https://doi.org/10.26552/com.C.2003.4.27-35

Jaroslav Janáček – Juraj Hurtík *

AN IMPACT OF IMPROVEMENT-EXCHANGE HEURISTICS TO QUALITY OF PROBABILISTIC TSP SOLUTION

This paper deals with a probabilistic travelling salesman problem (PTSP), which differs from a travelling salesman problem (TSP) [6] in the demand for a customer visit. In PTSP is each customer visited with a given probability only. An objective function for PTSP is in general hard to enumerate and it is even harder to estimate an influence of a local change of a PTSP solution on the final value of the objective function of the resulting solution. This hardiness complicates construction of an efficient heuristic for this problem. In this contribution we have focused on research of a relation between type of local change operation and the resulting objective function improvement. We have constrained here ourselves to improvement-exchange heuristics only, whereas inserting heuristics were broadly studied in the previous works [3], [5]. In this work we have tried to suggest several types of feasible solution changes (operations) including estimation of their impact on the objective function value. These operations have been embedded into the best admissible strategy and implemented as a part of software system, which enables exact or approximate evaluation of a PTSP route depending on the problem size. Using this system we have performed a sequence of numerical experiments to reveal relation between complexity of the used heuristics operations and final improvement of the resulting objective function value. This numerical result and associated conclusions are reported in the concluding part of this contribution.

1. Introduction

The probabilistic travelling salesman problem (PTSP) is a generalization of a well-known travelling salesman problem (TSP). Each node in PTSP has to be visited with some probability and visit necessities of all these nodes are stochastically independent. This problem was first introduced by Jaillet [2].

Let us denote the probability of the visit at node i by p_i. An actual visit of the node i is represented by a random binary variable $N_i \in \{0, 1\}$, for which the value 1 that appears with probability p_i, means a visit of node and the value 0 (with $(1-p_i)$ probability) means that the node is not to be visited. The distance between the nodes i, j is denoted by d_{ij}.

As we can see, the original TSP is just a special case of the problem presented above, where visit necessity probabilities of all nodes are set to 1. This difference makes a PTSP more difficult to solve, as it is ineffective to find a route every day according to the current set of customers (all nodes i with the current value $N_i = 1$). Thus, we need to find a solution for TSP that respects probabilities of all customers and an expected route length of, which, after skipping inactive customers, will be minimal. The main problem is to find the mentioned solution, which is called an a priori route.

In accordance with the above-mentioned problem we denote $\{1, 2, \ldots n\}$ set of customers, which should be visited whenever they are active (when their random variable value is 1). Each customer j raises his demand for visiting with probability p_j. Our goal is to determine such a sequence of all customers so that the expected length of the tour be minimal. An instance of the tour arises by skipping customers that are not active in the given instance from the sequence.

When we fix the position of one customer in the considered sequence, then the sequence $\{j(1), j(2), \ldots j(n)\}$ is uniquely determined by the solution.

The evaluation of objective function constitutes a key problem of PTSP modelling. As we mentioned in the first part of this contribution, the constants d_{ij} represent distances between customers. Let us denote $I = \{i(1), i(2), \ldots i(|I|)\}$ as increasing-way ordered set of indexes of all active customers from the sequence. The probability of the sequence given by the sequence I is:

$$\prod_{i \in I} p_{j(i)} \prod_{i \in N-I} (1 - p_{j(i)}) \tag{1}$$

The corresponding length of the route is then:

$$d_{j(i(|I|))j(i(1))} + \sum_{k=1}^{|I|-1} d_{j(i(k))j(i(k+1))} \tag{2}$$

From (1) and (2), we can write the expected value of route length for general sequence $j(1), j(2), \ldots j(n)$ as:

$$\sum_{I \subseteq N} \left(d_{j(i(|I|))j(i(1))} + \sum_{k=1}^{|I|-1} d_{j(i(k))j(i(k+1))} \right) \cdot \\ \cdot \prod_{i \in I} p_{j(i)} \prod_{i \in N-I} (1 - p_{j(i)}) . \tag{3}$$

The expression (3) is very hard to enumerate because the embedded sum goes over all subsets I of the set N and it means that 2^n items have to be processed. This complicated objective function evaluation constitutes a serious problem when an opera-

* **Jaroslav Janáček, Juraj Hurtík**
University of Žilina, Faculty of Management Science and Informatics, E-mail: jardo@frdsa.fri.utc.sk, jurhur@student.dorm.utc.sk

tion changing a given solution is suggested and the associated change of the objective function should be estimated. This problem, as concerned heuristics starting with a feasible solution, does not arise, when a classical TSP is solved.

2. Improvement-exchange operations for TSP

The improvement-exchange operations for the probabilistic travelling salesman problem were derived from the basic improvement-exchange operations for the classical travelling salesman problem. We have considered two types of the operations and both of them will process so called chains of a current TSP route.

