leq n' \leq t(n)/e$, and decreases for $t(n)/e \leq n' \leq n$. Thus, if $n \leq t(n)/e$, then $f(n')$ increases for every value of $n' < n$ and $U(n) = f(n) = (t(n)/n)^n$. On the other hand, if $n \geq t(n)/e$ then the maximum of $f(n')$ is for $n' = t(n)/e$ and, hence, $U(n) = e^{t(n)/e}$.   ∎

It follows from the proof of Corollary 10 that

Corollary 11 *For $t(n) \geq en$, we have $|N_n| \leq (t(n)/n)^n$.*

Note that the restriction $t(n) \geq en$ is important since otherwise the bound of Corollary 11 can be invalid. Indeed, if $t(n)$ is a constant, then for n large enough the upper bound implies that $|N_n| = 0$, which is not correct since there are neighborhoods of constant size that can be searched in constant time: consider a tour T, delete three arcs in T and add three other arcs to form a new tour T'. Clearly, the best of the two tours can be found in constant time by considering only the six arcs mentioned above. Notice that this observation was not taken into account in [248], where the bound $|N_n| \leq (2t(n)/n)^n$ was claimed. That bound is invalid for $t(n) \leq n/2$.

Corollary 10 immediately implies that linear-time algorithms can be used only for neighborhoods of size at most $2^{O(n)}$. Using Corollary 10, it is also easy to show the following:

Corollary 12 *The time required to search an ATSP neighborhood of size $2^{\Theta(n \log n)}$ is $\Omega(n^{1+\alpha})$ for some positive constant α.*

3.2. Upper Bounds for Polynomial Time Searchable Neighborhoods

Deineko and Woeginger [248] conjectured that there is no ATSP neighborhood of cardinality at least $\beta(n-1)!$ for any positive constant β provided P$\neq$NP. In this subsection based on [421] we prove that there is no ATSP neighborhood of cardinality at least $\beta(n-k)!$ for any constant $\beta > 0$ and fixed integer k provided NP$\nsubseteq$P/poly.

P/poly is a well-known complexity class in structural complexity theory, see e.g. [82], and it is widely believed that NP$\nsubseteq$P/poly for otherwise, as proved in the well-known paper by Karp and Lipton [499], it would imply that the so-called polynomial hierarchy collapses on the second level, which is thought to be very unlikely. The idea that defines P/poly is that, for each input size n, one is able to compute a polynomial-sized "key for size n inputs". This is called the "advice for size n inputs". It is allowed that the computation of this "key" may take time exponential in n (or worse). P/poly stands for the class of problems solvable in polynomial time (in input size n) given the poly-sized general advice for inputs of size n. For formal definitions of P/poly and related non-uniform complexity classes, consult [82].

Let S be a finite set and $\mathcal{F}$ be a family of subsets of S such that $\mathcal{F}$ is a *cover* of S, i.e., $\cup\{F : F \in \mathcal{F}\} = S$. The well-known *covering problem* is to find a cover of S containing the minimum number of sets in $\mathcal{F}$. While the following *greedy covering algorithm* (GCA) does not always produce a cover with minimum number of sets, GCA finds asymptotically optimal results for some wide classes of families, see e.g. [526]. GCA starts by choosing a set F in $\mathcal{F}$ of maximum cardinality, deleting F from $\mathcal{F}$ and initiating a "cover" $\mathcal{C} = \{F\}$. Then GCA deletes the elements of F from every remaining set in $\mathcal{F}$ and chooses a set H of maximum cardinality in $\mathcal{F}$, appends it to $\mathcal{C}$ and updates $\mathcal{F}$ as above. The algorithm stops when $\mathcal{C}$ becomes a cover of S. The following lemma have been obtained independently by several authors, see Proposition 10.1.1 in [47].

Lemma 13 *Let $|S| = s$, let $\mathcal{F}$ contain f sets, and let every element of S be in at least δ sets of $\mathcal{F}$. Then the cover found by GCA is of cardinality at most $1 + f(1 + \ln(\delta s/f))/\delta$.*

Using this lemma we can prove the following:

Theorem 14 *Let $\mathcal{T}$ be the set of all tours of the ATSP on n vertices. For every fixed integer $k \geq 1$ and constant $\beta > 0$, unless NP$\subseteq$ P/poly, there is no set Π of permutations on $\{1, 2, \ldots, n\}$ of cardinality at least $\beta(n - k)!$ such that every neighborhood $N_{\Pi}(T)$, $T \in \mathcal{T}$, is polynomial time searchable.*

Proof: Assume that, for some $k \geq 1$ and $\beta > 0$, there exists a set Π of permutations on $\{1, 2, \ldots, n\}$ of cardinality at least $\beta(n - k)!$ such that every neighborhood $N_{\Pi}(T)$, $T \in \mathcal{T}$, is polynomial time searchable. Let $\mathcal{N} = \{N_{\Pi}(T) : T \in \mathcal{T}\}$. Consider the covering problem with $S = \mathcal{T}$ and $\mathcal{F} = \mathcal{N}$. Observe that $|S| = |\mathcal{F}| = (n-1)!$. To see that every tour is in at least $\delta = (n - k)!$ neighborhoods of $\mathcal{N}$, consider a tour $Y = y_1 y_2 \ldots y_n y_1$ and observe that for every $\pi \in \Pi$,

$$Y \in N_{\Pi}(y_{\pi^{-1}(1)} y_{\pi^{-1}(2)} \cdots y_{\pi^{-1}(n)} y_{\pi^{-1}(1)}).$$

By Lemma 13 there is a cover $\mathcal{C}$ of S with at most $O(n^k \ln n)$ neighborhoods from $\mathcal{N}$. Since every neighborhood in $\mathcal{C}$ is polynomial time searchable and $\mathcal{C}$ contains only polynomial number of neighborhoods, we can construct the best tour in polynomial time provided $\mathcal{C}$ is found. To find $\mathcal{C}$ (which depends only on n, and not on the instance of the ATSP) we need exponential time and, thus, the fact that the best tour can be computed in polynomial time implies that NP$\subseteq$ P/poly. ∎

4.   Diameters of Neighborhood Structure Digraphs

The *distance* from a vertex x to a vertex y of an unweighted digraph D is 0 if $x = y$, the length of the shortest path from x to y, if D has one, and ∞, otherwise. The *diameter* of a digraph D is the maximum distance in D. Given neighborhood $N(T)$ for every tour T in $\overleftrightarrow{K}_n$ (i.e., a neighborhood structure), the corresponding *neighborhood digraph* (of order $(n-1)!$) is a directed graph with vertex set consisting of all tours in $\overleftrightarrow{K}_n$ and arc set containing a pair (T', T'') if and only if $T'' \in N(T')$. The diameter of the neighborhood graph is one of the most important characteristics of the neighborhood structure and the corresponding local search scheme [162, 248, 372]. Clearly, a neighborhood structure with a neighborhood digraph of smaller diameter seems to be more powerful than one with a neighborhood digraph of larger diameter, let alone a neighborhood structure whose digraph has infinite diameter (in the last case, some tours are not "reachable" from the initial tour during local search procedure).

