ISSN 1870-4069

Hybrid CW-GA Metaheuristic for the Traveling Salesman Problem

Irma Delia Rojas Cuevas, Santiago Omar Caballero Morales

Universidad Popular Autónoma del Estado de Puebla A.C., Puebla, Puebla, Mexico

irmadelia.rojas@upaep.edu.mx, santiagoomar.caballero@upaep.mx

Abstract. The Traveling Salesman Problem (TSP) is one of the most challenging problems in Logistics and Supply Chain Management. Its relevance in routing planning and distribution has significant impact on reduction of operative costs for all enterprises. However, due to its NP-hard complexity, it is difficult to obtain optimal solutions for TSP instances. This paper describes a hybrid approach based on Clarke and Wright (CW) and Genetic Algorithms (GA) to provide near optimal solutions for the TSP. Performance of this meta-heuristic was assessed by comparing it with other well-known methods such as CW, GA and Tabu-Search (TS). Results obtained from experiments with TSP instances corroborated the suitability of the hybrid approach for the TSP.

Keywords: traveling salesman problem, tabu search, metaheuristics, Clarke and Wright, genetic algorithms.

1 Introduction

The Traveling Salesman Problem (TSP) [1] is one of the most important and challenging problems in Logistics and Supply Chain Management. This is because operative costs associated to transportation and distribution are correlated to the efficiency of route planning, and TSP is focused on finding the route of minimum distance to cover a set of customer points.

However, solving the TSP is a computational task of NP-hard complexity. Thus, it is difficult to obtain exact or optimal solutions for large instances of the TSP (number of customer points higher than 100). This is the reason why the TSP is one of the combinational optimization problems that has attracted many researchers to propose and analyze metaheuristic algorithms to solve it in polynomial time. Among the algorithms used for this purpose, the following can be mentioned: Genetic algorithms (GA) [2], Tabu-Search (TS) [3], Clarke and Wright (CW) [4].

For GAs different strategies and implementations have been proposed. This has led to different reported performance for the TSP: average errors from best-known solutions between 1.56% and 7.64% for 14 TSP instances [5], 0.00% to 1.62% for 40 instances [6], 0.00% to 2.54% for 22 TSP instances [7], 0.00% to 2.46% for 29 instances

[8] and 0.00% to 0.61% for six TSP instances [9]. On the other hand, with TS, average errors from best known solutions of 0.00% to 6.00% have been reported [10]. However, in some cases, TS has provided better solutions than best-known solutions as reported in [11].

This work is focused on providing a hybrid metaheuristic to provide near-optimal solutions to different TSP instances. Based on work reported in the literature, the metaheuristic is aimed to provide solutions with an average error less than 7.64% [5]. The metaheuristic is integrated by the CW algorithm and a GA (CW-GA metaheuristic). This was performed in order to achieve sequential improvement on CW solutions by means of evolutionary operators. Comparison with the hybrid CW-TS approach led to confirm that GA can be a more suitable metaheuristic for hybridization with other methods for sequential improvement of TSP solutions.

The present work is structured as follows: in Section 2 the details of the CW, GA and TS algorithms are presented. Then, in Section 3 the assessments of the CW, GA, TS, CW-GA and CW-TS methods are presented and discussed. Finally, the conclusions are presented in Section 4.

2 Development of the Metaheuristics

In this section, the metaheuristics used for the hybrid approach are described. In Section 2.1 the CW algorithm is described while in Section 2.2 the TS algorithm is described. Then, in Section 2.3 the elements of the GA are described.