The TSP route will be given by sequence $J = < J(1), J(2), ..., J(n), J(1)>$ of n network nodes, which should be visited in the given order. A chain of this route is an arbitrary subsequence of the sequence J. Let us assume that only such chains will be taken into consideration, which do not contain the node $J(1)$. Then a chain may be determined by index i_p of its predecessor $P = J(i_p)$ and by index i_s of its successor $S = J(i_s)$. The very chain is formed by the following sequence of the nodes $J(i_p + 1)$, $J(i_p + 2)$, $J(i_p + 3)$, ..., $J(i_s - 1)$. The first node $B = J(i_p + 1)$ and the last one $E = J(i_s - 1)$ of the chain are referred as the beginning and the end of the chain respectively. If P and S are neighbouring nodes, then the associated chain is said to be empty what means that it has no node. The number of nodes of the chain is referred to as length of the chain. Then the chain can be determined by index i_p if the chain predecessor and by chain length L. In this case, the following equalities hold: $P = J(i_p)$, $S = J(i_p + L + 1)$,. $i_s = i_p + L + 1$ and under assumption $L > 0$, the additional equalities hold $B = J(i_p + 1)$, $i_b = i_p+1$, $E = J(i_p - L)$, $i_e = i_p + L = i_s - 1$.

The first type of the suggested operations is a unary operation defined on a set of chains of J, the length of which is greater or equal to 2. This unary operation is called inversion and consists of taking the chain off the sequence J and inserting it back at the same position in the reversed order. If the current solution is given by the sequence $J = <J(1), J(2), ..., J(i_p), J(i^b), J(i_b + 1), ..., J(i_e - 1), J(i_e), J(i_s), ..., J(n), J(1)>$, then, after the inversion of the chain $C = <J(i_b), J(i_b + 1),..., J(i_e - 1), J(i_e)>$, the following solution arises $J = <J(1), J(2), ..., J(i_p), J(i_e), J(i_e - 1), ..., J(i_b + 1), J(i_b), J(i_s), ..., J(n), J(1)>$ (Fig. 1). Whereas the travelled distance of the current solution is $d = \sum_{i=1}^{n} d_{J(i),\, J(i+1)}$, the travelled distance of the new solution is

$$\underline{d} = \sum_{i=1}^{i_p-1} d_{J(i),\, J(i+1)} + d_{J(i_p),\, J(i_e)} + \sum_{i=i_b}^{i_e-1} d_{J(i+1),\, J(i)} + d_{J(i_b),\, J(i_s)} + \sum_{i=i_s}^{n} d_{J(i),\, J(i+1)} .$$

The improvement of the classical travelling salesman problem provided by this operation can be expressed by the following difference of the arc lengths

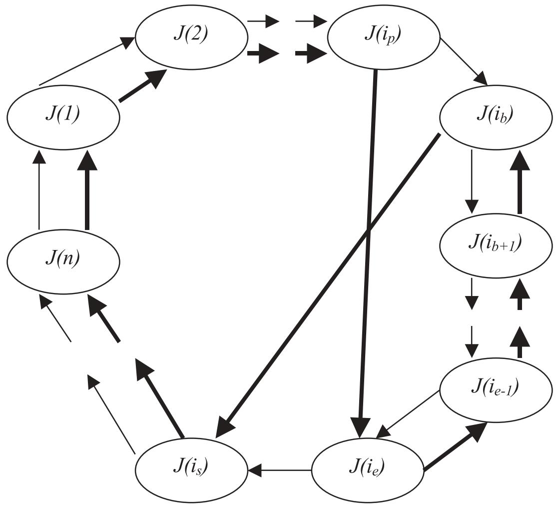

Fig. 1: Improvement of TSP via unary operation

$$d - \underline{d} = d_{J(i_p),\, J(i_b)} + d_{J(i_e),\, J(i_s)} - d_{J(i_p),\, J(i_e)} - d_{J(i_b),\, J(i_s)} + \left[\sum_{i=i_b}^{i_e-1} d_{J(i),\, J(i+1)} - \sum_{i=i_b}^{i_e-1} d_{J(i+1),\, J(i)} \right]. \tag{4}$$

If a symmetrical network is considered, where $d_{ij} = d_{ji}$ holds for arbitrary pair of indices i, j, then the last term of (4) is zero and the improvement is

$$d - \underline{d} = d_{J(i_p),\, J(i_b)} + d_{J(i_e),\, J(i_s)} - d_{J(i_p),\, J(i_e)} - d_{J(i_b),\, J(i_s)} = d_{PB} + d_{ES} - d_{PE} - d_{BS}.$$

The second type of the suggested operations is a binary operation defined on pairs of disjoint chains of J. Two chains of J are said to be disjoint, if they do not share any node nor the predecessor or successor of the other chain. It is only allowed that the successor of one chain is the predecessor of the other one. The further described binary operation is called mutual exchange of two chains and consists of removing both chains from their positions and of inserting them at the opposite positions.