4.1. Diameters of Pyramidal and the Balas-Simonetti Neighborhood Digraphs

If the diameter of the pyramidal neighborhood digraph is d_{PY}, then Theorem 1 implies that $(2^{n-2})^{d_{PY}} \geq (n-1)!$ and, thus, $d_{PY} = \Omega(\log n)$. The next theorem implies that $d_{PY} = \Theta(\log n)$.

Theorem 15 *[162] The diameter of the neighborhood digraph corresponding to $PY(x_1, H)$ is at most $\lceil \log_2 n \rceil$.*

Observe that this theorem implies that the diameter d_{PCV} of the pyramidal Carlier-Villon neighborhood introduced in the end of Subsection 2.1 is also $\Theta(\log n)$. Indeed, $|PCV(H)| \leq n2^{n-2}$ and, thus, $d_{PCV} = \Omega(\log n)$. On the other hand, $PY(x_1, H) \subseteq PCV(H)$ and, hence, $d_{PCV} \leq d_{PY}$. The next theorem is a new result.

Theorem 16 *The neighborhood digraph of $BS_k(x_1, H)$ is of diameter $O(n)$.*

Proof: Since $BS_k(x_1, H)$ includes $BS_2(x_1, H)$ for every $k \geq 2$, it suffices to prove this theorem for $k = 2$. Let $H = x_1 x_2 \ldots x_n x_1$ and $G = x_{\pi(1)} x_{\pi(2)} \ldots x_{\pi(n)} x_{\pi(1)}$. We will show that there is a sequence of tours $G = G_1, G_2, \ldots, G_{n+1} = H$, such that $G_{i+1} \in BS_2(x_1, G_i)$, $i = 1, 2, \ldots, n$.

We find G_i as follows. When i is even, and $G_{i-1} = x_1 x_{z_2} \ldots x_{z_n} x_1$ then let $G_i = x_1 x_{w_2} \ldots x_{w_n} x_1$ such that the following holds. The first two vertices on G_i are a sorted version of the first two vertices on G_{i-1} (i.e., $\{x_{z_1}, x_{z_2}\} = \{x_{w_1}, x_{w_2}\}$ and $w_1 < w_2$), the next two vertices are a sorted version of the next two vertices on G_{i-1}, etc. When i is odd, we leave the first vertex unchanged, but then the next two vertices are a sorted version of the next two vertices on G_{i-1}, etc.

Observe that $G_i \in BS_k(x_1, G_{i-1})$ holds. The claim that $G_{n+1} = H$ is equivalent to the assertion that tours $G_1, G_2, \ldots, G_{n+1}$ "sort" numbers $\pi(1), \pi(2), \ldots, \pi(n)$ (to $1, 2, \ldots, n$). It remains to observe that this assertion follows from Part (c) of Problem 28-1 in [218], p. 651, i.e., from the fact that every odd-even sorting network is a sorting network. The details are left to the interested reader. ∎

The result of this theorem can be improved to $O(n/k)$. We leave details to the interested reader.

4.2. Diameter of Assign Neighborhood Digraphs

For a positive integer $k \leq n/2$, the neighborhood digraph $\Gamma(n, k)$ of the assign neighborhood has vertex set formed by all tours in $\overleftrightarrow{K}_n$. An

arc (T, R) is in $\Gamma(n, k)$ if there exists a set Z of k non-adjacent vertices of T such that $R \in N(T, Z)$. Clearly, (T, R) is in $\Gamma(n, k)$ if and only if (R, T) is in $\Gamma(n, k)$, i.e., $\Gamma(n, k)$ is symmetric. We denote by $\text{dist}_k(T, R)$ the distance (i.e., the length of a shortest path) from T to R in $\Gamma(n, k)$.

For a tour T in $\overleftrightarrow{K}_n$, let $\mathcal{I}_{nk}$ denote the family of all sets of k non-adjacent vertices in T. Clearly, the neighborhood $N_k(T)$ of a tour T in $\Gamma(n, k)$ equals

$$\cup_{Z \in \mathcal{I}_{nk}} N(T, Z).$$

Thus if, for some k, $i(n, k) = |\mathcal{I}_{nk}|$ is polynomial in n, then from the fact that $N(T, Z)$ is polynomially searchable it follows that $N_k(T)$ is polynomially searchable. Otherwise, $N_k(T)$ may be non-polynomially searchable. Since polynomially searchable $N_k(T)$ are of our interest, we start with evaluating $i(n, k)$ in Theorem 17. It follows from Theorem 17 that, for fixed k, $i(n, k)$ and $i(n, n - k)$ are polynomial.

Theorem 17 *[418]* $i(n, k) = \binom{n-k}{k} + \binom{n-k-1}{k-1}$.

Corollary 18 *[418] If p is a non-negative fixed integer, then $N_{p+1}(T)$ and $N_{\lfloor (n-p)/2 \rfloor}(T)$ are polynomially searchable ($p < \lfloor n/2 \rfloor$).*

Proof: This follows from Theorem 17 taking into consideration that $\binom{m}{k} = \binom{m}{m-k}$. ∎

One can easily prove that if n is even, then $\Gamma(n, n/2)$ consists of an exponential number of strongly connected components and, thus, its diameter is infinite (for example, $x_1 x_2 ... x_n x_1$ and $x_1 ... x_{n-2} x_n x_{n-1} x_1$ belong to different strong components of this digraph). Therefore, below we consider $\Gamma(n, k)$ for $k < n/2$ only.

Theorem 19 $\text{diam}(\Gamma(n, \lfloor (n - 1)/2 \rfloor)) \leq 4$.

Proof: We assume that $n \geq 5$, as for $2 \leq n \leq 4$ this claim can be verified directly. Let $C = x_1 x_2 ... x_n x_1$ and $T = y_1 y_2 ... y_n y_1$ be a pair of distinct tours in $\overleftrightarrow{K}_n$. Put $k = \lfloor (n - 1)/2 \rfloor$. We will prove that $\text{dist}_k(T, C) \leq 4$, thus showing that $\text{diam}(\Gamma(n, k)) \leq 4$.

