Exponential neighbourhood local search for the traveling salesman problem

Gregory Gutin*
Department of Mathematics and Statistics
Brunel University of West London
Uxbridge, Middlesex, UB8 3PH, U.K.
z.g.gutin@brunel.ac.uk

Abstract

Scope and Purpose – While exact algorithms can only be used for solving small or quite moderate instances of the traveling salesman problem (TSP), local search remains the main practical tool for finding near optimal solutions for large scale instances. Exponential neighbourhood local search (ENLS) is a relatively new direction in local search for the TSP. In ENLS, one finds the best among very large, exponential, number of tours. Computational experiments reported by several researches demonstrate a very high potential of ENLS. In the present paper, we analyze theoretical properties of some exponential neighbourhoods.

Abstract – We analyze an approach to the TSP, introduced by Punnen (1996), which is a generalization of approaches by Sarvanov and Doroshko (1981) and Gutin (1984). We show that Punnen's approach allows one to find the best among $\Theta(exp(\sqrt{n/2})(n/2)!/n^{1/4})$ tours in the TSP with n cities (n is even) in $O(n^3)$ time. We describe an $O(n^{1+\beta})$-time algorithm (for any $\beta \in (0,2]$) that constructs the best among $2^{\Theta(n \log n)}$ tours. This algorithm provides low complexity solutions to a problem by Burkard, Deineko and Woeginger (1996) and may be quite useful for large scale instances of the TSP. We also show that for every positive integer r there exists an $O(r^5 n)$-time algorithm that finds the best among $\Omega(r^n)$ tours. This improves a result of Balas and Simonetti (1996) who showed that the best among $\Omega(r^n)$ tours can be obtained in time $O(r^2 2^r n)$.

1 Introduction, terminology and notation

Let G be a weighted complete directed or undirected graph on n vertices (the weights are assigned to the edges). In the traveling salesman problem (TSP) we are seeking for a

*G. Gutin completed his Ph.D. under supervision of Professor N. Alon in School of Mathematics, Tel Aviv University. He has been a visiting member of Department of Mathematics and Computer Science, Odense University, Denmark. Now he is a lecturer at Department of Mathematics and Statistics, Brunel University, U.K. His research interests include combinatorial optimization and graph theory.

hamiltonian (directed or undirected) cycle (called a *tour*) of G of minimum weight. The TSP is *symmetric* (*asymmetric*, respectively) if G is undirected (directed, respectively). The TSP is one of the fundamental problems in combinatorial optimization (see the books [13] and [19] and the chapter [11]).

While exact algorithms can only be used for solving small or quite moderate instances of the TSP, local search remains the main practical tool for finding near optimal solutions for the large scale problems (see [11, 19]). Since the pioneering works by Croes [4] and Lin [14], a large number of papers has been devoted to the development of local search heuristics, mostly 2-Opt, 3-Opt, Lin-Kernigan and their modifications, which use neighbourhoods of small polynomial size (some of the heuristics including 2-Opt and Lin-Kernigan are only suitable for the symmetric TSP). Now when it seems that the classical approach dealing with polynomial size neighbourhoods has come close to its limits [11], an alternative, exponential neighbourhoods local search (ENLS), has begun to emerge (see e.g. [1, 2, 3, 7, 8, 15, 16]). A possibility to obtain, in polynomial time, the best among a very large number of tours seems quite attractive and useful. Indeed, computational experiments in [1, 3, 17, 18] have already demonstrated a very high potential of ENLS. An important link between polynomial time solvable cases of the TSP and ENLS has also been discovered [1, 2, 3, 8]. We believe that ENLS will open new prospects for other difficult combinatorial problems including the maximum independent set problem where good results with respect to 2-Opt have already been noticed [12].