We shall distinguish two cases of chain exchanging operations. In the first case one of these chains is considered to be empty and in the second case the lengths of both chains are greater than zero.

Taking into account the first case, let us study the situation, when the inserted chain must follow its original direction of the travelling salesman route. We denote i_{p1}, i_{b1}, $ie1$ and i_{s1} the indices of the first non-empty chain and i_{p2}, i_{s2} the indices of the second empty chain predecessor and successor. If the current solution J has the form of $J = <J(1), J(2), ..., J(i_{p1}), J(i_{b1}), ..., J(i_{e1}), J(i_{s1}), ..., J(i_{p2}), J(i_{s2}), ..., J(n), J(1)>$, then after the exchange operation the following route comes into being $J = <J(1), J(2), ..., J(i_{p1}), J(i_{s1}), ..., J(i_{p2}), J(i_{b1}), ..., J(i_{e1}), J(i_{s2}), ..., J(n), J(1)>$ (Fig. 2).

The possible improvement can be described by

$$d - \underline{d} = d_{J(i_{p1}),\,J(i_{b1})} + d_{J(i_{e1}),\,J(i_{s1})} - d_{J(i_{p2}),\,J(i_{e2})} -$$

$$- d_{J(i_{p1}),\,J(i_{s1})} - d_{J(i_{p2}),\,J(i_{b1})} - d_{J(i_{e1}),\,J(i_{s2})} =$$

$$= d_{P1B1} + d_{E1S1} + d_{P2S2} - d_{P1S1} - d_{P2B1} - d_{E1S2}\,.$$

where $P_1 = J(i_{p1})$, $B_1 = J(i_{b1})$, $E_1 = J(i_{e1})$, $S_1 = J(i_{s1})$, $P_2 = J(i_{p2})$ and $S_2 = J(i_{s2})$.

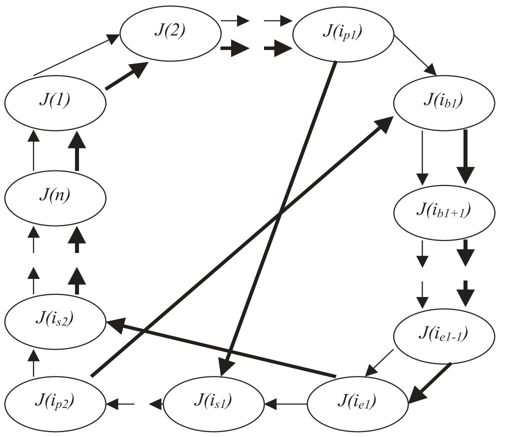

Fig. 2: Improvement of TSP via binary operation

If the non-empty chain is inserted at its new position in the reversed order and the distance matrix is symmetrical, then the associated improvement is

$$d - \underline{d} = d_{P1B1} + d_{E1S1} + d_{P2S2} - d_{P1S1} - d_{P2E1} - d_{B1S2}\,.$$

Considering the second case for a symmetrical distance matrix, we have to distinguish in general four possibilities, how to insert the exchanged chains at their new positions.

The basic possibility is, to let both newly inserted chains follow their former directions from the original route. Then we obtain from the original sequence $J = \langle J(1), J(2), ..., J(i_{p1}), J(i_{b1}), ..., J(i_{e1}), J(i_{s1}), ..., J(i_{p2}), J(i_{b2}), ..., J(i_{e2}), J(i_{s2}), ..., J(n), J(1)\rangle$, this one: $\underline{J} = \langle J(1), J(2), ..., J(i_{p1}), J(i_{b2}), ..., J(i_{e2}), J(i_{s1}), ..., J(i_{p2}), J(i_{b1}), ..., J(i_{e1}), J(i_{s2}), ..., J(n), J(1)\rangle$. and the associated improvement is

$$d - \underline{d} = d_{P1B1} + d_{E1S1} + d_{P2B2} + d_{E2S2} - d_{P1B2} -$$

$$- d_{E2S1} - d_{P2B1} - d_{E1S2}.$$

Further possibility is to make inversion of the first chain and to let the second chain follow its former orientation. Then we obtain from the original sequence J the new sequence $\underline{J} = \langle J(1), J(2), ..., J(i_{p1}), J(i_{b2}), ..., J(i_{e2}), J(i_{s1}), ..., J(i_{p2}), J(i_{e1}), ..., J(i_{b1}), J(i_{s2}), ..., J(n), J(1)\rangle$. with the associated improvement

$$d - \underline{d} = d_{P1B1} + d_{E1S1} + d_{P2B2} + d_{E2S2} - d_{P1B2} -$$

$$- d_{E2S1} - d_{P2E1} - d_{B1S2}\,.$$

The third possibility lets the first chain follow its former orientation and makes inversion of the second. This exchange leads to $\underline{J} = \langle J(1), J(2), ..., J(i_{p1}), J(i_{e2}), ..., J(i_{b2}), J(i_{s1}), ..., J(i_{p2}), J(i_{b1}), ..., J(i_{e1}), J(i_{s2}), ..., J(n), J(1)\rangle$. with

$$d - \underline{d} = d_{P1B1} + d_{E1S1} + d_{P2B2} + d_{E2S2} - d_{P1E2} -$$

$$- d_{B2S1} - d_{P2B1} - d_{B1S2}\,.$$

The last possibility consists of inversion of both chains and of exchange of their positions. This way the following result is obtained from J