We call a vertex v *even* (*odd*) with respect to C if $v = x_j$, where $1 \leq j \leq n$ and j is even (odd). For a set of vertices X of $\overleftrightarrow{K}_n$, let X_{odd} (X_{even}) be the set of odd (even) vertices in X.

First we consider the case of even n, i.e. $k = n/2 - 1$. The proof in this case consists of two steps. At the first step, we show that there exists a tour T'' whose vertices alternate in parity and such that $\text{dist}_k(T, T'') \leq 2$. Moreover, T'' has a pair of consecutive vertices which are also consecutive

in C. At the second step, we will see that $\text{dist}_k(T'', C) \leq 2$ as the odd and even vertices of T'' (except for the vertices of the above pair) can be separately reordered to form C. Thus, we will conclude that $\text{dist}_k(T, C) \leq 4$. Now, we proceed with the proof.

Clearly, T has a pair y_j, y_{j+1} such that y_{j+1} is odd and y_j is even. Let

$$Z = \{y_{j+2}, y_{j+4}, \ldots, y_{j+2k}\}$$

and let $|Z_{odd}| = s$. Remove the vertices of Z from T and then insert the s odd vertices of Z into the arcs $y_{j+1}y_{j+3}, \ldots, y_{j+2s-1}y_{j+2s+1}$ and $k - s$ even vertices of Z into the arcs

$$y_{j+2s+1}y_{j+2s+3}, y_{j+2s+3}y_{j+2s+5}, \ldots, y_{j+2k-1}y_{j+2k+1}.$$

We have obtained a tour

$$T' = y_j y_{j+1} v_{j+2} y_{j+3} v_{j+4} y_{j+5} \cdots y_{j+2k-1} v_{j+2k} y_{j+2k+1} y_j,$$

where $\{v_{j+2}, \ldots, v_{j+2k}\} = Z$.

Let $Z' = \{y_{j+3}, y_{j+5}, \ldots, y_{j+2k+1}\}$ and let $|Z'_{even}| = t$. Since the number of odd vertices in $V(\overleftrightarrow{K}_n) - \{y_j, y_{j+1}\}$ is equal to $k = |Z_{odd}| + |Z'_{odd}| = s + k - t$, we obtain that $s = t$. Remove Z' from T' and insert the t even vertices of Z' into the arcs $y_{j+1}v_{j+2}, v_{j+2}v_{j+4}, v_{j+6}v_{j+8}, \ldots, v_{j+2s-2}v_{j+2s}$ and the $k - s$ odd vertices of Z' into the arcs

$$v_{j+2s+2}v_{j+2s+4}, \ldots, v_{j+2k-2}v_{j+2k}, v_{j+2k}y_j.$$

We have derived a tour $T'' = u_1 u_2 \ldots u_n u_1$. Clearly, the vertices of T'' alternate in parity, i.e., for every m, if u_m is odd, then u_{m+1} is even.

Now we prove that the processes of insertion of Z and Z' can be performed in such a way that T'' contains a pair of consecutive vertices which are also consecutive in C (i.e. there exist indices p and q such that $u_p = x_q$ and $u_{p+1} = x_{q+1}$). Since $1 < |Z'| < n$, there exists a pair of distinct indices i, m such that $x_i, x_m \in Z'$ and $x_{i+1}, x_{m-1} \notin Z'$. Without loss of generality, we assume that i is odd. We consider two cases.

Case 1: $|Z'_{odd}| \geq 2$. We prove that we may choose index $q = i$. Since $x_{i+1} \notin Z'$ and $i + 1$ is even, either $y_j = x_{i+1}$ or $x_{i+1} \in Z_{even}$. If $x_{i+1} \in Z_{even}$, in the process of insertion of Z, we insert x_{i+1} into $y_{j+2k-1}y_{j+2k+1}$, i.e. $x_{i+1} = v_{j+2k}$. In the process of insertion of Z', we insert x_i into $v_{j+2k}y_j$ if $x_{i+1} = y_j$ or into $v_{j+2k-2}v_{j+2k}$, otherwise (i.e. $x_{i+1} = v_{j+2k}$).

Case 2: $|Z'_{odd}| = 1$. Thus, m is even. Since $n \geq 6$, it follows that $|Z'_{even}| \geq 2$. Analogously to Case 1, one may take $q = m - 1$.

Therefore, without loss of generality, we assume that $u_{n-1} = x_i$, $u_n = x_{i+1}$. Since $\{u_2, u_4, \ldots, u_{2k}, x_{i+1}\} = C_{even}$, we can delete $\{u_2, \ldots, u_{2k}\}$ from T'' and insert it into the obtained cycle to get the tour C' given by $C' = u_1 x_{i+3} u_3 x_{i+5} u_5 \ldots u_{2k-1} x_{i-1} u_{n-1} x_{i+1} u_1$. Analogously, we can delete $\{u_1, u_3, \ldots, u_{2k-1}\}$ from C' and insert it into the obtained cycle to get C. We conclude that $\text{dist}_k(T, C) \le 4$.

Now let n be odd; then $k = (n-1)/2$. Notice that, without loss of generality, we may assume that $x_n = y_n$ (to fix the initial labelings of T and C). Consider tours $X = x_1 x_2 \ldots x_n x_{n+1} x_1$ and $Y = y_1 y_2 \ldots y_{n-1} y_n y_{n+1} y_1$ in $\overleftrightarrow{K}_{n+1}$, where $y_n = x_n$, $y_{n+1} = x_{n+1}$. If we assume that $j = n$, $j+1 = n+1$, we can obtain, analogously to the case of even n, a tour Y'' such that the vertices of Y'' alternate in parity (with respect to their indices in X), x_{n+1} follows x_n in Y'' and $\text{dist}_k(Y, Y'') \le 2$. Now if $i = n$ and $i+1 = n+1$, then we can show, similarly to the case of even n, that $\text{dist}_k(Y'', X) \le 2$ and, thus, $\text{dist}_k(Y, X) \le 4$. Notice that, in the whole process of constructing X from Y, we have never removed x_n and x_{n+1} or inserted any vertex into the arc $x_n x_{n+1}$. Thus, we could contract the arc $x_n x_{n+1}$ to x_n and obtain C from T in four "steps". This shows that $\text{dist}_k(T, C) \le 4$.   ∎

We can extend Theorem 19 using the following:

Theorem 20 *[418] Let $\text{dist}_k(T, C) = 1$ for tours T and C and let m be an integer smaller than k. Then, $\text{dist}_m(T, C) \le \lceil k/m \rceil$.*

Corollary 21 *For every positive m,*

$$\text{diam}(\Gamma(n, m)) \le 4\lceil \lfloor (n-1)/2 \rfloor / m \rceil.$$

In particular, if p is a positive integral constant, then $\text{diam}(\Gamma(n, \lfloor (n-p)/2 \rfloor)) \le 8$ for every $n \ge 2p + 1$.

