Optimization Letters (2008) 2:351–361
DOI 10.1007/s11590-007-0064-3

ORIGINAL PAPER

Expanding neighborhood search–GRASP for the probabilistic traveling salesman problem

Yannis Marinakis · Athanasios Migdalas · Panos M. Pardalos

Received: 11 April 2007 / Accepted: 10 August 2007 / Published online: 1 September 2007

Abstract The Probabilistic Traveling Salesman Problem is a variation of the classic traveling salesman problem and one of the most significant stochastic routing problems. In probabilistic traveling salesman problem only a subset of potential customers need to be visited on any given instance of the problem. The number of customers to be visited each time is a random variable. In this paper, a variant of the well-known Greedy Randomized Adaptive Search Procedure (GRASP), the Expanding Neighborhood Search–GRASP, is proposed for the solution of the probabilistic traveling salesman problem. expanding neighborhood search–GRASP has been proved to be a very efficient algorithm for the solution of the traveling salesman problem. The proposed algorithm is tested on a numerous benchmark problems from TSPLIB with very satisfactory results. Comparisons with the classic GRASP algorithm and with a Tabu Search algorithm are also presented. Also, a comparison is performed with the results of a number of implementations of the Ant Colony Optimization algorithm from the literature and in six out of ten cases the proposed algorithm gives a new best solution.

Keywords Expanding neighborhood search–GRASP · Metaheuristics · Probabilistic traveling salesman problem

Y. Marinakis (✉) · A. Migdalas
Decision Support Systems Laboratory, Department of Production Engineering and Management, Technical University of Crete, 73100 Chania, Greece
e-mail: marinakis@ergasya.tuc.gr

A. Migdalas
e-mail: migdalas@ergasya.tuc.gr

P. M. Pardalos
Department of Industrial and Systems Engineering, Center of Applied Optimization, University of Florida, Gainesville, USA
e-mail: pardalos@cao.ise.ufl.edu

1 Introduction

The **Probabilistic Traveling Salesman Problem (PTSP)** is an extension of the well-known Traveling Salesman Problem (TSP), which has been extensively studied in the field of combinatorial optimization. The probabilistic traveling salesman problem is perhaps the most fundamental stochastic routing problem that can be defined [26]. It was introduced in 1985 by Jaillet in his Ph.D. thesis [17]. Some theoretical properties of the PTSP derived in [17] have been later published in [19]. In the PTSP, a demand at each node occurs (with probability p), or does not occur (with probability $1 - p$) during a given day. The main difference of the probabilistic traveling salesman problem from the traveling salesman problem is that while in the TSP the objective is to find the shortest tour through all the cities such that no city is visited twice and the salesman returns at the end of the tour back to the starting city, in the PTSP the objective is to minimize the expected length of the a priori tour where each customer requires a visit only with a given probability. The a priori tour can be seen as a template for the visiting sequence of all customers. In a given instance, the customers should be visited based on the sequence of the a priori tour while the customers who do not need to be visited will simply be skipped [22]. The PTSP belongs to the class of NP-hard problems [1]. This means that no polynomial time algorithm is known for its solution. Thus, there is a great need for powerful heuristics that find good suboptimal solutions in reasonable amounts of time.

A formulation of the problem is the following [4,17]: Let a full connected graph whose the nodes set is denoted by $V = \{1, 2, \ldots, n\}$. Each node i $(i = 1, 2, \ldots, n)$ is associated with a presence probability p_i that represents the possibility that node i will be present in a given solution. Given an a priori tour τ, if problem instance $S(\subseteq V)$ will occur with probability $p(S)$ and will require covering a total distance $L_\tau(S)$ to visit the subset S of customers, that problem instance will receive a weight of $p(S)L_\tau(S)$ in the computation of the expected length. If we denote the length of the tour τ by L_τ (a random variable), then our problem is to find an a priori tour through all n potential customers, which minimizes the quantity

$$E[L_\tau] = \sum_{S \subseteq V} p(S) L_\tau(S) \tag{1}$$

with the summation being over all subsets of V. The probability for the subset of customers S to require a visit is:

$$p(S) = \prod_{i \in S} p_i \prod_{i \in V - S} (1 - p_i). \tag{2}$$

In fact, let us consider (without loss of generality) an a priori tour $\tau = (1, 2, \ldots, n)$ then its expected length is

$$E[L_\tau] = \sum_{i=1}^{n} \sum_{j=i+1}^{n} d_{ij} p_i p_j \prod_{k=i+1}^{j-1} (1 - p_k) + \sum_{i=1}^{n} \sum_{j=i+1}^{n} d_{ij} p_i p_j \prod_{k=i+1}^{n} (1 - p_k) \prod_{l=1}^{j-1} (1 - p_l) \tag{3}$$

where d_{ij} is the distance between the nodes i and j.

This expression is derived by looking at the probability for each arc of the complete graph to be used, that is, when the a priori tour is adapted by skipping a set of customers which do not require a visit.

Since the introduction of the problem in 1985 by Jaillet [17], a number of heuristics and metaheuristics algorithms have been proposed for its solution. Initially, researchers focused on heuristics [2,3,18,27,29]. Recent studies focus on adopting new algorithmic approaches based on metaheuristics such as ant colony optimization (ACO) [4–7,9], genetic algorithm [23], threshold accepting (TA) [30], scatter search [22] and stochastic annealing [8]. An aggregation method for the solution of the probabilistic traveling salesman problem is presented in [10]. An exact algorithm based on an integer L-shaped method has been used to solve 50-node instances [20] while in [11] a stochastic dynamic programming algorithm has been applied.

In this paper, we demonstrate how a variant of the Greedy Randomized Adaptive Search Procedure (GRASP) [28], the expanding neighborhood search–GRASP [25] can be used, in order to give very good solutions in the probabilistic traveling salesman problem. The main contribution of the paper is the application of a very efficient algorithm, the ENS–GRASP algorithm, for the solution of the probabilistic traveling salesman problem that gives new best solutions in most instances used in the comparisons. The rest of the paper is organized as follows: In Sect. 2 the proposed algorithm, the expanding neighborhood search–GRASP is presented and analyzed in detail. Computational results are presented and analyzed in Sect. 3 while in Sect. 4 conclusions and future research are given.

2 Expanding neighborhood search–GRASP for the probabilistic traveling salesman problem

2.1 Introduction

Greedy Randomized Adaptive Search Procedure [12,28] is an iterative two phase search method which has gained considerable popularity in combinatorial optimization. Each iteration consists of two phases, a **construction phase** and a **local search phase**. In the construction phase, a randomized greedy function is used to built up an initial solution. This randomized technique provides a feasible solution within each iteration. This solution is then exposed for improvement attempts in the local search phase. The final result is simply the best solution found over all iterations. In the first phase, a **randomized greedy technique** provides feasible solutions incorporating both greedy and random characteristics. This phase can be described as a process which stepwise adds one element at a time to the partial (incomplete) solution. The choice of the next element to be added is determined by ordering all elements in a candidate list (Restricted Candidate List—RCL) with respect to a greedy function. The heuristic is adaptive because the benefits associated with every element are updated during each iteration of the construction phase to reflect the changes brought on by the selection of the previous element. The probabilistic component of a **GRASP** is characterized by randomly choosing one of the best candidates in the list but not necessarily the top candidate. The greedy algorithm is a simple one pass procedure for solving the problem (the probabilistic traveling salesman problem in our case). In the second phase, a **local search** is initialized from these points, and the final result is simply the best solution found over all searches (cf. multi-start local search). Marinakis et al. [25] proposed an algorithm for the solution of the traveling salesman problem that adds new features to the original GRASP algorithm and has been proved very efficient for the solution of the problem.

The expanding neighborhood search–GRASP (ENS–GRASP) [24,25] consists of two phases. In the **first phase**, initial feasible solutions are produced and in the **second phase**, the expanding neighborhood search is utilized in order to improve these solutions. In the *first phase*, initially, an array is converted into a heap using the heap data structure, then the RCL is constructed, from where the candidate elements for inclusion in the partial solution are selected randomly, and the RCL is readjusted. Finally, a step-wise construction algorithm is used for the construction of the initial feasible solution. The second phase of the algorithm consists of the expanding neighborhood search strategy (see Sect. 2.2.3).