As Burkard, Deineko and Woeginger [2] and Punnen [16] give quite long lists of exponential neighbourhoods and polynomial algorithms for their complete search, we restrict ourselves to remarks on the algorithms with record complexity or record size of the neighbourhoods. Linear time algorithms exploring exponential neighbourhoods are obtained by Balas and Simonetti [1], Carlier and Villon [3], and Glover and Punnen [8]. Only algorithms by Balas and Simonetti [1] can find the best among $\Omega(r^n)$ tours in linear time (i.e. $O(r^2 2^r n)$ time) for any positive integer r. $O(n^3)$-time algorithms obtaining the best among $\Theta((\frac{n}{2})!)$ tours were independently found by Sarvanov and Doroshko [20], Gutin [9], and Punnen [16]. We note that the algorithms in [9, 16, 20] explore neighbourhoods with logarithm of their sizes equal $\Theta(n \log n)$, and thus provide a solution to the following problem by Burkard, Deineko and Woeginger [2]: does there exist a polynomial time algorithm for finding the best among $2^{\Theta(n \log n)}$ tours?

The main idea behind the algorithms in [9, 16, 20] can be described as follows. Partition the cities of the TSP into two groups $x_1, x_2, ..., x_m$ and $y_1, y_2, ..., y_k$ ($k+m = n$ and $m \leq k$) and form the subtour (cycle) $C = y_1 y_2 ... y_k y_1$. An insertion of the cities $x_1, x_2, ..., x_m$ between the cities in C, such that for every j at most one x_i can be inserted between y_j and y_{j+1}, gives a tour of a special form. All tours of this form are called the neighbourhood $N = N(y_1 ... y_k; x_1 ... x_m)$. Using algorithms for the assignment problem, we can obtain a tour of minimum weight among the tours in N in time $O(n^3)$. In [20] only the neighbourhood N with $m = n/2$ (n is even) was considered; in [9] the two cases $m = n/2$ (n is even) and $m = (n-1)/2$ (n is odd) were treated; and the above general approach was very recently introduced by Punnen [16]. (However, Punnen [16] has not attempted to obtain a value of m that provides the maximum to the size of the neighbourhood.) In [10] some probabilistic analysis of the cases $m = n/2$ (n is even) and $m = (n-1)/2$ (n is odd) was carried out. Note that the above approach is suitable for both symmetric and asymmetric TSP's as a part of metaheuristics.

In Section 2 of this paper, we demonstrate that Punnen's approach leads us to considerably larger neighbourhoods for certain values of m. In fact, we obtain the optimum, in this sense, value of m. In Section 3 we give some applications of the above approach. Punnen [16] describes an $O(n^4)$-algorithm for finding the best among $\Theta(n(n/2)!)$ tours (n is even). We show that the best among $\Theta(n^s exp(\sqrt{n/2})(n/2)!/n^{\frac{1}{4}})$ tours can be found in time $O(n^{3+s})$ for every non-negative integer s. As the complexity $O(n^3)$ is too high for large scale instances of the TSP, we suggest a partition of the cities into groups that results, in particular, in near-linear time algorithms which explore neighbourhoods of size still equal $2^{\Theta(n \log n)}$ (they are lower complexity solutions for the problem by Burkard, Deineko and Woeginger) and an $O(r^5 n)$-time algorithm which searches neighbourhoods of size $\Omega(r^n)$, given any positive integer r (algorithms by Balas and Simonetti [1] require $O(r^2 2^r n)$ time for that purpose).

As we already mentioned, the above approach can be used for both symmetric and asymmetric TSP's, yet, because of obvious similarity, we will discuss only the symmetric case in this paper.

In the rest of this paper, G stands for a complete graph on n vertices (= cities); $d(x, y)$ is the weight of an edge xy of G; S_k is the set of all permutations on $\{1, ..., k\}$; for a real r, $[r]_0$ ($[r]_1$, resp.) is the maximum integer (semi-integer, resp.) that does not exceed r (a semi-integer is a number of the form $p/2$, where p is an odd integer); sometimes, we use $[r]$ instead of $[r]_0$; for an integer m, $\sigma(m) = m \; mod \; 2$.