Proof: The first inequality follows directly from the above two theorems and the triangle inequality for distances in graphs. The first inequality implies the second one. Indeed, $n \ge 2p + 1$ implies

$$\frac{(n-1)/2}{(n-p-1)/2} \le 2, \quad \frac{\lfloor (n-1)/2 \rfloor}{\lfloor (n-p)/2 \rfloor} \le 2.$$

∎

5.   Domination Analysis

Recall that the domination number, $\text{domn}(\mathcal{H}, n)$, of a heuristic $\mathcal{H}$ for the TSP is the maximum integer $k = k(n)$ such that, for every instance

$\mathcal{I}$ of the TSP on n vertices, $\mathcal{H}$ produces a tour T which is not worse than at least k tours in $\mathcal{I}$ including T itself.

In this section, we describe some important results in domination analysis of TSP heuristics. In Subsection 5.1, domination numbers of ATSP and STSP heuristics are compared. In Subsection 5.2, we consider TSP heuristics of large domination number, at least $\Omega((n-2)!)$. It turns out that several well-known heuristics have a large domination number. In Subsection 5.3 we briefly discuss bounds on the largest possible domination number of a polynomial time TSP heuristic. TSP heuristics of small domination number are considered in Subsection 5.4. It is somewhat surprising that such heuristics as the greedy, nearest neighbor and double tree algorithms are all of domination number 1.

5.1.   Domination Number of Heuristics for the STSP and ATSP

In this subsection we observe that, in certain cases (e.g., for lower bounds on domination number), it is enough to study heuristics for the ATSP since one can readily obtain similar results on heuristics for the STSP from the corresponding ones for the ATSP. This justifies that we mostly study ATSP heuristics in this section. We also prove an assertion that relates the maximum possible domination numbers of polynomial time heuristics for the ATSP and STSP.

For a tour $H = x_1 x_2 \ldots x_n x_1$ in $\overleftrightarrow{K}_n$, the tour $x_n x_{n-1} \ldots x_1 x_n$ will be denoted by $\overline{H}$.

Since an instance of the STSP can be transformed into an "equivalent" instance of the ATSP by replacing every edge xy of K_n by the pair xy, yx of arcs of costs equal to the cost of the edge xy, every heuristic for the ATSP can be used for the STSP[5]. Observe that a polynomial time heuristic $\mathcal{A}$ for the ATSP with domination number $d(n)$ has domination number at least $d(n)/2$ for the STSP. The factor $\frac{1}{2}$ is due to the fact that a pair $Q, \overline{Q}$ of tours in $\overleftrightarrow{K}_n$ is indistinguishable in K_n.

One of the central natural questions on the domination number is to determine the maximum domination number of a polynomial time heuristic for the ATSP. We call it the *maximum domination number of the ATSP*. We can introduce the similar parameter for the STSP. The STSP being, in a sense, a special case of the ATSP, one may suspect that the maximum domination number of the STSP is larger than that of the ATSP. We will now show that this is not true.

---

[5]This, in particular, allows one to apply ATSP heuristics to the STSP without redefining them, see, e.g. Subsection 5.4.

Theorem 22 *[417] For every polynomial heuristic $\mathcal{H}$ for the STSP, there is a polynomial heuristic $\mathcal{H}'$ for the ATSP such that* $\text{domn}(\mathcal{H}', n) \geq \text{domn}(\mathcal{H}, n)$.

Proof: To an instance of ATSP with cost function c assign an instance of STSP defined on the same set of vertices and with cost function c' defined by $c'(x, y) = \frac{1}{2}(c(x, y) + c(y, x))$ for every $x \neq y$. Let $T = x_1 x_2 ... x_n x_1$ be a tour found by the heuristic $\mathcal{H}$ applied to (K_n, c') and let $\mathcal{S}$ be the set of all tours R in (K_n, c') such that $c'(T) \leq c'(R)$. The cycle $T = x_1 x_2 ... x_n x_1$ can be considered as a tour in $(\overleftrightarrow{K}_n, c)$. For a tour Q in $(\overleftrightarrow{K}_n, c)$, let Q^-, Q^+ be defined as follows:

$$\{Q^-, Q^+\} = \{Q, \overline{Q}\}, c(Q^-) = \min\{c(Q), c(\overline{Q})\}.$$

This theorem now follows from the fact that for every $Z \in \mathcal{S}$, $c(T^-) \leq c(Z^+)$ as $c(T^-) \leq c'(T) \leq c'(Z) \leq c(Z^+)$.   ■

5.2.   Heuristics of Domination Number $\mathbf{\Omega((n-2)!)}$

While the assertion of the next theorem for odd n was already known to Rev Kirkman (see [104], p. 187), the even case result was only established by Tillson [792] as a solution to the corresponding conjecture by J.C. Bermond and V. Faber (who observed that the decomposition does not exist for $n = 4$ and $n = 6$).

Theorem 23 *For every $n \geq 2$, $n \neq 4$, $n \neq 6$, there exists a decomposition of $A(\overleftrightarrow{K}_n)$ into tours.*

Let $T(\overleftrightarrow{K}_n)$ $(\tau(n, c))$ be the total cost of all tours (the average cost of a tour) in $(\overleftrightarrow{K}_n, c)$. Since every arc of $\overleftrightarrow{K}_n$ is contained in $(n-2)!$ tours, $\tau(n, c) = T(\overleftrightarrow{K}_n)/(n-1)! = (n-2)!c(\overleftrightarrow{K}_n)/(n-1)!$, and hence, $\tau(n, c) = c(\overleftrightarrow{K}_n)/(n-1)$. This formula can also be shown using linearity of expectation. For the STSP, it is easy to see that $\tau(n, c) = 2c(K_n)/(n-1)$, where as above $\tau(n, c)$ is the average cost of a tour.

The following result was first obtained by Sarvanov [739] when n is odd, and Gutin and Yeo [423] when n is even. As we see below Theorem 24 allows us to show that certain heuristics are of domination number at least $(n-2)!$.