2.2 Main phases of ENS–GRASP

2.2.1 Data representation

In each iteration of the algorithm, a data structure for tour representation is needed. The choice of the data structure is a very significant part of the algorithm as it plays a critical role for the efficiency of the algorithm. The heap and the disjoint set data structures [31] are used in the construction phase. For more details about the data structures see [25].

2.2.2 Construction phase of expanding neighborhood search–GRASP

Initially, the RCL of all the edges of a given graph $G = (V, E)$ is created by ordering all the edges from the smallest to the largest cost using a heap data structure. From this list, the first D edges, where D is chosen empirically after thorough testing, are selected in order to form the final restricted candidate list. This type of RCL is a **cardinality based RCL**. The candidate edge for inclusion in the tour is selected randomly from the RCL using a random number generator. Finally, the RCL is readjusted in every iteration by replacing the edge which has been included in the tour by another edge that does not belong to the RCL, namely the $(D + m)$th edge where m is the number of the current iteration. Once an element has been chosen for inclusion in the tour, a tour construction heuristic is applied in order to insert it in the partial tour.

The construction heuristic is the **pure greedy algorithm**. Initially, the degree of all the nodes is set equal to zero. Then, one edge at a time is selected randomly from the RCL. Each edge (i, j) is inserted in the partial route taking into account that the degree of i and j should be less than 2 and the new edge should not complete a tour with fewer than n vertices. In this algorithm, a number of paths are created which are finally joined to a single tour. For each candidate edge there are the following possibilities:

- If both end-nodes of the edge do not belong to any path (their degree is equal to 0), then they are joined together, a path is created and their degrees are increased by one.

- If the one node belongs to a path and its degree is equal to 1 and the other node is a single node (its degree is equal to 0), then they are joined together and their degrees are increased by one.
- If both nodes belong to different paths and their degrees are equal to 1, then, they are both in the beginning or in the end of their paths and the two paths are joined into a path. The degrees of the nodes are then increased by one.
- If both nodes have degree equal to 1 but they are in the same path, then, there are two possibilities:
  1. If the path has less nodes than the number of nodes in the problem, then, the edge is rejected because if the two nodes are joined, a tour with less nodes (a subtour) would be created.
  2. If the number of nodes in the path is equal to the number of nodes in the problem, then, the two edges are joined producing thus a feasible tour.
- If one or both edges has degree equal to 2, then the edge is rejected and we proceed to the next edge.

2.2.3 Expanding neighborhood search

Expanding neighborhood search (ENS) is a new metaheuristic algorithm [25] that can be used for the solution of a number of combinatorial optimization problems with remarkable results. The main features of this algorithm is the use of the Circle Restricted Local Search Moves Strategy, the ability of the algorithm to change between different local search strategies and the use of an expanding strategy. These features are explained in detail in the following. In the **Circle Restricted Local Search Moves (CRLSM)** strategy [24], the computational time is decreased significantly compared to other heuristic and metaheuristic algorithms because all the edges that are not going to improve the solution are excluded from the search procedure. This happens by restricting the search into circles around the candidate for deletion edges.

In the following, a description of the circle restricted local search moves strategy for a **2-opt trial move** [21] is presented. In this case, there are three possibilities based on the costs of the candidates for deletion and inclusion edges:

- If both new edges increase in cost, a 2-opt trial move can not reduce the cost of the tour (e.g., in Fig. 1, for both new edges the costs $C2$ and $C4$ are greater than the costs $B2$ and A of both old edges).
- If one of the two new edges has cost greater than the sum of the costs of the two old edges, a 2-opt trial move, again, can not reduce the cost of the tour (e.g. in Fig. 1, the cost of the new edge $C3$ is greater than the sum of the costs $A + B3$ of the old edges).
- The only case for which a 2-opt trial move can reduce the cost of the tour is when at least one new edge has cost less than the cost of one of the two old edges (e.g., in Fig. 1, the cost $C1$ of the new edge is less than the cost of the old edge A) and the other edge has cost less than the sum of the costs of the two old edges (e.g., $C5 < A + B1$ in Fig. 1).