2.1 Clarke and Wright (CW)

Clarke and Wright is commonly used for Vehicle Routing Problems (VRP). In this case it is used for the TSP (VRP is also known as the multiple TSP or *m*TSP). The CW algorithm is described as follows:

- First the Euclidean distance between all customer points is computed:

$$c_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}, \tag{1}$$

  where (x_i, y_i) and (x_j, y_j) are the geographical locations of customer point i and j respectively.
- Second, the saving value between customer i and j is computed as:

$$s_{ij} = c_{Di} + c_{jD} - c_{ij}, \tag{2}$$

  where c_{Di} is the traveling distance between the Depot (start-end point for the TSP route of minimum distance) and customer i, c_{jD} is the traveling distance between customer j and the Depot, and c_{ij} is the traveling distance between customer i and j. Eq. (2) is not the original proposed by Clarke and Wright, it is the modified expression proposed by Bodin [12]. After computation, all savings values are stored in a *savings list* with its corresponding customer points.

- Third, the values in the *savings list* are sorted in decreasing order. Finally, the route is constructed by continuous merging of customer points *(i,j)* with the highest saving value. Thus, the merging procedure starts from the top of the *savings list*. Figure 1 presents an extension of the description of the CW algorithm.

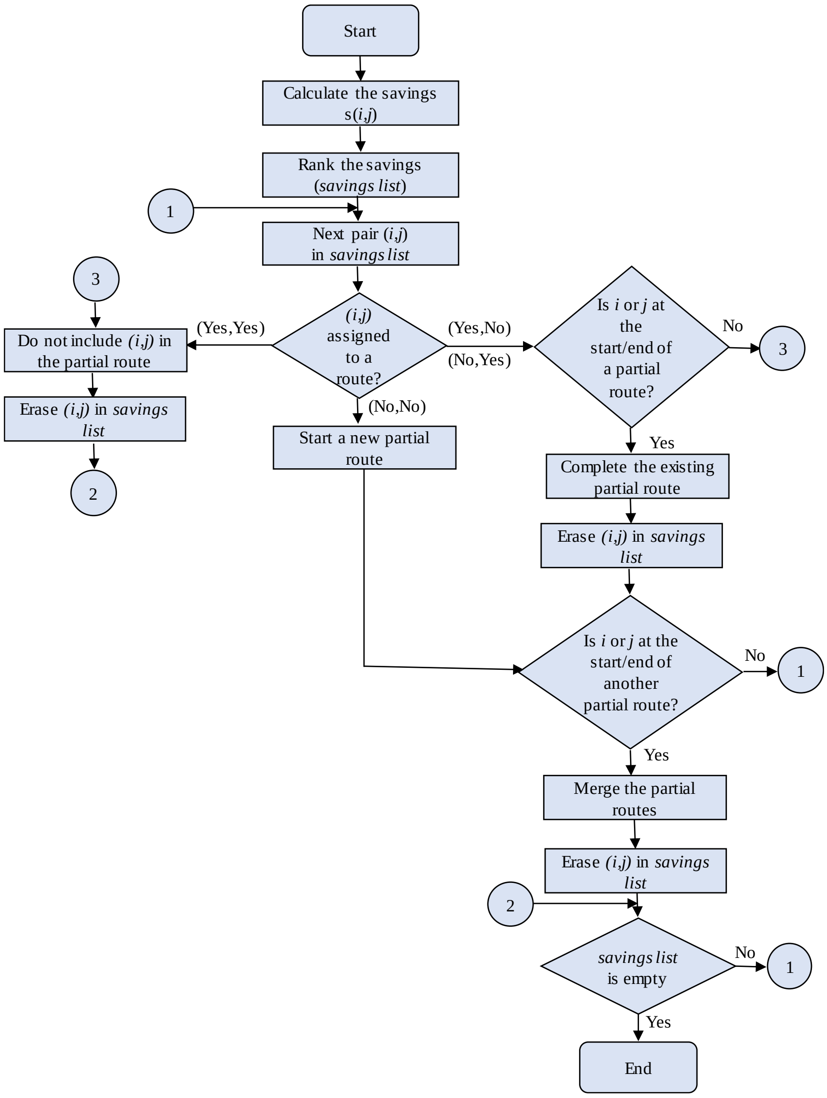

Fig. 1. Description of the Clarke and Wright (CW) algorithm.

