ISSN 1870-4069

The Mexican *Magical Towns* Traveling Salesman Problem: Preliminary Results

Sandra J. Gutiérrez, Nareli Cruz-Cortés, Hiram Calvo,
Marco A. Moreno-Armendáriz

Instituto Politécnico Nacional, Centro de Investigación en Computación,
CDMX, Mexico

B160499@sagitario.cic.ipn.mx, nareli@cic.ipn.mx, hcalvo@cic.ipn.mx,
mam_armendariz@cic.ipn.mx

Abstract. The Traveling Salesman Problem (TSP) is a classical NP-hard combinatorial problem that has been intensively studied through several decades. A large amount of literature is dedicated to this problem. TSP can be directly applied to some real life problems, and to be generalized to some other set of important combinatorial problems. Nowadays, a large collection of TSP instances can be found such as TSPLIB and National TSP webpage, among others. In this paper we experiment with a different TSP instance, that is, a set of 111 Mexican towns (called *Magical Towns*), by applying five different algorithms: (1) Ant Colony Optimization, (2) Particle Swarm Optimization, (3) Greedy Metaheuristic (4) Lin-Kernighan and (5) Hybrid Metaheuristic. The best results were obtained by the hybrid metaheuristic, with a final cost function of 120.65.

Keywords: ant colony optimization, Lin-Kernighan, *MAX-MIN* ant system, traveling salesman problem, *magical towns.*

1 Introduction

The Traveling Salesmen Problem (TSP) is one of the problems in mathematics and computer science which is easy to understand and needs much effort to solve. The idea of the problem is to find the shortest path for a salesman starting from a given city, visiting n cities only once and finally arriving at the city of origin.

Much of the work on the TSP is motivated by its use as a platform for analyzing the performance of general methods that can be applied to a wide range of discrete optimization problems. This is not to say, however, that the TSP does not find applications in many fields. Indeed, the numerous direct applications of the TSP bring life to the research area and help to direct future work.

pp. 159–168; rec. 2017-09-25; acc. 2017-10-23

A large collection of TSP instances can be found along the specialized literature, for instance TSPLIB[1], National TSP webpage[2], art constructed by solving instances of the TSP[3], among many others. However, there is almost no published work related to Mexican instances.

On the other hand, a Mexican Magical Town certificate is awarded to places that preserve and display tradition, beauty and culture through symbolic attributes, legends and history. The *Magical Towns Program* is a proposal made by Mexico's Ministry of Tourism launched in 2001 and, by 2017, a total of 111 towns and villages in all 31 states have been awarded the title *Magical Town*. The main objective of this program is to invite to visit this kind of towns that provide a magical experience with a typical illustration of an authentic Mexican village, natural beauty landscapes, folklore and historical sites, in order to increase profits from travel and tourism.

In this paper we apply some heuristic algorithms in a specific TSP problem, in this case Mexican *Magical Towns*. Specifically, a comparison among the following algorithms is presented: two bio-inspired algorithms: Particle Swarm Optimization [1], *Max-Min* Ant System [2], besides the best-known algorithm for TSP proposed by Lin & Kerninghan [3], a simple greedy strategy and an hybrid algorithm. Our comparative study shows that the algorithm with the best results is the hybrid algorithm.

The rest of the paper will proceed as follows: the following section establishes the TSP solved with different artificial intelligence (AI) techniques and describes some routing planning problems in Mexico. Some optimization metaheuristics used in this paper are described in Section 3. The methodology for this investigation is presented in Section 4. Computational results and discussion are presented in Section 5. Finally, future work and conclusions are presented in Section 6.

2 Traveling Salesman Problem

The most used representation of TSP is an edge-weighted graph $G = (V, E)$ with V being set of $n = |V|$ nodes or vertices representing cities and $E \subseteq V \times V$ being set of directed edges or arcs. Each arc $(i, j) \in E$ is assigned with a value of length d_{ij} which is distance between cities i and j with $i, j \in V$. TSP can be either asymmetric or symmetric in nature. In case of asymmetric TSP, distance between pair of nodes i, j is dependent on direction of traversing edge or arc i.e. there is at least one arc(i, j) for which $d_{ij} \neq d_{ji}$. In contrast, in symmetric TSP the relation $d_{ij} = d_{ji}$ holds for all arcs in E. The goal in TSP is thus to find minimum length Hamiltonian Circuit of graph, where Hamiltonian Circuit is a closed path visiting each of n nodes of G exactly once. Thus, an optimal solution to TSP is permutation π of node index $1, \ldots, n$ such that length $f(\pi)$

[1] https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/

[2] http://www.math.uwaterloo.ca/tsp/world/countries.html

[3] http://www2.oberlin.edu/math/faculty/bosch/tspart-page.html

is minimal, where $f(\pi)$ is the objective or cost function, given by equation 1 [4]:

$$f(\pi) = \sum_{i=1}^{n-1} d_{\pi(i)\pi(i+1)} + d_{\pi(n)\pi(1)}. \tag{1}$$

Besides the minimization of the total travel distance, other objectives could be considered too, for instance, minimization of travel time, minimization of money (using free ways), etc.