Theorem 24 *Consider any instance of the ATSP and a tour H such that $c(H) \leq \tau(n, c)$. If $n \neq 6$, then H is not worse than at least $(n-2)!$ tours.*

Proof: The result is trivial for $n = 2, 3$. If $n = 4$, the result follows from the simple fact that the most expensive tour T in $\overleftrightarrow{K}_n$ has cost $c(T) \geq c(H)$.

Assume that $n \geq 5$ and $n \neq 6$. Let $D_1 = \{C_1, \ldots, C_{n-1}\}$ be a decomposition of the arcs of $\overleftrightarrow{K}_n$ into tours (such a decomposition exists by Theorem 23). Given a tour R in $\overleftrightarrow{K}_n$, clearly there is an automorphism of $\overleftrightarrow{K}_n$ that maps C_1 into R. Therefore, if we consider D_1 together with the decompositions $(D_1, \ldots, D_{(n-1)!})$ of $\overleftrightarrow{K}_n$ obtained from D_1 using all automorphisms of $\overleftrightarrow{K}_n$ which map the vertex 1 into itself, we will have every tour of $\overleftrightarrow{K}_n$ in one of D_i's. Moreover, every tour is in exactly $n - 1$ decompositions D_i's (by mapping a tour C_i into a tour C_j $1 \leq i \neq j \leq n - 1$) we fix the automorphism).

Choose the most expensive tour in each of D_i and form a set $\mathcal{E}$ from all distinct tours obtained in this manner. Clearly, $|\mathcal{E}| \geq (n - 2)!$. As $\sum_{i=1}^{n-1} c(C_i) = c(\overleftrightarrow{K}_n)$, every tour T of $\mathcal{E}$ has cost $c(T) \geq \tau(n, c)$. Therefore, $c(H) \leq c(T)$ for every $T \in \mathcal{E}$. ∎

To see that the assertion of Theorem 24 is almost best possible, choose a tour H and an arc a not in H. Let every arc in H be of cost one, let $c(a) = n(n - 1)$ and let every arc not in $A(H) \cup \{a\}$ be of cost zero. Clearly the cost of H is less than the average (which is $n^2/(n - 1)$), but only tours using the arc a have higher cost. Thus, H is not worse than exactly $(n - 2)! + 1$ tours (including itself).

The first remark in Subsection 5.1 and Theorem 24 imply that, for the STSP, the assertion similar to Theorem 24 holds with $(n - 2)!$ replaced by $(n - 2)!/2$. However, Rublineckii [733] proved the following stronger result.

Theorem 25 *Consider an instance (K_n, c) of the STSP and a tour H such that $c(H) \leq \tau(n, c)$. Then H is not worse than at least $(n - 2)!$ tours when n is odd and $(n - 2)!/2$ tours when n is even.*

The ideas in the proof of Theorem 25 are similar to those used in the proof of Theorem 24. Instead of Theorem 23, Rublineckii [733] used a much simpler result that the edges K_n ($2K_n$) can be decomposed in edge-disjoint tours when n is odd (even), where $2K_n$ is the complete multigraph with 2 edges between every pair of distinct vertices.

The vertex insertion algorithm for the ATSP work as follows. First, we fix some ordering $v_1, \ldots, v_n$ of the vertices of $\overleftrightarrow{K}_n$. Then, we perform

$n - 1$ steps. On the first step we form the cycle $v_1v_2v_1$. On step k, $2 \leq k \leq n-1$, given the k-cycle $v_{\pi(1)}v_{\pi(2)} \ldots v_{\pi(k)}v_{\pi(1)}$ from the previous step, we find the value j_0 of j, which minimizes the expression

$$c(v_{\pi(j)}, v_{k+1}) + c(v_{k+1}, v_{\pi(j+1)}) - c(v_{\pi(j)}, v_{\pi(j+1)}),$$

$1 \leq j \leq k$, and insert v_{k+1} between $v_{\pi(j_0)}$ and $v_{\pi(j_0+1)}$ forming a $(k+1)$-cycle. Clearly, the vertex insertion algorithm for the STSP differs from the ATSP one in the fact that it starts from a cycle with three vertices. The following theorem was first proved by E.M. Lifshitz (see [733]) for the STSP.

Theorem 26 *Let H_n be a tour constructed by the vertex insertion algorithm $\mathcal{A}$ for the TSP with n vertices. Then $c(H_n) \leq \tau(n, c)$.*

Proof: We prove this result only for the ATSP by induction on n. The theorem is trivially true for $n = 2$. Let $H_{n-1} = v_{\pi(1)}v_{\pi(2)} \ldots v_{\pi(n-1)}v_{\pi(1)}$ be the cycle constructed in Step $n - 2$ of the algorithm and assume that in Step $n - 1$, it was decided to insert v_n between $v_{\pi(j_0)}$ and $v_{\pi(j_0+1)}$ in order to obtain H_n. Let V be the vertex set of $\overleftrightarrow{K}_n$ and, for a partition $X \cup Y = V$, let $(X, Y) = \{(x, y) : x \in X, y \in Y\}$. Then, we have

$$c(H_n) =$$

$$c(H_{n-1}) + c(v_{\pi(j_0)}, v_n) + c(v_n, v_{\pi(j_0+1)}) - c(v_{\pi(j_0)}, v_{\pi(j_0+1)}) \leq$$

$$c(H_{n-1}) + \frac{\sum_{i=1}^{n-1} c(v_{\pi(i)}, v_n) + c(v_n, v_{\pi(i+1)}) - c(v_{\pi(i)}, v_{\pi(i+1)})}{n - 1} =$$

$$c(H_{n-1}) + \frac{c(V - v_n, v_n) + c(v_n, V - v_n) - c(H_{n-1})}{n - 1} \leq$$

$$\frac{(n - 2)\tau(n - 1, c) + c(v_n, V - v_n) + c(V - v_n, v_n)}{n - 1} =$$

$$\frac{c(\overleftrightarrow{K}_n - v_n) + c(v_n, V - v_n) + c(V - v_n, v_n)}{n - 1} = \tau(n, c),$$

where $\tau(n - 1, c)$ is the average cost of a tour in $\overleftrightarrow{K}_n - v_n$. ∎

Theorems 24 and 26 imply the following result (similar result holds for the STSP, see Theorem 25).

Theorem 27 *[683] For the ATSP vertex insertion algorithm $\mathcal{A}$ and $n \neq$ 6 we have* $\text{domn}(\mathcal{A}, n) \geq (n-2)!$.