2.2 Tabu Search (TS)

TS considers a neighborhood search procedure to iteratively move from one potential solution S_0 to an improved solution *sBest* in the neighborhood of S_0. In order to avoid revisiting previous solutions a *tabu list* is considered. This list keeps the last movements that led to a solution *hidden* from the search process during a number of iterations. The overall TS process is as follows [13]: a *tabu list* of size $N/4$ is considered for a 2-opt movement strategy with a *Stop Condition* of 1000 iterations. A generation of feasible solutions (neighborhood) is obtained with the 2-opt strategy and assessment for the *tabu list* is based on the minimum distance criteria. If the movement that led to the solution of minimum distance is not in the *tabu list* and this solution is better than the best solution previously found then the movement is applied, the best solution is updated, and the movement is added the *tabu list*. If the movement does not improve the quality of the solution, the movement is just added to the *tabu list*. If the movement is already in the *tabu list* then no change or update is performed. Figure 2 presents an extension of the description of the TS algorithm.

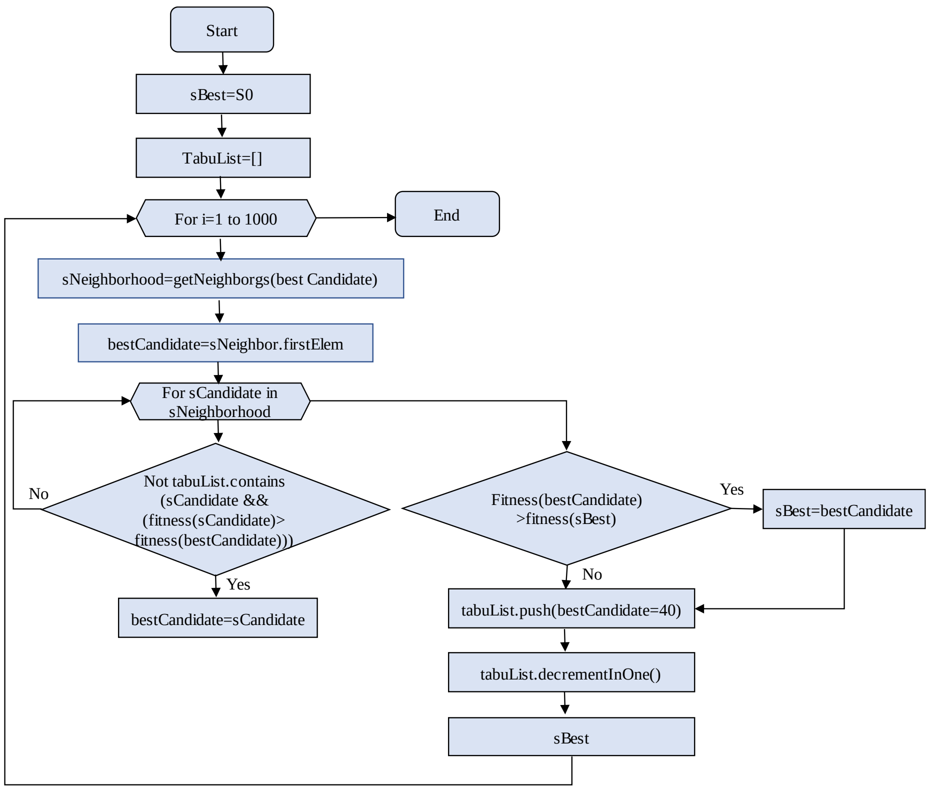

Fig. 2. Description of the Tabu-Search (TS) algorithm.

2.3 Genetic Algorithm (GA)

In a GA, a population of candidate solutions for an optimization problem is evolved to obtain better solutions. Each candidate solution has a set of properties which can be mutated or be used to exchange information with other solutions. In this case, the properties are the sequences of nodes in a tour or route of minimum distance.