$\underline{J} = \langle J(1), J(2), ..., J(i_{p1}), J(i_{e2}), ..., J(i_{b2}), J(i_{s1}), ..., J(i_{p2}), J(i_{e1}), ..., J(i_{b1}), J(i_{s2}), ..., J(n), J(1)\rangle$. with the associated improvement

$$d - \underline{d} = d_{P1B1} + d_{E1S1} + d_{P2B2} + d_{E2S2} - d_{P1E2} -$$

$$- d_{B2S1} - d_{P2E1} - d_{B1S2}\,.$$

Considering the above-mentioned operations for the classical travelling salesman problem, the change of the objective function value caused by the particular operation is immediately known.

Unfortunately, we can hardly estimate the impact of the individual improvement to the resulting objective function value. In fact, such a situation can occur that application of the most advantageous operation will prevent the algorithm from following some sequence of less advantageous operations, which could reach a very good solution.

3. Strategies for PTSP Heuristics

Considering the classical travelling salesman problem, at least the local impact of an improvement-exchange operation on the objective function value of the improved solution can be enumerated exactly and fast enough. The similar approach to probabilistic TSP is almost impossible due complexity of the objective function. We can do here some conditioned assumptions and study the particular strategy impact on the total improvement of the objective function of the tested problems. The problem originates from the fact that the shortening of the TSP route need not bring any saving in the expected value of PTSP route length. Nevertheless we can judge that the improvement described in the previous section could take part on the PTSP objective function improvement proportionally to the probability, with which the employed nodes occur in the PTSP route.

The classical strategies of combinatorial heuristics differ in the way in which a so-called neighbourhood of a current solution is processed. The neighbourhood of some solution is defined as the set of solutions, which can be reached from the current one using only one operation from a set of permitted operations.

For example, if inversion of chain of length two is the only permitted operation, then the neighbourhood of a current solution is formed by all the results which can be obtained by inversion of one chain of length equal to two in the current solution.

There are two basic strategies how to process a current solution neighbourhood. They are the first and best admissible strategies. The strategy "First Admissible" (FA) tests the permitted operations on the current solution in a given order and whenever it finds the operation with positive improvement (admissible operation or move), it performs the associated move to the better solution and updates the current solution. This process is repeated until such a current solution is met where no admissible operation exists. The strategy "Best Admissible" (BA) searches through the whole neighbourhood and updates the best admissible operation, i.e. the operation with the greatest positive improvement. After the neighbourhood searching is completed, the best admissible operation is performed and the current solution is updated. This cycle is repeated until no admissible operation on the current solution is found.

To design PTSP strategy, we refused the first admissible one, because it does not allow us to employ the probabilities of the nodes associated with the given operation. The first admissible strategy performs the first admissible move without comparing it to the others.

That is why we decided to use the best admissible strategy which tests all permitted operations on the current solution and computes the associated improvement as shown in the previous section. To employ the visit probabilities of the nodes which are associated with the considered operation, we expressed the probability that the substantial nodes occur in the necessary configuration.

For example, if the inversion of the chain with predecessor *P*, beginning *B*, end *E* and successor *S* should be performed, it is necessary for them to be visited in the given instance. The probability that all the nodes will be visited simultaneously is $p_{in} = p_P * p_B * p_E * p_S$.

The best admissible strategy enables to weigh each improvement associated with operation by the associated probability of the node configuration and to select the best expected improvement and the associated operation. In the suggested improvement-exchange heuristic, we assigned to operation of inversion the above-mentioned probability $p_{in} = p_P * p_B * p_E * p_S$.

When chain exchanges are considered, then if the first chain has its length equal to one and the second one has its length equal to zero, then $p_{ex} = p_{P1} * p_{B1} * p_{S1} * p_{P2} * p_{S2}$.

If the first chain has the length greater than one, then $p_{ex} = p_{P1} * p_{B1} * p_{E1} * p_{S1} * p_{P2} * p_{S2}$. If the both chains have their lengths equal to one, then $p_{ex} = p_{P1} * p_{B1} * p_{S1} * p_{P2} * p_{B2} * p_{S2}$ and if both chains have their lengths greater than one, then $p_{ex} = p_{P1} * p_{B1} * p_{E1} * p_{S1} * p_{P2} * p_{B2} * p_{E2} * p_{S2}$.