Taking these observations into account, the circle restricted local search moves strategy restricts the search to edges where one of their end-nodes is inside a circle with radius

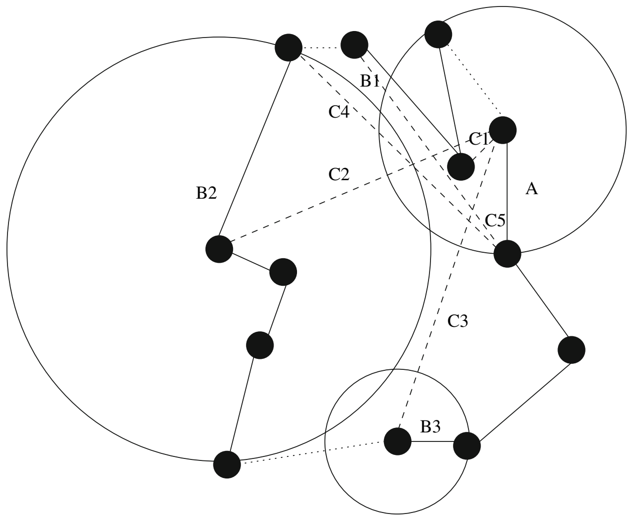

Fig. 1 Circle restricted local search moves strategy

length at most equal to the sum of the costs (lengths) of the two candidates for deletion edges.

The algorithm has the ability to change between different local search strategies. The idea of using a larger neighborhood to escape from a local minimum to a better one, had been proposed initially by Garfinkel and Nemhauser [13] and recently by Hansen and Mladenovic [16]. Garfinkel and Nemhauser proposed a very simple way to use a larger neighborhood. In general, if with the use of one neighborhood a local optimum was found, then a larger neighborhood is used in an attempt to escape from the local optimum. On the other hand, Hansen and Mladenovic proposed a more systematical method to change between different neighborhoods, called Variable Neighborhood Search.

The expanding neighborhood search method starts with one prespecified length of the radius of the circle of the CRLSM strategy. Inside this circle a number of different local search strategies are applied until all the possible trial moves have been explored and the solution cannot further be improved in this neighborhood. Subsequently, the length of the radius of the circle is increased and, again, the same procedure is repeated until the stopping criterion is activated. The main differences of ENS from the other two methods is the use of the circle restricted local search move strategy which restricts the search in circles around the candidates for deletion edges and the more sophisticated way that the local search strategy can be changed inside the circles.

In expanding neighborhood search strategy, the size of the neighborhood is **expanded** in each external iteration [24]. Each different length of the neighborhood constitutes an external iteration. Initially, the size of the neighborhood, s, is defined based on the circle restricted local search moves strategy, for example $s = A/2$, where A is the cost of one of the candidates for deletion edges. For the selected size of the neighborhood, a number of different local search strategies are applied until all the possible trial moves have been explored and the solution cannot further be improved in