It is well known [6] that

$$\sqrt{2\pi n}(n/e)^n e^{(12n+1)^{-1}} < n! < \sqrt{2\pi n}(n/e)^n e^{(12n)^{-1}},$$

We will use the following consequence of this inequality

$$\sqrt{2\pi n}(n/e)^n < n! < \sqrt{4\pi n}(n/e)^n. \qquad (1)$$

2 Neighbourhoods $N(y_1 ... y_k; x_1 ... x_m)$

Suppose that the vertices of G are partitioned into two groups $x_1, x_2, ..., x_m$ and $y_1, y_2, ..., y_k$ ($k + m = n$ and $m \leq k$). For simplicity of the exposition, add to the first group $k - m$

fictitious vertices $x_{m+1}, ..., x_k$ and consider the neighbourhood

$$N(y_1...y_k; x_1...x_m) = \{y_1 x_{\tau(1)} y_2 x_{\tau(2)} y_3 ... y_{k-1} x_{\tau(k-1)} y_k x_{\tau(k)} y_1 : \ \tau \in S_k\} \qquad (2)$$

of, say, tour $y_1 x_1 y_2 x_2 ... y_m x_m y_{m+1} y_{m+2} ... y_k y_1$. For $\tau(j) > m$, we ignore the presence of $x_{\tau(j)}$ by assuming that $d(y_j, x_{\tau(j)}) + d(x_{\tau(j)}, y_{j+1}) = d(y_j, y_{j+1})$. To find an optimal tour in (2), we construct a weighted complete bipartite graph B with partite sets $X = \{x_1, , ..., x_k\}$ and $Z = \{z_1, ..., z_k\}$. The weight of an edge $x_i z_j$ is calculated as follows: if $i \le m$, then $w(x_i, z_j) = d(y_j, x_i) + d(x_i, y_{j+1})$, where $y_{k+1} = y_1$; otherwise (i.e. $i > m$), $w(x_i, z_j) = d(y_j, y_{j+1})$. It follows from the above definitions that a tour

$$y_1 x_{\tau(1)} y_2 x_{\tau(2)} y_3 ... y_{k-1} x_{\tau(k-1)} y_k x_{\tau(k)} y_1$$

in (2) corresponds to the perfect matching $x_{\tau(i)} z_i$, $i = 1, 2, ..., k$ (where $\tau \in S_k$), in B and the weight of the tour is equal to the weight of the matching. Thus, to find an optimal tour in (2) it is suffices to construct a minimum weight perfect matching in B (the assignment problem whose complexity is $O(k^3) = O(n^3)$).

We remind that a tour of (2) can be viewed as the result of an insertion of the cities $x_1, ..., x_m$ into the subtour $C = y_1 y_2 ... y_k y_1$ such that for every j at most one x_i is inserted between y_j and y_{j+1} ($y_{k+1} = y_1$). Let $ins(n, m)$ be the number of tours in (2) and let $n \ge 5$. As there are $k = n - m$ 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 $ins(n, m) = (n - m)(n - m - 1)...(n - 2m + 1)$. It is natural to find a value of m ($m \le n/2$) that provides the maximum for $ins(n, m)$ for a *fixed* n. Let $maxins(n) = \max\{ins(n, m) : \ 1 \le m \le n/2\}$.

Assume first that n is even. Consider $f(p) = ins(n, n/2 - p)$, where p is a non-negative integer smaller than $n/2$. For $p \ge 1$, the difference $\Delta f(p) = f(p) - f(p-1) = b(-2p(2p-1) + (n/2 + p)) = bq(p)/2$, where $q(p) = -8p^2 + 6p + n$, $b = (n/2 + p - 1)(n/2 + p - 2)...(2p + 1)$. 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 $p \ge 1$, $q(p)$ decreases and has a positive root $r = \sqrt{\frac{1}{8}(n + \frac{9}{8})} + \frac{3}{8}$. Thus, $f(p)$ is maximum for either $p = [r]$ or $p = [r] + 1$. Now, following C. Schulze, we show that $f([r]) > f([r] + 1)$. Let $h(p) = 8p^2 + 10p + 2$; $h(p)$ increases when $p \ge 0$. Clearly, $h(r - 1) = n$. As $[r] > r - 1$, we obtain that $h([r]) > h(r - 1)$. Thus, $h([r]) > n$ and $(2[r] + 2)(2[r] + 1) > n/2 + [r] + 1$. Therefore, $f([r]) > f([r] + 1)$.

Analogously, when n is odd, we obtain that $f(p)$ is maximum for $p = [r]_1$. Now we can state the following:

Theorem 2.1 *For a fixed $n \ge 5$, the maximum size of the neighbourhood (2) equals*

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

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

Below we obtain an asymptotic formula for $maxins(n)$. Note that, for $n \leq 2m+1$, $ins(n,m) = [\frac{n+1}{2}]!$.