4. Numerical Experiments

To verify and evaluate our improvement-exchange heuristics, we have decided to study their behaviour on the set of sixty PTSP problems, which were derived from the problems referred in [1]. This set of benchmark problems consists of six subsets, where each subset contains ten problems of the same size. In the subsets, problem sizes of 20, 30, 50, 100, 150 and 250 nodes were considered. The associated subsets were denoted by M020, M030, M050, M100, M150 and M200 respectively. As the improvement-exchange heuristic needs a starting solution and quality of this solution may influence the contribution of the tested heuristic, we used two different dual heuristics to obtain various starting solutions. The improvement phase was performed with each of these solutions. The dual heuristics used to obtain the starting solution were described and studied in [3] and [4]. This set of dual heuristics contains the so-called Pelikan's algorithm [5] and Repeated Nearest Neighbour algorithm [3], which are denoted in the following tables as "Pelikan" and "RNN" respectively. As we have decided to explore the dependence of solution quality and time consumption on the complexity of the algorithm, which searches over current solution neighbourhood, we defined the order of improvement-exchange heuristic for this purpose. This order equals to the maximal length of chains, which is processed by the particular heuristic.

This way, the above mentioned algorithm performing only inversions of chains of length equal to two is denoted as heuristic of order $o = 2$. If an inversion heuristic of order three is considered, then each inversion of chain the length of which equals to two or three, must be inspected to process neighbourhood of a current solution. If the chain exchange algorithm is considered, then heuristic of order two must evaluate each pair of chains, the lengths of which are one – zero, two – zero, one – one, two – one, two – two, to select the best move, which should be accomplished to leave the current solution.

To evaluate the objective function value (expected length according to expression (3)), we employed two approaches, enumeration and simulation [3], [4]. For the problems of a smaller size, we suggested and implemented a straightforward enumeration algorithm. This algorithm is based on a last-in-first-out structure and the enumeration scheme for the probability and length of an instance enumeration, which resembles a complete depth-searching process in a branching tree. In the enumeration process we start with an instance in that all customers are active, then the instance is processed in which all customers but the last are active, further the instance with only last but one active customer comes, and as the next case, the instance will be studied with two last inactive customers etc.

A faster way how to estimate the expected length of the route is the use of simulation technique. This method consists of a performed random generation of individual instances and their probabilistic evaluation. As we have found [3] that the enumeration process has exponential time consumption depending on a problem size, the simulation was the only method, which can settle with objective function of solutions containing more than thirty nodes.

In the following tables the average time consumption of improvement exchange heuristic is given in seconds for the particular subset of problems. The numerical experiments were performed on PC P4 1400 MHz, 128 MB RAM.

The solution quality is described by the average value of starting solutions and of the average improvement. This improvement is given once more in the tables in percentage, where the average starting solution is taken as a base (100 %).

Table 1

Problem	Dual heuristic	Average starting value	Primal heuristic	Order	Average improvement	Percentage	Average time [s]
M020	Pelikan	778.48	Inversion	2	-1.62	-0.21	0.00
Enumeration	Pelikan	778.54	Inversion	2	-1.59	-0.20	0.00
M020	Pelikan	778.48	Inversion	3	-1.53	-0.19	0.00
Enumeration	Pelikan	778.54	Inversion	3	-1.58	-0.20	0.00
M020	Pelikan	778.48	Inversion	4	-1.58	-0.23	0.00
Enumeration	Pelikan	778.54	Inversion	4	-1.80	-0.20	0.00
M020	Pelikan	778.48	Inversion	5	-1.55	-0.20	0.00
Enumeration	Pelikan	778.54	Inversion	5	-1.56	0.24	0.00
M020	Pelikan	778.48	Exchange	1	1.89	0.24	0.00
Enumeration	Pelikan	778.54	Exchange	1	1.90	0.78	0.00
M020	Pelikan	778.48	Exchange	2	6.04	0.77	0.00
Enumeration	Pelikan	778.54	Exchange	2	6.00	0.76	0.00
M020	Pelikan	778.48	Exchange	3	5.94	0.77	0.00
Enumeration	Pelikan	778.54	Exchange	3	6.00	0.76	0.00
M020	Pelikan	778.48	Exchange	4	5.88	0.76	0.00
Enumeration	Pelikan	778.54	Exchange	4	6.00	0.77	0.00
M020	Pelikan	778.48	Exchange	5	6.00	0.77	0.00
Enumeration	Pelikan	778.54	Exchange	5	6.00	0.77	0.00