Gutin and Yeo [423] proved that the following ATSP algorithm always produces a tour of cost at most the average cost: choose an arc e such that the average cost of a tour through e is minimum, contract e and repeat the above choice and contraction until only two arcs remain. The output is the tour obtained from the two arcs together with the contracted ones. A similar algorithm was described by Vizing [810].

Given neighborhood structure N, the *best improvement* local search (LS) algorithm starts from an arbitrary tour; at every iteration it finds the best tour T' in the neighborhood $N(T)$ of the current tour T and replaces T by T'. The algorithm stops when $c(T') = c(T)$, in which case T is a *local optimum* with respect to N. Normally practical LS codes do not use the best improvement strategy; instead they find a better (than T) tour T' at every iteration as long as it is possible. This strategy saves running time and often yields better practical results, but the *first improvement* LS is difficult to formalize since the way to find the first improvement varies from code to code. Thus, let us restrict ourselves to the best improvement versions of 2-Opt and 3-Opt.

The k-Opt, $k \geq 2$, neighborhood of a tour T consists of all tours that can be obtained by deleting a collection of k edges (arcs) and adding another collection of k edges (arcs). Rublineckii [733] showed that every local optimum for 2-Opt and 3-Opt for the STSP is of cost at least the average cost of a tour and, thus, by Theorem 25 is of domination number at least $(n-2)!/2$ when n is even and $(n-2)!$ when n is odd. Observe that this result is of restricted interest since, to reach a k-Opt local optimum, one may need exponential time (see Section 3 in [463]). However, Punnen, Margot and Kabadi [684] managed to prove the following result.

Theorem 28 *For the STSP the best improvement 2-Opt algorithm produces a tour of cost at most $\tau(n, c)$ in at most*

$$O(\min\{n^3 \; log n, n \log(c(H_0) - \tau(c, n))\})$$

iterations, where H_0 is the initial tour.

Punnen, Margot and Kabadi observed that Theorem 28 holds also for 3-Opt and the pyramidal Carlier-Villon neighborhood. The last result can be extended to the ATSP because of Theorem 22. It is pointed

out in [684] that analogous results hold also for the well-known Lin-Kernighan algorithm [563] and shortest path ejection chain algorithm of Glover [372, 682] (see also Chapter 8).

5.3.   Bounds on Maximum Domination Number of Polynomial Heuristics

Clearly, unless P=NP, there is no polynomial time ATSP algorithm with domination number $(n-1)!$. Punnen, Margot and Kabadi [684] proved that unless P=NP, there is no polynomial time ATSP algorithm with domination number at least $(n-1)! - k$ for any constant k. This result can be extended from constant k to some function in n [684].

Gutin and Yeo [420] showed that, if there is a constant $r > 1$ such that for every sufficiently large k a k-regular digraph of order at most $rk - 1$ can be decomposed into Hamiltonian cycles in polynomial time in n, then the maximum domination number of the ATSP is $\Theta((n-1)!)$. This result is of interest due to the fact that Häggkvist [428, 429] announced (not published) that the above Hamiltonian decomposition exists for every $1 < r \leq 2$, see also Alspach et al. [19]. His approach is constructive and implies a polynomial algorithm to find such a decomposition. If Häggkvist's result holds, the main theorem in [420] implies that, in polynomial time, one can always find a tour, which is not worse than 50% of all tours.

Notice that the 50% threshold may seem to be easily achievable at first glance: just find the best in a large sample $\mathcal{S}$ of randomly chosen tours. A random tour has approximately a 50% chance of being better than 50% of all tours. However, in this approach the probability that the best tour of $\mathcal{S}$ is more expensive than 50% of all tours is always positive (if we consider only polynomial size samples of random tours). The difficulty of the problem by Glover and Punnen is well illustrated by the problem [605] to find a tournament on n vertices with the number of Hamiltonian cycles exceeding the average number of Hamiltonian cycles in a tournament of order n. This problem formulated long time ago has not been solved yet.

5.4.   Heuristics with Small Domination Numbers

Chapters 9 and 10 describe experimental results indicating that the greedy algorithm performs rather badly in the computational practice of the ATSP and STSP, see also [200, 377, 425, 463]. The aim of this subsection is to show that greedy-type algorithms are no match, with respect to the domination number, to heuristics considered in Subsection 5.2. This provides some theoretical explanation why "being greedy" is

not so good for the TSP. This subsection is based on Gutin, Yeo and Zverovich [424].

Before considering greedy-type algorithms in detail, we would like to notice that Punnen, Margot and Kabadi [684] recently constructed STSP instances for which the well-know double tree heuristic produces the unique worst tour. Note that these instances even satisfy the triangle inequality, i.e., for them the double tree heuristic computes a tour which is at most only twice more expensive than the cheapest tour. The authors of [684] also showed that the famous Christofides heuristic is of domination number at most $\lceil n/2 \rceil!$.

The *greedy algorithm* (GR) builds a tour in $(\overleftrightarrow{K}_n, c)$ by repeatedly choosing the cheapest eligible arc until the chosen arcs form a tour; an arc $a = uv$ is *eligible* if the out-degree of u in D and the in-degree of v in D equal zero, where D is the digraph induced by the set S of chosen arcs, and a can be added to S without creating a non-Hamiltonian cycle. The *nearest neighbor algorithm (NN)* starts its tour from a fixed vertex i_1, goes to the nearest vertex i_2 (i.e., $c(i_1, i_2) = \min\{c(i_1, j) : j \neq i_1\}$), then to the nearest vertex i_3 (from i_2) distinct from i_1 and i_2, etc. Computational experience with NN for the ATSP and STSP is discussed in Chapters 9 and 10, and [200, 463]. We will also consider a stronger version of NN, the *repetitive NN algorithm (RNN)*, which starts NN from each of the vertices in turn and chooses the best tour. In the rest of the chapter we assume that NN starts from vertex 1.

The following theorem was first proved in [424]. We give a different proof by adapting the proof of a much more general result from [419]. The result holds for a wide family of CO problems including the assignment problem, i.e., the domination number of the greedy algorithm for the assignment problem is proved to be 1.

Theorem 29 *The domination number of GR for the TSP is 1.*

Proof: This proof holds for both ATSP and STSP, but for simplicity we assume that we deal with the STSP. We will consider tours of STSP as sets of their edges. For a set $S = \{e_1, \ldots, e_s\}$ of edges forming a partial tour in K_n (i.e., this set of edges can be extended to a tour), $Z(e_1, \ldots, e_s)$ denotes the set of edges not in S such that each edge from $Z(e_1, \ldots, e_s)$ can be added to S to form a (larger) partial tour.