Theorem 2.2 $maxins(n) = \Theta\left(\frac{e^{\sqrt{n/2}}[\frac{n+1}{2}]!}{n^{\frac{1}{4}+[\frac{1}{2}]_{\sigma(n)}}}\right)$.

Proof: Let $f(n) = \left([\frac{n+1}{2}]!\right)^{-1} \frac{(n/2+p_0)!}{(2p_0)!}$. Since $n/2 + [1/2]_{\sigma(n)} = [(n+1)/2]$,

$$f(n) = \frac{(n/2+p_0)(n/2+p_0-1)...(n/2+[3/2]_{\sigma(n)})}{(2p_0)!}.$$

It suffices to show that

$$f(n) = \Theta\left(\frac{e^{\sqrt{n/2}}}{n^{\frac{1}{4}+[\frac{1}{2}]_{\sigma(n)}}}\right).$$

By (1),

$$(2p_0)! = \Theta\left(\sqrt{p_0}(2p_0/e)^{2p_0}\right). \tag{3}$$

It is easy to verify that

$$\sqrt{n/8} - 1 < p_0 < \sqrt{n/8} + 1. \tag{4}$$

It is well known that

$$\left(1 + \frac{1}{n}\right)^n = \Theta(1) \tag{5}$$

By (4) and (5), we obtain that

$$\left(\frac{n/2+p_0}{n/2}\right)^{p_0} = \Theta(1).$$

This implies that

$$(n/2+p_0)(n/2+p_0-1)...(n/2+[3/2]_{\sigma(n)}) = \Theta\left((n/2)^{p_0-[1/2]_{\sigma(n)}}\right). \tag{6}$$

By (3) and (6),

$$f(n) = \Theta\left(\frac{(n/2)^{p_0-[1/2]_{\sigma(n)}}}{\sqrt{p_0}(2p_0/e)^{2p_0}}\right) = \Theta\left(\frac{\left(\frac{ne^2}{8p_0^2}\right)^{p_0}}{p_0^{1/2}n^{[1/2]_{\sigma(n)}}}\right). \tag{7}$$

It follows from (4) and (5) that

$$\left(\frac{n}{8p_0^2}\right)^{p_0} = \Theta(1). \tag{8}$$

By (7), (4) and (8),

$$f(n) = \Theta\left(\left(\frac{n}{8p_0^2}\right)^{p_0} \frac{e^{\sqrt{n/2}}}{n^{\frac{1}{4}+[\frac{1}{2}]_{\sigma(n)}}}\right) = \Theta\left(\frac{e^{\sqrt{n/2}}}{n^{\frac{1}{4}+[\frac{1}{2}]_{\sigma(n)}}}\right).$$

□

Consider also the following numerical example illustrating the significant difference between $maxins(n)$ and $ins(n, n/2)$ (n is even). Let $n = 2000$. Then, by Theorem 2.1 $p_0 = 16$ and

$$\frac{maxins(2000)}{1000!} = \frac{1016!}{1000!32!} > \left(\frac{1001}{16 \times 17}\right)^{16} > 3.6^{16}.$$

3 Applications

Punnen [16] describes an $O(n^4)$-algorithm for finding an optimal tour among $\Theta(n(n/2)!)$ ones. We can accomplish a more general task of constructing an $O(n^{3+s})$-algorithm ($s \geq 0$) for finding the best among $\Theta(n^s ins(n, m))$ tours: search $O(n^s)$ neighbourhoods (2) which are pairwise independent, i.e. have empty intersections. A simple way of creating independent neighbourhoods (2) is to fix the order of y_{k-1} and y_k and vary the order of the rest of the vertices y_j. Clearly, $N(y_{\pi(1)}...y_{\pi(k-2)}y_{k-1}y_k; x_1...x_m) \cap N(y_{\tau(1)}...y_{\tau(k-2)}y_{k-1}y_k; x_1...x_m) = \emptyset$, where π and τ are distinct permutations on $\{1, ..., k-2\}$. Note that this approach can easily be parallelized.