TSP is a classical NP-hard combinatorial problem that has been intensively studied among several decades using heuristics like iterated-local search, branch and bound and graph coloring. Even evolutionary algorithms, ant colony optimization (ACO) algorithms and hybridization of techniques have been used [5].

2.1 Mexican TSP

To our knowledge, there have been only few studies about routing optimization in Mexico. In [6] and [7] authors presents a novel method using ant colonies for optimizing patrolling routes for personnel working in public security and the case study is municipality of Cuautitlán Izcalli in Mexico. Also in [8] authors design a routing planning software with the aim of distance minimization and travel cost for State of Puebla using dynamic programming and Floyd algorithm. Finally, in [9] authors use minimum spanning tree to collect trash across Atizapán de Zaragoza, México, while in [10], they use nearest neighborhoods to optimize the collect of platforms across the country. These works are related to our paper, but there are not about Mexican TSP instances.

3 Optimization Heuristics and Metaheuristics

3.1 *MAX* - *MIN* Ant System (*MM*AS)

A main ant colony optimization algorithm (ACO) is Ant System (AS), a bio-inspired metaheuristic proposed by M. Dorigo [2]. In AS the real ant behavior is used as methaphor to get the best suitable path by artificial agents, its working is based on a chemical substance called *pheromone* which is released by the ants on their path, it helps ants to locate each other in order to get location of the nearest source of food. The higher the pheromone level, the greater the number of ants following that path [11].

Each k ant generates a complete tour by choosing the cities according to a probabilistic state transition rule given by equation 2, if $c_{ij} \in \mathbf{N}(s^P)$, otherwise $p_{ij}^k = 0$ [2]:

$$p_{ij}^k = \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{c_{il} \in \mathbf{N}(s^P)} \tau_{il}^\alpha \cdot \eta_{il}^\beta}, \tag{2}$$

where τ_{ij} is the pheromone associated with cities i and j, $\eta_{ij} = 1/d_{ij}$, d_{ij} is the distance between cities i and j, $\mathbf{N}(s^P)$ is the set of cities that remain to be visited, edges (i, l) where l is a city not yet visited. The parameters α and β control the influence between pheromone and heuristic information η_{ij}.

Ants prefer to move to cities which are connected by short edges with a high amount of pheromone. Once all ants have completed their tours the following updating rule is applied: a fraction of the pheromone evaporates on all edges (edges that are not refreshed become less desirable), and then each ant deposits an amount of pheromone on edges which belong to its tour in proportion to how short its tour was.

In [12] authors include *MAX - MIN* AS, it is an improved version in which only the best ant updates the pheromone trails and the value of the pheromone is bound. The pheromone update rule is shown in equation 3:

$$\tau_{ij} \leftarrow [(1-\rho) \cdot \tau_{ij} + \Delta\tau_{ij}^{best}]_{\tau_{min}}^{\tau_{max}}, \tag{3}$$

where ρ is the evaporation rate, $\Delta\tau_{ij}^{best}$ is the quantity of pheromone laid on edge (i, j) by the best ant, τ_{min} and τ_{max} are the upper and lower bounds, respectively, which are typically obtained empirically and tuned on the specific problem considered. If (i, j) belongs to the best tour $\Delta\tau_{ij}^{best}$ is equal to $1/L_{best}$, otherwise it is equal to zero, and L_{best} is the length of the tour of the best ant. There have been many works related to solving TSP using *MM* AS presented in [13] and [14].