Let $T' = \{e_1', e_2', \ldots, e_n'\}$ be an arbitrary fixed tour and let T be an arbitrary tour distinct from T'. It is easy to see that

$$\sum_{j=0}^{n-1} |Z(e_1', e_2', \ldots, e_j') \cap T| < n(n+1)/2. \tag{1}$$

Let $M > n$, let $c(e'_i) = iM$ for each $e'_i \in T'$ and, for $e \notin T'$, let $c(e) = 1 + jM$ if $e \in Z(e'_1, e'_2, \ldots, e'_{j-1})$ but $e \notin Z(e'_1, e'_2, \ldots, e'_j)$. Clearly, GR constructs T' and $c(T') = Mn(n+1)/2$.

Let $T = \{e_1, e_2, \ldots, e_k\}$. Assume that $c(e_i) \in \{aM, aM + 1\}$. Then clearly

$$e_i \in Z(e'_1, e'_2, \ldots, e'_{a-1}),$$

but $e_i \notin Z(e'_1, e'_2, \ldots, e'_a)$, so e_i lies in $Z(e'_1, e'_2, \ldots, e'_j\}) \cap T$, provided $j \leq a - 1$. Thus, e_i is counted a times in the sum in (1). Hence,

$$
\begin{aligned}
c(T) &= \sum_{i=1}^{n} c(e_i) \leq n + M \sum_{j=0}^{n-1} |Z(\{e'_1, e'_2, \ldots, e'_j\}) \cap T| \\
&\leq n + M(n(n+1)/2 - 1) = n - M + c(T'),
\end{aligned}
$$

which is less than the cost of T' as $M > n$. Since GR finds T', and T is arbitrary, we see that GR finds the unique most expensive tour.   ∎

The proof of Theorem 29 implies that the domination number of NN for TSP is also 1 (indeed, NN will construct the same tour as GR). However, the following two theorems show that the situation is slightly better for RNN.

Theorem 30 *[424] Let $n \geq 4$. The domination number of RNN for the ATSP is at least $n/2$ and at most $n - 1$.*

Proof: We first consider the following instance of the ATSP, which proves that RNN for the ATSP has domination number at most $n - 1$. Let $N > 2n$. Let all arcs $(i, i + 1)$, $1 \leq i < n$, have cost iN, all arcs $(i, i + 2)$, $1 \leq i \leq n - 2$, cost $iN + 1$, and all remaining forward arcs (i, j) cost $iN + 2$. Let a backward arc (i, j) have cost $(j - 1)N$.

When NN tour T starts at $i \notin \{1, n\}$, it has the form $(i, 1, 2, \ldots, i - 1, i + 1, i + 2, \ldots, n, i)$ and cost

$$\ell = \sum_{k=1}^{n-1} kN - N + 1.$$

When T starts at 1 or n, we simply have $T = (1, 2, \ldots, n, 1)$ of cost $\sum_{k=1}^{n-1} kN > \ell$. Let $\mathcal{F}$ denote the set of all tours T described above (note that $|\mathcal{F}| = n - 1$). Observe that any tour in $\mathcal{F}$ has cost at least ℓ. Let C be any tour not in $\mathcal{F}$. Let B denote the set of backward arcs in C, and define the length of a backward arc (i, j) by $i - j$. Let q denote the sum of the lengths of the arcs in B. Since C is a tour (and therefore there is a path from n to 1) we have $q \geq n - 1$. The cost of C is at

most $\sum_{i=1}^{n}(iN+2) - qN - |B|N$, since if (i,j) is an arc in B, then the corresponding term $iN+2$ in the sum can be replaced by the real cost $(j-1)N = iN + 2 - (i-j+1)N - 2$ of the arc. We have

$$\sum_{i=1}^{n}(iN+2) - qN - |B|N \leq \ell + 2n + N(n+1-q-|B|) - 1.$$

Since C is not in $\mathcal{F}$ we have $|B| \geq 2$, implying that $2n + N(n+1-q-|B|) - 1$ is negative except for the case of $q = n-1$ and $|B| = 2$. We may conclude that the cost of C is less than ℓ, as $q = n-1$ and $|B| = 2$ would imply that C belongs to $\mathcal{F}$. Therefore all cycles not in $\mathcal{F}$ have cost less than those in $\mathcal{F}$.

In order to prove that RNN has domination number at least $n/2$, assume that this is false, and proceed as follows. RNN constructs n tours, but several of them may coincide. By the assumption, there exist at least three tours that coincide. Let $F = x_1 x_2 \ldots x_n x_1$ be a tour such that $F = F_i = F_j = F_k$, where F_s is the tour obtained by starting NN at x_s and x_i, x_j and x_k are distinct. Without loss of generality, we may assume that $i = 1$ and $2 < j \leq 1 + (n/2)$. For every m, with $j < m \leq n$, let C_m be the tour obtained by deleting the arcs $(x_i, x_{i+1}), (x_j, x_{j+1}), (x_m, x_{m+1})$ and adding the arcs

$$(x_i, x_{j+1}), (x_m, x_{i+1}), (x_j, x_{m+1}).$$

Note that $c(C_m) \geq c(F)$, since $c(x_i, x_{i+1}) \leq c(x_i, x_{j+1})$ (because we used NN from x_i to construct F_i), $c(x_j, x_{j+1}) \leq c(x_j, x_{m+1})$ (since we used NN from x_j to construct F_j) and $c(x_m, x_{m+1}) \leq c(x_m, x_{i+1})$ (since NN chose the arc $x_m x_{m+1}$ on F_j, when the arc $x_m x_{i+1}$ was available). Therefore the cost of F is at most that of $F, C_{j+1}, C_{j+2}, \ldots, C_n$, implying that the domination number is at least $n - j + 1 \geq n/2$, a contradiction. ∎

We call a tour $x_1 x_2 \ldots x_n x_1$, $x_1 = 1$, of the STSP *pyramidal* if $x_1 < x_2 < \ldots < x_k > x_{k+1} > \ldots > x_n$ for some index k. Since every pyramidal tour $x_1 x_2 \ldots x_n x_1$, $x_1 = 1$, is determined by the set $\{x_2, x_3, \ldots, x_{k-1}\}$ or the set $\{x_{k+1}, x_{k+2}, \ldots, x_n\}$ (clearly, $x_k = n$), we obtain that the number of pyramidal tours of the STSP is 2^{n-3}.