Table 2

Problem	Dual heuristic	Average starting value	Primal heuristic	Order	Average improvement	Percentage	Average time [s]
M020	RNN	918.08	Inversion	2	16.36	1.78	0.00
Enumeration	RNN	918.51	Inversion	2	16.48	1.79	0.00
M020	RNN	918.08	Inversion	3	33.27	3.62	0.00
Enumeration	RNN	918.51	Inversion	3	33.33	3.63	0.00
M020	RNN	918.08	Inversion	4	51.59	5.62	0.00
Enumeration	RNN	918.51	Inversion	4	51.77	5.64	0.00
M020	RNN	918.08	Inversion	5	55.51	6.04	0.00
Enumeration	RNN	918.51	Inversion	5	55.67	6.06	0.00
M020	RNN	918.08	Exchange	1	100.31	10.92	0.00
Enumeration	RNN	918.51	Exchange	1	100.43	10.93	0.00
M020	RNN	918.08	Exchange	2	134.34	14.63	0.00
Enumeration	RNN	918.51	Exchange	2	134.50	14.93	0.00
M020	RNN	918.08	Exchange	3	137.14	14.94	0.00
Enumeration	RNN	918.51	Exchange	3	137.23	15.73	0.00
M020	RNN	918.08	Exchange	4	144.50	15.75	0.00
Enumeration	RNN	918.51	Exchange	4	144.66	15.74	0.00
M020	RNN	918.08	Exchange	5	144.54	15.75	0.00
Enumeration	RNN	918.51	Exchange	5	144.66	15.74	0.00

Table 3

Problem	Dual heuristic	Average starting value	Primal heuristic	Order	Average improvement	Percentage	Average time [s]
M030	Pelikan	1014.61	Inversion	2	0.51	0.05	0.00
Enumeration	Pelikan	1014.57	Inversion	2	0.42	0.04	0.00
M030	Pelikan	1014.61	Inversion	3	-0.77	-0.08	0.00
Enumeration	Pelikan	1014.57	Inversion	3	-0.72	-0.07	0.00
M030	Pelikan	1014.61	Inversion	4	-0.16	-0.02	0.00
Enumeration	Pelikan	1014.57	Inversion	4	-0.30	-0.03	0.00
M030	Pelikan	1014.61	Inversion	5	-0.24	-0.02	0.00
Enumeration	Pelikan	1014.57	Inversion	5	-0.32	-0.03	0.00
M030	Pelikan	1014.61	Exchange	1	-2.63	-0.26	0.00
Enumeration	Pelikan	1014.57	Exchange	1	-2.61	-0.26	0.00
M030	Pelikan	1014.61	Exchange	2	-15.76	-1.57	0.00
Enumeration	Pelikan	1014.57	Exchange	2	-15.92	-1.57	0.00
M030	Pelikan	1014.61	Exchange	3	-16.20	-1.60	0.00
Enumeration	Pelikan	1014.57	Exchange	3	-16.27	-1.60	0.00
M030	Pelikan	1014.61	Exchange	4	-7.42	-0.73	0.01
Enumeration	Pelikan	1014.57	Exchange	4	-7.57	-0.75	0.00
M030	Pelikan	1014.61	Exchange	5	-5.66	-0.56	0.01
Enumeration	Pelikan	1014.57	Exchange	5	-5.82	-0.57	0.00

Table 4

Problem	Dual heuristic	Average starting value	Primal heuristic	Order	Average improvement	Percentage	Average time [s]
M030	RNN	1330.74	Inversion	2	16.49	1.24	0.00
Enumeration	RNN	1330.93	Inversion	2	16.52	1.24	0.00
M030	RNN	1330.74	Inversion	3	18.52	1.39	0.00
Enumeration	RNN	1330.93	Inversion	3	18.68	1.40	0.00
M030	RNN	1330.74	Inversion	4	31.98	2.40	0.00
Enumeration	RNN	1330.93	Inversion	4	31.72	2.38	0.00
M030	RNN	1330.74	Inversion	5	64.04	4.81	0.00
Enumeration	RNN	1330.93	Inversion	5	64.10	4.82	0.00
M030	RNN	1330.74	Exchange	1	113.23	8.51	0.00
Enumeration	RNN	1330.93	Exchange	1	113.48	8.53	0.00
M030	RNN	1330.74	Exchange	2	226.74	17.04	0.01
Enumeration	RNN	1330.93	Exchange	2	226.92	17.05	0.01
M030	RNN	1330.74	Exchange	3	280.20	21.06	0.02
Enumeration	RNN	1330.93	Exchange	3	280.24	21.07	0.02
M030	RNN	1330.74	Exchange	4	285.11	21.53	0.02
Enumeration	RNN	1330.93	Exchange	4	285.28	21.52	0.02
M030	RNN	1330.74	Exchange	5	292.71	22.00	0.03
Enumeration	RNN	1330.93	Exchange	5	292.96	22.01	0.03