The GA begins with an initial population of *A* individuals which can be randomly generated or be obtained by another method. Then, this population is evolved by producing *offspring* from selected individuals (*parents*). Potential parents were selected based on their *fitness* to solve the combinatorial problem. On each generation, the fitness of every individual in the population is evaluated; the fitness is the value of the objective function to be solved (in this case, the total distance of the TSP route). For this case, parents were randomly selected from the best 50% of individuals.

Then, offspring were obtained by means of the *reproduction operators* known as *crossover* and *mutation*. For this case, 0.80×*A* offspring were produced with OX crossover, 0.05×*A* were produced with CX crossover, and 0.15×*A* were produced with 2-opt, 1-opt and shift mutation. Figure 3 presents the extended description of the GA.

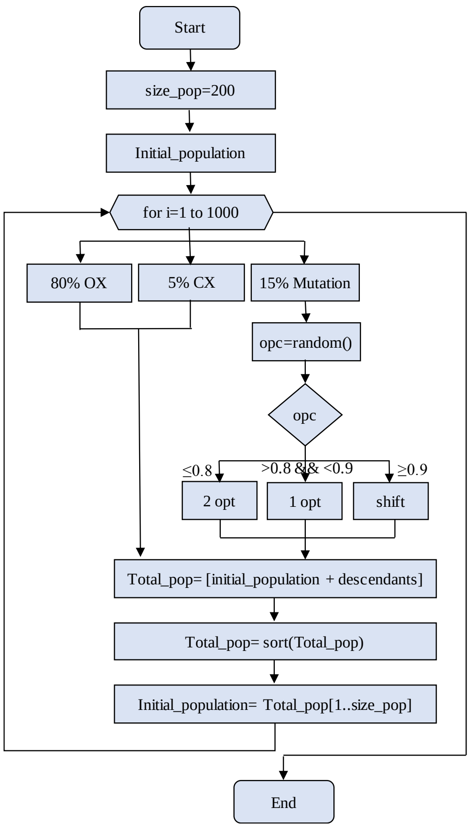

Fig. 3. Description of the Genetic Algorithm (GA).

- **2-opt:** Figure 4 presents the reproduction mechanism of this operator. In the first step (a) two points are random selected. Then, in the second step (b) the points are reconnected and the cost (total distance) of the route is computed. If the cost of this route is better than the previous cost, the two points remain in the new position, otherwise, they return to their original position. With the **1-opt** operator, a single point is selected and it is moved to a randomly selected location.

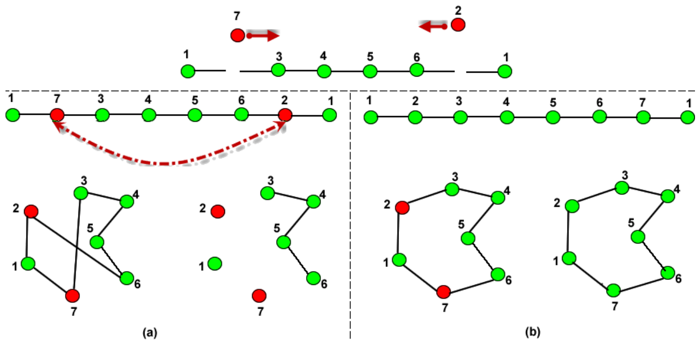

Fig. 4. Sample tour with seven locations: (2, 7) are swapped.

- **Shift**: Figure 5 presents the two steps of this process. In the first step (a) two points are randomly selected. Then, in the second step (b) the order of all points between the two selected points (including these points) is reversed. If the cost of the new route is better than the previous cost, the shifted points remain in their new positions, otherwise, they are returned to their original position.