The next theorem gives an upper bound for the domination number of RNN for the STSP. Even though the theorem leaves a possibility that this domination number is exponential, it is still much smaller than $\Theta((n-2)!)$.

Theorem 31 *[424] Let $n \geq 4$. The domination number of RNN for the STSP is at most 2^{n-3}.*

Proof: We consider the following instance of the STSP, which proves that RNN for the STSP has domination number at most 2^{n-3}. Let $N > 2n$. Let all edges $(i, i+1)$, $1 \le i < n$, have cost iN, all edges $(i, i+2)$, $1 \le i \le n-2$, cost $iN+1$, and all remaining edges (i, j), $i < j$, cost $iN+2$.

Let c_{RNN} be the cost of the cheapest tour constructed by RNN. It is straightforward to verify that

$$c_{\mathrm{RNN}} = c(12\ldots n1) = \sum_{i=1}^{n-1} iN + N + 2. \tag{2}$$

Let $T = x_1 x_2 \ldots x_n x_1$ be a tour in K_n, $x_1 = 1$; we orient all edges of T such that T becomes a directed cycle T'. Some of arcs in T' are forward, others are backward. For a backward arc $e = (j, i)$, we define its length as $q(e) = j - i$. We denote the sum of the lengths of backward arcs in T' by $q(T')$. (By the definition of a backward arc the length of every backward arc is positive.) Let $c_{\max}$ be the cost of the most expensive non-pyramidal tour T. Since the number of pyramidal tours is 2^{n-3}, to prove this theorem it suffices to show that $c_{\max} < c_{\mathrm{RNN}}$.

Observe that $q(T') \ge n$ for every T' corresponding to a non-pyramidal tour T. Let H be a non-pyramidal tour of cost $c_{\max}$, and let $e_i = (i, j)$ be an arc of H'. If e_i is forward, then $c(e_i) \le iN+2$, and if e_i is backward, then $c(e_i) \le jN + 2 = iN + 2 - q(e_i)N$. Thus,

$$c_{\max} \le \sum_{i=1}^{n}(iN+2) - q(H')N \le \sum_{i=1}^{n-1} iN + 2n$$

as $q(H') \ge n$. Since $N > 2n$ and by (2), we conclude that indeed $c_{\max} < c_{\mathrm{RNN}}$. ∎

By the observation in the first paragraph of Subsection 5.1 and the lower bound in Theorem 30, the domination number of RNN for the STSP is at least $n/4$. It would be interesting to find the exact values of the domination number of RNN for the ATSP and STSP.

6.   Further Research

Exponential neighborhoods can be included into a quite general approach in combinatorial optimization (CO): restrict the feasible set of solutions of a CO problem such that one can find the best solution of the restricted problem in polynomial time. This method, which we suggest to call the *polynomial restriction approach* (PRA) is somewhat dual to the analysis of polynomial solvable cases of the TSP: while in the latter

one restricts instances to consider, in the PRA we restrict the solution set for all instances of the TSP. There is some interaction between the two approaches, see e.g. Glover and Punnen [382], but in essence they are quite different. Notice that PRA may be of interest not only for exponential neighborhoods; non-neighborhood type sets of exponential size, where the best tour can be computed in polynomial time, may be used in exact algorithms (see below) or in certain meta-heuristics.

The following approach is obviously hardly practical, but perhaps its modifications may be of interest to practical exact algorithms. All tours of the ATSP can be enumerated and represented as leaves of a special rooted tree $\mathcal{T}$ as follows. The root of $\mathcal{T}$ (e.g. the first level of $\mathcal{T}$) is the vertex 1. Every node of the tth level of $\mathcal{T}$ corresponds to a path $i_1 i_2 ... i_t$ in $\overleftrightarrow{K}_n$ such that $i_1 = 1$, and every edge of $\mathcal{T}$ is of the form $\{i_1 i_2 ... i_{t-1}, i_1 i_2 ... i_t\}$ and has weight $c(i_{t-1} i_t)$ (except for $t = n$ when the weight is $c(i_{n-1} i_n) + c(i_n i_1)$). It is clear how to develop a simple branch-and-bound algorithm using $\mathcal{T}$: search $\mathcal{T}$ by the means of the depth first search. The well-known Held-Karp dynamic programming algorithm [443] solves the ATSP to optimality in time $O(n^2 2^n)$. A simple modification of this algorithm can be used to find a cheapest Hamiltonian path between a pair of given vertices in $\overleftrightarrow{K}_{\log n}$ in time $O(n \log^2 n)$. This modification can be applied to cut $\log n$ last levels of $\mathcal{T}$, i.e., visit in the worst case less than $n^{\log n}(n - \log n)!$ leaves instead of $(n-1)!$ leaves.

Our study of exponential neighborhoods for the ATSP suggests the following natural question.

Problem 32 *Do there exist polynomially searchable neighborhoods of size more than $\Theta(e^{\sqrt{n/2}} \lfloor \frac{n+1}{2} \rfloor! n^k)$ for any positive integer k?*

The following question is stronger in a sense; it was raised by Deineko and Woeginger [248], who conjectured that the answer to Problem 33 is *yes* (under the assumption that P≠NP).

Problem 33 *Do there exist polynomially searchable neighborhoods of size at least $\lfloor \alpha(n-1) \rfloor!$ for some fixed $\alpha > \frac{1}{2}$?*

While one can see certain progress in the theoretical study of exponential neighborhoods, their use in computational algorithms has been less successful so far. We hope that this chapter will motivate extensive experimental study of various exponential neighborhoods.

The following problem, which we raised earlier, is one of the central questions in domination analysis for the TSP.

Problem 34 *Determine the maximum domination number of a polynomial heuristic for the ATSP (STSP).*

We provided exact values and bounds for the domination number of various heuristics for the ATSP and STSP. Observe that, for a given TSP instance $\mathcal{I}$, most of TSP heuristics will retain domination number if we increase the cost of every arc (edge) in $\mathcal{I}$ by the same positive constant M. This implies that such heuristics will have the same domination number even if we restrict the set of instances from all ATSP (STSP) instances to those for which the triangle inequality holds.

The Euclidean TSP is a special class of STSP instances with triangle inequality and it is of great importance to practice. It would be quite interesting to obtain domination number results for Euclidean TSP heuristics (where the set of instances is restricted to the Euclidean TSP ones).

The domination number reflects the worst case behavior of a heuristic. If the worst case instances of the TSP are rather untypical for some heuristic, the domination number may not indicate the true value of the heuristic. Perhaps, certain probabilistic parameters, such as the average domination number, may provide further indication of the quality of the heuristic.