Table 5

Problem	Dual heuristic	Average starting value	Primal heuristic	Order	Average improvement	Percentage	Average time [s]
M050	Pelikan	9098.88	Inversion	2	15.34	0.14	0.00
M050	Pelikan	9098.88	Inversion	3	-12.28	-0.13	0.00
M050	Pelikan	9098.88	Inversion	4	-9.24	-0.10	0.00
M050	Pelikan	9098.88	Inversion	5	-20.79	-0.22	0.00
M050	Pelikan	9098.88	Exchange	1	-3.83	-0.04	0.00
M050	Pelikan	9098.88	Exchange	2	-98.68	-1.08	0.01
M050	Pelikan	9098.88	Exchange	3	-89.73	-0.99	0.02
M050	Pelikan	9098.88	Exchange	4	-39.99	-0.44	0.03
M050	Pelikan	9098.88	Exchange	5	30.40	0.33	0.05
M050	RNN	13967.38	Inversion	2	94.63	0.67	0.00
M050	RNN	13967.38	Inversion	3	196.94	1.41	0.00
M050	RNN	13967.38	Inversion	4	284.12	2.06	0.00
M050	RNN	13967.38	Inversion	5	357.90	2.56	0.00
M050	RNN	13967.38	Exchange	1	2433.36	14.40	0.02
M050	RNN	13967.38	Exchange	2	3451.34	24.68	0.06
M050	RNN	13967.38	Exchange	3	3868.67	27.66	0.11
M050	RNN	13967.38	Exchange	4	4162.85	29.77	0.18
M050	RNN	13967.38	Exchange	5	4577.41	32.73	0.26

Table 6

Problem	Dual heuristic	Average starting value	Primal heuristic	Order	Average improvement	Percentage	Average time [s]
M100	Pelikan	11291.04	Inversion	2	38.29	0.33	0.00
M100	Pelikan	11291.04	Inversion	3	139.75	1.23	0.00
M100	Pelikan	11291.04	Inversion	4	110.33	0.97	0.00
M100	Pelikan	11291.04	Inversion	5	437.66	3.25	0.00
M100	Pelikan	11291.04	Exchange	1	200.93	1.77	0.02
M100	Pelikan	11291.04	Exchange	2	279.40	2.47	0.07
M100	Pelikan	11291.04	Exchange	3	330.47	2.92	0.18
M100	Pelikan	11291.04	Exchange	4	310.40	2.74	0.28
M100	Pelikan	11291.04	Exchange	5	309.22	2.73	0.43
M100	RNN	17501.28	Inversion	2	318.64	1.82	0.00
M100	RNN	17501.28	Inversion	3	562.79	3.21	0.00
M100	RNN	17501.28	Inversion	4	829.97	4.71	0.00
M100	RNN	17501.28	Inversion	5	1038.6	5.93	0.00
M100	RNN	17501.28	Exchange	1	3447.5	19.69	0.15
M100	RNN	17501.28	Exchange	2	4473.37	25.77	0.41
M100	RNN	17501.28	Exchange	3	4831.20	27.60	0.76
M100	RNN	17501.28	Exchange	4	5128.79	29.30	1.09
M100	RNN	17501.28	Exchange	5	5197.54	29.69	1.59

Table 7

Problem	Dual heuristic	Average starting value	Primal heuristic	Order	Average improvement	Percentage	Average time [s]
M150	Pelikan	28194.35	Inversion	2	12.23	0.04	0.01
M150	Pelikan	28194.35	Inversion	3	20.59	0.07	0.00
M150	Pelikan	28194.35	Inversion	4	40.97	0.136	0.00
M150	Pelikan	28194.35	Inversion	5	30.35	0.10	0.00
M150	Pelikan	28194.35	Exchange	1	287.88	1.02	0.08
M150	Pelikan	28194.35	Exchange	2	430.04	1.52	0.31
M150	Pelikan	28194.35	Exchange	3	454.66	1.61	0.59
M150	Pelikan	28194.35	Exchange	4	470.26	1.65	0.91
M150	Pelikan	28194.35	Exchange	5	475.27	1.68	1.45
M150	RNN	45080.75	Inversion	2	827.10	1.83	0.00
M150	RNN	45080.75	Inversion	3	1146.54	2.54	0.00
M150	RNN	45080.75	Inversion	4	1493.51	3.31	0.00
M150	RNN	45080.75	Inversion	5	1791.81	4.13	0.00
M150	RNN	45080.75	Exchange	1	10578.41	23.46	0.54
M150	RNN	45080.75	Exchange	2	12715.01	28.20	1.39
M150	RNN	45080.75	Exchange	3	13964.41	30.97	2.48
M150	RNN	45080.75	Exchange	4	14953.12	33.16	4.07
M150	RNN	45080.75	Exchange	5	15076.52	33.44	5.71