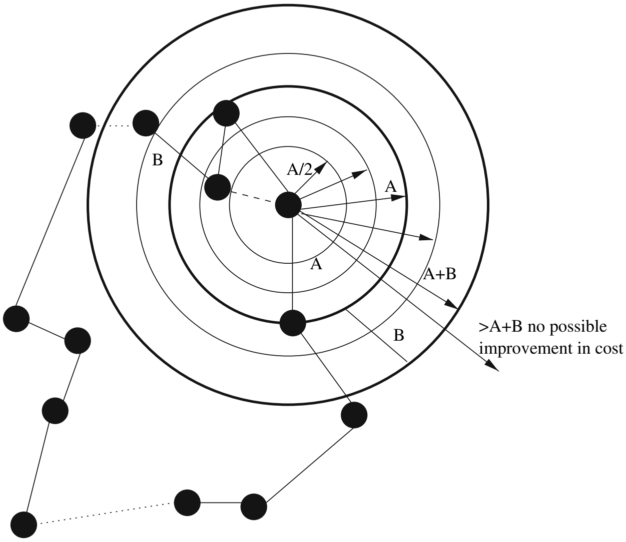

Fig. 2 Expanding neighborhood search strategy

this neighborhood. The local search strategies are changed if the current local search strategy finds a local optimum. Then the neighborhood is expanded by increasing the length of the radius of the CRLSM strategy s by a percentage θ (e.g. $\theta = 10\%$) and the algorithm continues. When the length of the radius of the CRLSM strategy is equal to A, the length continues to increase until the length becomes equal to $A + B$, where B is the length of the other candidate for deletion edge. If the length of the radius of the CRLSM strategy is equal to $A + B$, and the algorithm has reached the maximum number of iterations, then the algorithm terminates with the current solution. In Fig. 2, the expanding neighborhood search method is presented.

In the expanding neighborhood search strategy two local search algorithms are used. The **2-opt** and the **3-opt** algorithms which were introduced by Lin [21] for the classic TSP. In the first algorithm, the neighborhood function is defined as exchanging two edges of the current solution with two other edges, while in the second the neighborhood function is defined as exchanging three edges of the current solution with three other edges not in the current solution. When a new tour is achieved its expected length is calculated. If it is less than the expected length of the current tour then the new tour is considered as the current tour and the procedure continues from this one.

3 Computational results

The expanding neighborhood search–GRASP was implemented in Fortran 90 and was compiled using the Lahey f95 compiler on a Centrino Mobile Intel Pentium M750 at 1.86 GHz, running Suse Linux 9.1. Homogeneous probabilistic traveling salesman problem instances were generated starting from TSP instances and assigning to each customer a probability p of requiring a visit. The test instances were taken from the TSPLIB (http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95).

Table 1 Results of the expanding neighborhood–GRASP

Instance	Probabilities								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
eil51	130.30	196.90	242.63	279.11	311.04	339.90	366.44	388.49	407.92
kroA100	9,079.86	11,765.67	13,720.92	15,277.22	16,584.32	17,736.13	18,767.65	19,671.21	20,508.77
eil101	200.03	286.93	353.42	408.48	455.73	498.36	536.55	569.60	601.51
ch150	2,520.10	3,475.31	4,107.56	4,603.52	5,016.85	5,378.24	5,707.12	6,010.45	6,292.01
d198	7,525.03	9,438.94	10,695.67	11,690.01	12,538.45	13,275.78	13,979.83	14,617.66	15,225.26
pr439	50,402.09	66,628.49	76,266.54	83,370.69	88,914.76	93,830.37	97,905.62	101,535.63	104,735.27
p654	19,766.74	22,300.16	24,607.91	26,553.08	28,388.01	29,884.43	31,229.16	32,478.67	33,646.79
rat783	3,618.01	4,973.17	5,864.85	6,548.55	7,097.85	7,547.45	7,944.74	8,302.37	8,625.26
pr1002	111,959.65	147,476.25	173,113.73	192,062.84	207,916.81	223,088.96	233,874.34	244,851.42	254,819.95
fl1400	9,727.04	12,107.13	13,868.61	15,352.95	16,541.93	17,500.70	18,381.10	19,086.88	19,729.55

Table 2 Comparisons of the proposed algorithm with other metaheuristic approaches

Instance	GRASP		Tabu Search		ENS–GRASP	
	0.1	0.5	0.1	0.5	0.1	0.5
eil51	130.95	315.34	130.82	313.50	130.30	311.04
kroA100	9,191.29	16,717.81	9,116.64	16,658.46	9,079.86	16,584.32
eil101	203.06	465.05	202.42	461.52	200.03	455.73
ch150	2,530.27	5,112.86	2,554.59	5,071.51	2,520.10	5,016.85
d198	7,564.90	12,702.50	7,525.03	12,606.23	7,525.03	12,538.45
pr439	50,898.31	91,955.72	50,848.29	90,463.31	50,402.09	88,914.76
p654	20,069.91	28,748.1953	20,034.61	28,510.05	19,766.74	28,388.01
rat783	3,737.69	7,299.24	3,705.31	7,123.76	3,618.01	7,097.85
pr1002	111,967.65	215,116.35	113,868.22	210,639.29	111,959.65	207,916.81
fl1400	9,960.55	17,092.57	9,767.43	16,570.99	9,727.04	16,541.93