Now we turn our attention to low complexity algorithms. First, let us consider the following modification of the TSP. We wish to find a minimum weight hamiltonian path between v_1 and v_n in G. Punnen's approach can be easily modified to produce an ENLS algorithm for this problem. Partition the cities $v_1, v_2, ..., v_n$ into two groups $x_1, ..., x_m$ and $y_1, ..., y_k$ such that $y_1 = v_1$, $y_k = v_n$, $k + m = n$ and $m \leq k-1$. Form the path $P = y_1 y_2 ... y_k$ and insert $x_1, ..., x_m$ between the vertices in P such that for every $j = 1, 2, ..., k-1$ at most one x_i is inserted between y_j and y_{j+1}. We can consider the set of hamiltonian (v_1, v_n)-paths that can be obtained in this way as a neighbourhood $N'(y_1...y_k; x_1...x_m)$ analogous to (2). Add fictitious vertices $x_{m+1}, ..., x_{k-1}$. Then

$$N'(y_1...y_k; x_1...x_m) = \{y_1 x_{\tau(1)} y_2 x_{\tau(2)} y_3 ... y_{k-1} x_{\tau(k-1)} y_k : \ \tau \in S_{k-1}\} \tag{9}$$

Similarly to (2), for $\tau(j) > m$, we ignore the presence of $x_{\tau(j)}$ in (9) by assuming that $d(y_j, x_{\tau(j)}) + d(x_{\tau(j)}, y_{j+1}) = d(y_j, y_{j+1})$. The obvious analog B' of the complete bipartite graph B allows us to find the best among paths in (9) in time $O(n^3)$. Let $ins'(n, m)$ be the number of paths in (9). There is no need to investigate $ins'(n, m)$ since $ins'(n, m) = ins(n-1, m)$.

Now we can introduce a simple, yet, powerful approach for constructing low complexity ENLS algorithms. Let $b = b(n)$ be a function whose values are positive even integers.

Partition the cities $v_1, ..., v_n$ into blocks $B_1, B_2, ..., B_{q+1}$, where $|B_i| = b$ for $i = 1, ..., q$, $|B_{q+1}| = n \ mod \ b$, $q = [n/b]$. Let u_i, w_i be distinct cities in B_i, $i = 1, ..., q$. Using neighbourhoods of type (9), in time $O(b^3)$, we can find a path P_i which is the best among

$$h = maxins(b-1) \geq (b/2)! \geq \left(\frac{b}{2e}\right)^{\frac{b}{2}}$$

hamiltonian (u_i, w_i)-paths in the subgraph of G induced by B_i, $i = 1, ..., q$. Let P_{q+1} be a hamiltonian path in the subgraph of G induced by B_{q+1} (P_{q+1} is possibly empty). The tour $T = P_1 P_2 ... P_{q+1} u_1$ is the best among (at least)

$$t(n, b) = h^{[n/b]} \geq \left(\frac{b}{2e}\right)^{\frac{n-b}{2}}$$

tours.

First, choose a constant α, $0 < \alpha < 1$, and let $b = [n^\alpha]$ or $[n^\alpha] + 1$ (depending on which of the two is even). Then, $\log t(n, b) = \Theta(n \log n)$. We can find T in time $O(\frac{n}{b} b^3) = O(n^{1+2\alpha})$.

Now let r be a positive integer. Clearly, $(n-b)/2 \geq 2n/5$ for $n \geq 5b$. Thus, for $n \geq 5b$ and $b = b_0 + 1$ or $b_0 + 2$, where $b_0 = [2er^{5/2}]$,

$$t(n, b) \geq \left(\frac{b}{2e}\right)^{\frac{n-b}{2}} \geq r^n.$$

The tour T can be obtained in $O(b^2 n) = O(r^5 n)$ time.