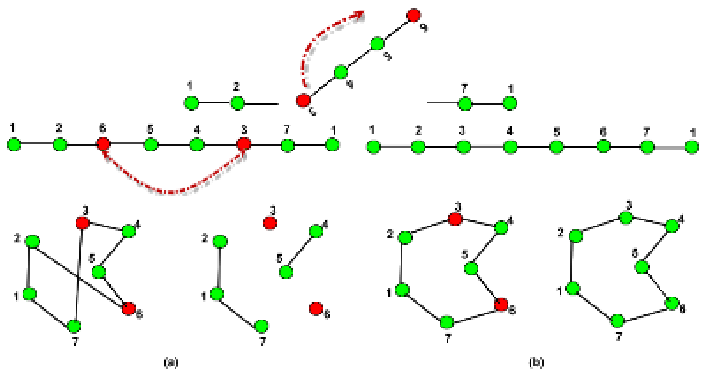

Fig. 5. Sample tour with seven locations: (2,7) are shifted.

- **Cycle Crossover (CX) and Order Crossover (OX):** With these operators, points between parent solutions can be exchanged, leading to offspring that contain route information from different solutions.

3 Assessment

A set of 30 symmetric TSP instances were considered to assess the approach [14]. For this case, GA and TS were initialized with randomly generated solutions. As shown in

Table 1, the constructive method CW has the lowest average error (8.39%) from well-known solutions. GA follows with 47.50% and TS has the highest average error with 106.01%. Also, as presented in Figure 6, GA has better and faster convergence than TS. However, none of them reach the optimal solution, or are close to the performance of CW.

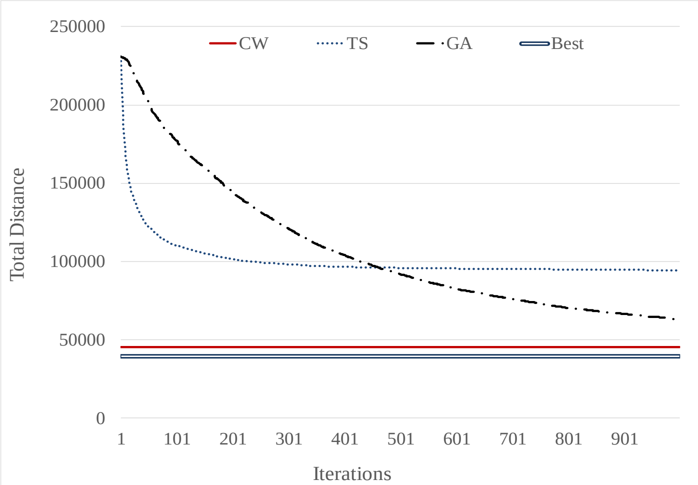

Fig. 6. Convergence of CW, TS, and GA algorithms.

In order to assess the hybrid approach, the initial solutions for TS and GA were obtained with the CW algorithm. This led to the hybrid metaheuristics CW-TS and CW-GA. As presented in Table 1, the CW-GA outperforms CW, TS, GA and the CW-TS method by obtaining an average error of 5.57%. In contrast, CW-TS achieved an average error of 6.61%. Convergence of CW-GA and CW-TS is presented in Figure 7.

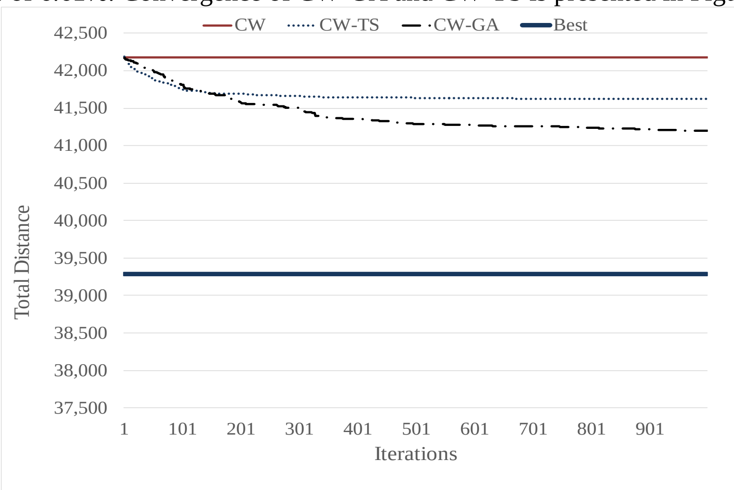

Fig. 7. Convergence of CW, CW-TS and CW-GA algorithms.