The algorithm was tested on a set of ten Euclidean sample problems with sizes ranging from 51 to 1,400 nodes. For each PTSP instance tested, nine experiments were done varying the value of the customer probability p from 0.1 to 0.9 with a 0.1 interval. The length of RCL varies from 30 to 150. The number of iterations of the ENS–GRASP algorithm is set equal to 1,000. In Table 1, the results of the algorithm (the optimal expected length) are presented. Each instance is described by its TSPLIB name and size, e.g. in Table 1 the instance named kroA100 has size equal to 100 nodes.

A comparison with other metaheuristic approaches for the solution of the probabilistic traveling salesman problem is presented in Table 2. In this table, two other metaheuristic algorithms are used for the solution of the probabilistic traveling salesman problem. The first one is an implementation of GRASP algorithm without using the expanding neighborhood search strategy running for 1,000 iterations and using

Table 3 Comparisons of the proposed algorithm with other metaheuristic approaches from the literature

Method	kroA100	eil101	ch150	d198	rat783
	Probability = 0.5				
ENS–GRASP	**16,584.32**	**455.73**	**5,016.85**	**12,538.45**	**7,097.85**
GRASP	16,717.81	465.05	5,112.86	12,702.50	7,299.24
Tabu Search	16,658.46	461.52	5,071.51	12,606.23	7,123.76
pACS [4]	16,605.4	470.7	5,164.3	12,745.5	7,334.1
pACS-S [4]	16,793.61	476.03	5,318.64	13,109.46	7,423.20
pACS+1-shift [4]	17,723.7	500.6	5,467.1	13,539.7	7,785.5
pACS-S+1-shift-S [4]	16,679.3	460.7	5,051.3	12,613.3	7,261.4
pACS+1-shift-T [4]	16,839.5	491.7	5,447.5	13,261.4	7,427.4
pACS+1-shift-P [4]	16,681.61	465.96	5,099.03	13,502.58	7,346.53
TSP-ACO [9]	N/M (not mentioned)	467.441	N/M	N/M	N/M
Angle-ACO [9]	N/M	463.570	N/M	N/M	N/M
Depth-ACO [9]	N/M	460.563	N/M	N/M	N/M
HS/1-Shift [9]	N/M	463.368	N/M	N/M	N/M
	Probability = 0.1				
ENS–GRASP	9,079.86	200.03	2,520.10	**7,525.03**	3,618.01
GRASP	9,191.29	203.06	2,530.27	7,564.90	3,737.69
Tabu Search	9,116.64	202.42	2,554.59	**7,525.03**	3,705.31
pACS [4]	**9,039.4**	**199.7**	**2,493.6**	7,556.1	**3,368.9**
pACS-S [4]	9,098.05	201.68	2,533.73	7,584.43	3,472.58
pACS+1-shift [4]	11,715.1	283.6	3,418.3	9,312.1	4,619.2
pACS-S+1-shift-S [4]	9,161.6	236.1	2,814.2	8,028.3	3,514.5
pACS+1-shift-T [4]	9,229.9	209.8	2,577.8	7,748.7	3,488.3
pACS+1-shift-P [4]	9,084.80	201.61	2,540.74	7,650.94	3,440.87