We have proved the following:

Theorem 3.1 *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.*

Problem 3.2 *Does there exist a linear time algorithm for finding the best among $2^{\Theta(n \log n)}$ tours?*

Remark 3.3 *Very recently V. Deineko and G. Woeginger [5] showed the following upper bound to the size of exponential neighbourhood N_n for the TSP depending on the search time $t(n)$: $|N_n| \leq (2t(n)/n)^n$. The bound implies a negative answer to Problem 3.2 as well as the fact that the first part of Theorem 3.1 is optimal in a sense: time required to search a neighbourhood of size $2^{\Theta(n \log n)}$ is $\Omega(n^{1+\alpha})$ for some positive constant α.*

Acknowledgment. Most of the results in this paper were obtained when the author was visiting Dept of Maths and CS, Odense University, Denmark in January 1997. I would like to thank the staff of the department and especially Jørgen Bang-Jensen and Bjarne Toft for their hospitality. I would also like to thank Cord Schulze and Anders Yeo for useful and stimulating discussions of the paper. This work was partially supported by Brunel University grant BRIEF.

References

[1] E. Balas and N. Simonetti, Linear time dynamic programming algorithms for some new classes of restricted TSP's. *Proc. IPCO V*, LNCS **1084**, Springer Verlag, 316-329 (1996).

[2] R.E. Burkard, V.G. Deineko and G.J. Woeginger, The travelling salesman problem and the PQ-tree. *Proc. IPCO V*, LNCS **1084**, Springer Verlag, 490-504 (1996).

[3] J. Carlier and P. Villon, A new heuristic for the traveling salesman problem. RAIRO **24,** 245-253 (1990).

[4] G.A. Croes, A method for solving travelling-salesman problems. *Operations Res.* **6**, 791-812 (1958).

[5] V. Deineko and G.J. Woeginger, A study of exponential neighbourhoods for the traveling salesman problem and the quadratic assiggnment problem. *Technical Report Woe05*, Technical Univ. Graz, Austria, July (1997).

[6] W. Feller, *An Introduction to Probability Theory and Its Applications.* Wiley, New York (1968).

[7] F. Glover, Ejection chains, reference structures, and alternating path algorithms for traveling salesman problem. *Research Report.* Univ. Colorado-Boulder, Graduate School of Business (1992).

[8] F. Glover and A. Punnen, The traveling salesman problem: New solvable cases and linkages with the development of approximation algorithms. *J. Oper. Res. Soc.* **48** 502-510 (1997).

[9] G. Gutin, On an approach for solving the traveling salesman problem. In *Proc. The USSR Conference on System Res.* Moscow, 184–185 (1984) (in Russian).

[10] G. Gutin, On the efficiency of a local algorithm for solving the traveling salesman problem. *Automation and Remote Control. no. 11, part 2*, 1514-1519 (1988).

[11] D.S. Johnson and L.A. McGeoch, The traveling salesman problem: a case study in local optimization. *Local Search in Combinatorial Optimization,* E.H.L. Aarts and J.K. Lenstra (eds.), Wiley, N.Y., 215-310 (1997).

[12] S. Khanna, R. Motwani, M. Sudan, and U. Vazirani, On syntactic versus computational views of approximability. In *Proc. 35th IEEE Symp. on Found. Comput. Sci.* (1994).

[13] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys, *The travelling salesman problem.* Wiley, Chichester (1985).

[14] S. Lin, Computer solutions to the travelling salesman problem. *Bell System Tech. J.* **44,** 2245-2269 (1965).

[15] C.N. Potts and S.L. van de Velde, Dynasearch - Iterative local improvement by dynamic programming: Part I, the travelling salesman problem. *Technical Report.* Univ. Twente, The Netherlands (1995).

[16] A.P. Punnen, The traveling salesman problem: new polynomial approximation algorithms and domination analysis. Manuscript, 1996.

[17] A.P. Punnen and F. Glover, Implementing ejection chains with combinatorial leverage for the TSP, submitted.

[18] C. Rego, Relaxed tours and path ejections for the traveling salesman problem. *Technical Report.* Lab. PRiSM, Univ. de Versailles (1996).

[19] G. Reinelt, *The traveling salesman problem: Computational Solutions for TSP Applications. Springer Lecture Notes in Computer Sci.* **840**, Springer-Verlag, Berlin (1994).

[20] V.I. Sarvanov and N.N. Doroshko, Approximate solution of the traveling salesman problem by a local algorithm with scanning neighbourhoods of factorial cardinality in cubic time. In *Software: Algorithms and Programs.* Math. Institute of Belorussian Acad. of Sci., Minsk, no. 31, 11-13 (1981) (in Russian).