Table 1. Comparative results of CW, GA, TS, CW-GA and CW-TS.

				GA		TS		CW-GA		CW-TS	
	Best	CW	%	Value	%	Value	%	Value	%	Value	%
Eil51	426	437	2.62	437	2.62	437	2.62	437.15	2.62	437.15	2.62
Berlin52	7,542	8,291	9.93	7,748	2.73	8,117	7.63	7747.9	2.73	8117.4	7.63
Eil76	538	574	6.73	595	10.52	817	51.82	574.21	6.73	570.58	6.06
pr76	108,159	113,930	5.34	115,340	6.64	171,630	58.68	111570	3.15	112590	4.10
kroA100	21,282	23,049	8.30	24,852	16.77	38,759	82.12	21784	2.36	22523	5.83
kroB100	22,141	24,392	10.17	24,747	11.77	41,775	88.68	23944	8.14	23958	8.21
kroC100	20,749	22,516	8.52	23,233	11.97	38,223	84.22	21997	6.01	22084	6.43
kroD100	21,294	22,902	7.55	25,477	19.64	37,072	74.10	22746	6.82	22720	6.70
kroE100	22,068	24,496	11.00	25,516	15.62	40,990	85.74	23619	7.03	23518	6.57
Eil101	629	691	9.85	679	7.87	683	8.64	678.5	7.87	683.37	8.64
Lin105	14,379	15,763	9.63	14,878	3.47	15,092	4.96	14878	3.47	15092	4.96
pr107	44,303	49,157	10.96	50,245	13.41	119,580	169.91	45872	3.54	47814	7.92
pr124	59,030	63,978	8.38	73,611	24.70	143,790	143.59	60389	2.30	62938	6.62
Bier127	118,282	124,030	4.86	121,530	2.75	122,830	3.85	121530	2.75	122830	3.85
Ch130	6,110	6,817	11.58	6,623	8.40	6,696	9.59	6623.1	8.40	6696	9.59
pr136	96,772	106,200	9.74	132,010	36.41	165,780	71.31	106000	9.54	105880	9.41
pr144	58,537	63,447	8.39	86,919	48.49	177,080	202.51	62085	6.06	62920	7.49
Ch150	6,528	6,972	6.81	6,927	6.11	6,931	6.17	6926.9	6.11	6930.9	6.17
kroA150	26,524	28,526	7.55	38,486	45.10	49,763	87.61	27879	5.11	27913	5.24
kroB150	26,130	27,833	6.52	36,831	40.95	52,699	101.68	27115	3.77	27078	3.63
pr152	73,682	78,221	6.16	101,870	38.26	270,850	267.59	77581	5.29	78068	5.95
D198	15,780	17,704	12.19	17,086	8.28	17,312	9.71	17086	8.28	17312	9.71
kroA200	29,368	32,115	9.35	31,377	6.84	31,641	7.74	31377	6.84	31641	7.74
kroB200	29,437	33,059	12.30	52,559	78.55	68,518	132.76	31724	7.77	32367	9.95
pr226	80,369	84,902	5.64	164,690	104.92	377,510	369.72	129600	2.33	129600	2.33
pr264	49,135	53,087	8.04	124,070	152.51	213,500	334.52	82180	2.25	84317	4.91
Ts225	126,643	129,600	2.33	315,920	149.46	338,180	167.03	52459	6.77	52658	7.17
A280	2,579	2,851	10.56	6,951	169.53	6,643	157.60	2798.7	8.52	2811	9.00
pr299	48,191	54,178	12.42	134,020	178.10	145,030	200.95	52160	8.24	52274	8.47
lin318	42,029	45,484	8.22	127,190	202.62	120,730	187.25	44676	6.30	44336	5.49
Average			**8.39**		**47.50**		**106.01**		**5.57**		**6.61**

4 Conclusions

Performance of individual metaheuristics was variable: GA obtained an average error throughout 30 TSP instances of 47.50%. Meanwhile, TS obtained an average error of 106.01% from best known results. However, if CW is considered as the generator of initial solutions, the hybrid CW-GA metaheuristic is able to obtain an average error of 5.57%. In comparison, the hybrid CW-TS obtained an average error of 6.61%. CW obtained an average error of 8.39%.