the same size of the RCL as in the ENS–GRASP. The second is a classic Tabu Search [14,15] running for 1,000 iterations and with size of the Tabu List equal to 10. In the table, only the results with two different probabilities are presented. The results of the ENS–GRASP are better than the results of the two other metaheuristics in all instances. It can, also, be seen that the results of the Tabu Search metaheuristic are better than the results of the GRASP algorithm without the ENS strategy, thus the use of the ENS strategy is very significant in order to have improvement in the results. The results of the algorithm are, also, compared (Table 3) with the results of a number of implementations of Ant Colony Optimization metaheuristic taking from [4,9]. In these implementations, they are used the same instances as in this paper and, thus, comparisons of the results can be performed. We use for the comparisons five instances with two different probabilities. As it can be observed the proposed algorithm gives a new best solution in six out of the ten cases that have been used in the comparisons (the best solution is indicated with bold characters in Table 3). More precisely, the proposed algorithm gives better results in all instances when the probability is equal to 0.5 and in one instance when the probability is equal to 0.1. For the other four instances the pACS [4] algorithm gives the best solution and the proposed algorithm in three out of the four instances gives the second best solution.

4 Conclusions and future research

In this paper, the expanding neighborhood search–GRASP is presented in order to solve the probabilistic traveling salesman problem. This algorithm is a general algorithm that can be applied with remarkable results both to quality and computational efficiency to many combinatorial optimization and supply chain problems, such as the traveling salesman problem and the vehicle routing problem. The algorithm was tested in a number of instances from the TSPLIB. Homogeneous probabilistic traveling salesman problem instances were generated starting from TSP instances and assigning to each customer a probability p of requiring a visit. In order to show the efficiency of the algorithm the results of the ENS–GRASP were compared with the results of the classic GRASP and a Tabu Search metaheuristic. Also, a comparison was performed with the results of a number of implementation of the ant colony optimization algorithm and in six out of ten cases the proposed algorithm gave a new best solution. Future research will focus in the application of the algorithm in other stochastic problems like the Stochastic Vehicle Routing Problem and in the development of other metaheuristic approaches for the solution of the probabilistic traveling salesman problem and the stochastic vehicle routing problem.

References

1. Bertsimas, D. J.: Probabilistic Combinatorial Optimization Problems. Ph.D. thesis, MIT, Cambridge (1988)
2. Bertsimas, D., Howell, L.: Further results on the probabilistic traveling salesman problem. Eur. J. Oper. Res. **65**(1), 68–95 (1993)
3. Bertsimas, D., Chervi, P., Peterson, M.: Computational approaches to stochastic vehicle routing problems. Transp. Sci. **29**(4), 342–352 (1995)
4. Bianchi, L.: Ant Colony Optimization and Local Search for the Probabilistic Traveling Salesman Problem: A Case Study in Stochastic Combinatorial Optimization. Ph.D. Thesis, Universite Libre de Bruxelles, Belgium (2006)
5. Bianchi, L., Knowles, J., Bowler, N.: Local search for the probabilistic traveling salesman problem: correction to the 2-p-opt and 1-shift algorithms. Eur. J. Oper. Res. **162**(1), 206–219 (2005)
6. Bianchi, L., Gambardella, L.M., Dorigo, M.: An ant colony optimization approach to the probabilistic traveling salesman problem. In: Merelo Guervòs, J.J., Adamidis, P., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) Proceedings of the Seventh Parallel Problem Solving from Nature (PPSN VII), Lecture Notes in Computer Science, vol. 2439, pp. 883–892. Springer, Berlin (2002)
7. Bianchi, L., Gambardella, L.M., Dorigo, M.: Solving the homogeneous probabilistic traveling salesman problem by the ACO metaheuristic. In: Dorigo, M., DiCaro, G., Sampels, M. (eds.) Proceedings of Third International Workshop ANTS 2002, Lecture Notes in Computer Science, vol. 2463, pp. 176–187. Springer, Berlin (2002)
8. Bowler, N.E., Fink, T.M.A., Ball, R.C.: Characterization of the probabilistic traveling salesman problem. Phys. Rev. E 68 (2003)
9. Branke, J., Guntsch, M.: Solving the probabilistic TSP with ant colony optimization. J. Math. Model. Algorithms **3**(4), 403–425 (2004)
10. Campbell, A.M.: Aggregation for the probabilistic travelling salesman problem. Comput. Oper. Res. **33**, 2703–2724 (2006)