Table 8

Problem	Dual heuristic	Average starting value	Primal heuristic	Order	Average improvement	Percentage	Average time [s]
M250	Pelikan	11637.91	Inversion	2	4.21	0.03	0.00
M250	Pelikan	11637.91	Inversion	3	-3.21	-0.02	0.00
M250	Pelikan	11637.91	Inversion	4	13.60	0.11	0.00
M250	Pelikan	11637.91	Inversion	5	25.36	0.21	0.00
M250	Pelikan	11637.91	Exchange	1	118.83	1.02	3.96
M250	Pelikan	11637.91	Exchange	2	150.52	1.29	4.02
M250	Pelikan	11637.91	Exchange	3	231.43	1.98	3.60
M250	Pelikan	11637.91	Exchange	4	263.78	2.27	5.62
M250	Pelikan	11637.91	Exchange	5	288.40	2.47	8.04
M250	RNN	17824.70	Inversion	2	321.80	1.80	0.00
M250	RNN	17824.70	Inversion	3	455.90	2.55	0.00
M250	RNN	17824.70	Inversion	4	681.64	3.82	0.00
M250	RNN	17824.70	Inversion	5	884.54	4.96	0.00
M250	RNN	17824.70	Exchange	1	3352.14	18.80	4.35
M250	RNN	17824.70	Exchange	2	3980.14	22.32	10.87
M250	RNN	17824.70	Exchange	3	4732.49	26.11	15.52
M250	RNN	17824.70	Exchange	4	4820.04	27.04	24.89
M250	RNN	17824.70	Exchange	5	4883.50	27.39	33.70

As can be seen in the Tables 1–4 in the column "Average starting value", the results obtained by simulation and enumeration processes are almost identical. It supports our hypothesis that the simulation process can be used for evaluation of results in the cases, when the enumeration process fails due to time complexity.

From the comparison of the results obtained by a two-phase procedure Pelikan – improvement exchange and the results obtained by Nearest-Neighbour – improvement exchange, it can be found that the Pelikan's dual heuristic provides very good results which can not be improved by the studied primal heuristics. It can be noticed that this improvement exchange heuristic may deteriorate the input solution to some small extent. On the contrary to the procedure Pelikan – improvement exchange, the improvement exchange heuristic approved its use when a less quality input solution produced by the Nearest-Neighbour algorithm is used as a starting solution.

As regards of larger size from 100 to 250 customers (Tables 6–8), a small improvement from 1 to 2 % can be found even when improvement – exchange heuristic is used to Pelikan's resulting solution. Comparing it with the results obtained in the experiments with the Repeated Nearest-Neighbour procedure, it can be seen that the improvement ranges from 2 to 30 % in these cases.

5. Conclusions

The improvement – exchange heuristic, whose advantage was tested in the above mentioned experiments, turned out to be very useful when a good solution of the probabilistic travelling salesman problem is sought. Especially in the larger instances when the starting solution is obtained by common dual heuristic, there is employment of an improvement – exchange heuristic unavoidable. Furthermore, the experiments showed a considerable increase of the obtained improvement connected with raising order of the heuristic, especially when chain exchange is performed instead of simple inversion.

Similar to our previous results [3], [4], the Pelikan's heuristic proved to be able to provide excellent solutions which are hard to improve and which seem to be a local minima with respect to the used exchange operations. We think that it is caused by the fact that our exchange - improvement heuristic took into consideration only probability of the inspected improvement, which is evaluated using only the edges between neighbouring nodes in a current route. This approach does not explore the mutual positions of the customers with higher probability, which are separated by customers with lower probability in the current route.

In contradiction to this insufficiency the Pelikan's heuristic inserts customers to a route in accordance to their decreasing probability and at each step it locates a current customer into the current route, in which each customer has greater probability than the current one. We can suggest the study of improvement – exchange heuristic overcoming above-mentioned insufficiency for a future research.

Acknowledgement
This paper is supported by the grant VEGA 1/0498/03.

References

[1] ftp://ftp.zib.de/pub/Packages/mp-testdata/tsp/tsplib/., internet server with TSP benchmarks
[2] JAILLET, P.: *A priori solution of the traveling salesman problem in which a random subset of customers are visited.*, Operation Research 36, 1988
[3] JANÁČEK, J., HURTÍK, J.: *Heuristics algorithm and evaluation of probabilistic traveling salesman problem solution.* Journal of Information, Control and Management Systems, ŽU Žilina, 2003, to appear
[4] JANÁČEK, J., HURTÍK, J.: *Simulation and evaluation of a probabilistic traveling salesman problem.* In: Proceedings of the conference Transcom 2003, Žilina
[5] PELIKÁN, J.: *A modified insert algorithm for the probabilistic traveling salesman problem.* In: Proceedings of the 8th International Scientific Conference Quantitative Methods in Economy and Business-Methodology and Practice in the New Millenium, Bratislava September 18-20, 2002, pp 634-638
[6] PEŠKO, Š., HURTÍK, J.: *Solving TSP via SQL queries.* In: Proceedings of the conference Quantitative Methods in Economics (Multiple Criteria Decision Making XI), Nitra December 5-6, 2002, pp 194-197
[7] PEŠKO, Š.: *The pyramid method for traveling salesman problem (in slovak).* Communications, 2000, No. 4, pp. 29-34