Regarding convergence, initially TS has faster convergence than GA. However, after 400 iterations, the convergence of GA continues, achieving better results than TS.

Individual metaheuristics can provide very suitable solutions for routing problems as the TSP. However, performance may vary significantly from one metaheuristic to other. Better performance can be achieved if integration between metaheuristics is performed. In this case, GA could improve the performance of the constructive CW algorithm, and its evolutionary features were more competitive than the features of TS.

Thus, with an average error of 5.57%, the hybrid CW-GA may provide better solutions than individual metaheuristics.

Acknowledgements. The main author would like to thank Tecnológico Nacional de México/Instituto Tecnológico de Puebla and Consejo Nacional de Ciencia y Tecnología (CONACYT) for the scholarship to pursue a Ph.D. degree.

References

1. Glover F., McMillan C.: The general employee scheduling problem: An integration of MS and AI. Computers & Operations Research, 13(5), 563–573. doi: /10.1016/0305-0548(86)90050-X (1986)
2. Holland, J. H., Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence. Ann Arbor, MI: University of Michigan Press (1975)
3. Glover, F.: Future paths for integer programming and links to artificial intelligence. Computers & Operations Research, 13(5), 533–549.doi: 10.1016/0305-0548(86)90048-1 (1986)
4. Clarke, G., & Wright, J. W. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research, 12(4), 568–581 (1964)
5. Albayrak, M., Allahverdi, N.: Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms. Expert Systems with Applications, 3(38), 1313–1320 (2011)
6. Antosiewicz, M., Grzegorz, K., & Kamiński, B.: Choice of best possible metaheuristic algorithm for the travelling salesman problem with limited computational time: quality, uncertainty and speed. Journal of Theoretical and Applied Computer Science, 7(1), 46–55 (2013)
7. Doyuran, T., Çatay, B.: A robust enhancement to the Clarke-Wright savings algorithm. The Journal of the Operational Research Society, 62(1), 223–231 (2011)
8. He, Y., Qiu, Y., Liu, G., Lei, K.: A parallel adaptive tabu search approach for traveling salesman problems. In: Natural Language Processing and Knowledge Engineering, IEEE NLP-KE'05, Proceedings of IEEE International Conference on, 796–801 (2005)
9. Ma, J., Yang, T., Hou, Z., Tan, M., Liu, D.: Neurodynamic programming: A case study of the traveling salesman problem. Neural Computing and Applications, 17(4), 347–355 (2008)
10. Marinakis, Y., Magdalene, M.: A hybrid multi-swarm particle swarm optimization algorithm for the probabilistic traveling salesman problem. Computers & Operations Research, 37(3), 432–442 (2010)
11. Marinakis, Y., Athanasios, M., Panos, M. P.: A hybrid genetic—GRASP algorithm using Lagrangean relaxation for the traveling salesman problem. Journal of Combinatorial Optimization, 10(4), 311–326 (2008)
12. Yan, X., Liu, H. M., Yan, J., Wu, Q. H.: A fast evolutionary algorithm for traveling salesman problem. In: Natural Computation, Third International Conference on IEEE, 4, 85–90 (2007)
13. Tsubakitani, S., Evans J. R.: Optimizing Tabu List size for the Traveling Salesman Problem. Computers & Operations Research, 25(2), 91–97 (1998)
14. Yang, N., Ping, L., Baisha, M.: An angle-based crossover tabu search for the traveling salesman problem. In: Natural Computation ICNC 2007, Third International Conference on, IEEE, 4, 512–516 (2007)