11. Choi, J., Lee, J.H., Realff, M.J.: An algorithmic framework for improving heuristic solutions: Part II. A new version of the stochastic traveling salesman problem. Comput. Chem. Eng. **28**(8), 1297–1307 (2004)
12. Feo, T.A., Resende, M.G.C.: Greedy randomized adaptive search procedure. J. Glob. Optim. **6**, 109–133 (1995)
13. Garfinkel, R., Nemhauser, G.: Integer Programming. Wiley, New York (1972)
14. Glover, F.: Tabu search I. ORSA J. Comput. **1**(3), 190–206 (1989)
15. Glover, F.: Tabu search II. ORSA J. Comput. **2**(1), 4–32 (1990)
16. Hansen, P., Mladenovic, N.: Variable neighborhood search: principles and applications. Eur. J. Oper. Res. **130**, 449–467 (2001)
17. Jaillet, P.: Probabilistic traveling salesman problems. PhD Thesis, MIT, Cambridge (1985)
18. Jaillet, P., Odoni, A.R.: The probabilistic vehicle routing problem. In: Golden, B.L., Assad, A.A. (eds.) Vehicle Routing: Methods and Studies. Elsevier, North Holland, Amsterdam, pp. 293–318 (1988)
19. Jaillet, P.: A priori solution of a traveling salesman problem in which a random subset of the customers are visited. Oper. Res. **36**(6), 929–936 (1988)
20. Laporte, G., Louveaux, E., Mercure, H.: A priori optimization of the probabilistic traveling salesman problem. Oper. Res. **42**, 543–549 (1994)
21. Lin, S.: Computer solutions of the traveling salesman problem. Bell Syst. Tech. J. **44**, 2245–2269 (1965)
22. Liu, Y.H.: A hybrid scatter search for the probabilistic traveling salesman problem. Comput. Oper. Res (available on line) (2007)
23. Liu, Y.-H., Jou, R.-C., Wang, C.-J.: Genetic algorithms for the probabilistic traveling salesman problem. In: Proceedings of the conference on E-logistics. Taoyuan, Taiwan, pp. 77–82 (2004)
24. Marinakis, Y.: Vehicle routing in distribution problems. Ph.D. Thesis, Technical University of Crete, Department of Production Engineering and Management, Chania, Greece (2005)
25. Marinakis, Y., Migdalas, A., Pardalos, P.M.: Expanding neighborhood GRASP for the traveling salesman problem. Comput. Optim. Appl. **32**, 231–257 (2005)
26. Powell, W.B., Jaillet, P., Odoni, A.: Stochastic and dynamic networks and routing. In: Ball, M.O., Magnanti, T.L., Momma, C.L., Nemhauser, G.L. (eds.) Network Routing, Handbooks in Operations Research and Management Science, vol. 8, pp. 141–295. Elsevier, Amsterdam (1995)
27. Psaraftis, H.N.: Dynamic vehicle routing problems. In: Golden, B.L., Assad, A.A. (eds.) Vehicle Routing: Methods and Studies, pp. 223–248. Elsevier, North Holland, Amsterdam (1988)
28. Resende, M.G.C., Ribeiro, C.C.: Greedy randomized adaptive search procedures. In: Glover, F., Kochenberger, G.A. (eds.) Handbook of Metaheuristics, pp. 219–249. Kluwer, Boston (2003)
29. Rossi, F.A., Gavioli, I.: Aspects of heuristics methods in the probabilistic traveling salesman problem. In: Andreatta, G., Mason, F., Serafini, P. (eds.) Advanced School on Stochastics in Combinatorial Optimization, pp. 214–227. World Scientific Press, Singapore (1987)
30. Tang, H., Miller-Hooks, E.: Approximate procedures for the probabilistic traveling salesperson problem. Transp. Res. Rec. **1882**, 27–36 (2004)
31. Tarjan, R.: Data Structures and Network Algorithms. Society for Industrial and Applied Mathematics, Philadelphia (1983)