3.2 Particle Swarm Optimization for TSP

The particle swarm optimization (PSO) is a popular metaheuristic based on social behaviour of flock of birds and school of fish; it was originally presented by Kennedy and Eberhart [15]. In PSO, each particle represents a potential solution with an initial velocity and moves to a new position. At each iteration, every particle calculates its velocity and updates its position according to equations 4 and 5, respectively [1]:

$$V_i^{(t)} = wV_i^{(t-1)} \otimes c_1r_1(P_i - X_i^{(t-1)}) \otimes c_2r_2(G - X_i^{(t-1)}), \tag{4}$$

$$X_i^{(t)} = X_i^{(t-1)} + V_i^{(t)}. \tag{5}$$

It is very important to mention that original PSO was only for continuous search spaces and because TSP is discrete, combinatorial problem, some modifications must be made. To solve TSP with PSO, each particle represents a complete tour as a feasible solution and velocity is a measure to update the tour for better solution. Many prominent PSO based methods use Swap Sequence (SS) for velocity. A SS is a collection of several Swap Operators (SOs) and each one indicates two positions in a tour, those might be swapped. All SOs of a SS are applied on a particle's tour maintaining order and hence implications of the SS transforms the TSP tour into a new one.

In equation 4, c_1 and c_2 are learning factors, r_1 and r_2 are random values between 0 and 1, P_i is the previous best tour, G is the best tour particles ever encountered, $P_i - X_i^{(t-1)}$ means to obtain SOs needed to make $X_i^{(t-1)}$ sequence equal to P_i sequence, which is the same for $G - X_i^{(t-1)}$, $c_1 r_1 (P_i - X_i^{(t-1)})$ means all SOs for $(P_i - X_i^{(t-1)})$ should be maintained with the probability of $c_1 r_1$, which is the same for $c_2 r_2 (G - X_i^{(t-1)})$, and w is the scaling factor of influence of the previous velocity in present velocity. Each particle moves to a new tour solution $X_i^{(t)}$ applying all SS on its previous solution $X_i^{(t-1)}$ using equation 5. A SS is the union of SOs, $SS = (SO_1, SO_2, \ldots, SO_n)$.

The operator $\otimes$ defines the merging operation. If $SS_1 = SO(1,2), SO(5,2)$ and $SS_2 = SO(5,3), SO(4,1)$, the new swap sequence is:

$$SS(new) = SS_1 \otimes SS_2 = SO(1,2), SO(5,2), SO(5,3), SO(4,1).$$

Finally if the new solution $X_i^{(t)}$ is superior to P_i, update P_i, and do the same for G. Some relevant works that used this metaheuristic were presented in [16] and [17].

3.3 Greedy Algorithms

A greedy algorithm is a mathematical process that follows the problem solving heuristic of making the locally optimal choice at each stage, deciding which next step will provide the most obvious benefit with the aim of finding a global optimum, in this case chooses the nearest city no matter where it starts, but availability of nearest cities it is important [18]. The advantage of using a greedy algorithm is that solutions to smaller instances of the problem can be straightforward and easy to understand. The disadvantage is that it is entirely possible that the most optimal short-term solutions may lead to the worst possible long-term outcome [19].

3.4 Lin-Kernighan

The Lin-Kernighan (LK) algorithm belongs to the class called *local search algorithms*. It was first presented by Lin and Kernighan in [20]. This algorithm is highly effective for generating optimal solutions for the symmetric TSP [21,22,23]. However, the design and implementation of an algorithm based on this heuristic is not trivial. Basically, it is a generalization of the k-opt local search. The k-opt neighborhood includes all the TSP tours that can be obtained by removing k edges from the original tour T and adding k edges such that the resulting tour is feasible. Lin-Kernighan explores only parts of the k-opt neighborhood that seem to be most promising: it removes a edge from a given tour, then the resulting path is rearranged looking for a minimization of its cost. To make the tour feasible again, only one edge must be added to close up the tour. Then, it is possible that the final aggregated edge does not minimize the total cost because this was not considered during the minimization process.

3.5 Hybrid Metaheuristics

Research in metaheuristics for combinatorial optimization problems has lately experienced a remarkable shift towards the hybridization of metaheuristics with other techniques for optimization. Simultaneously, the focus of research has changed from being rather algorithm-oriented to being more problem-oriented. Nowadays the focus is on solving the problem at hand in the best way possible, rather than promoting a certain metaheuristic. This new approach is full of a multitude of powerful hybrid algorithms that were obtained by combining components from several different optimization techniques [24].

4 Methodology

The algorithms to solve Mexican *Magical Towns* TSP were *MM*AS, PSO, LK, greedy and hybrid algorithms. All algorithms were implemented in Matlab®.

The database of *magical towns* coordinates[4] was incomplete, there were missing towns[5], so we manually completed it by searching in Google Maps the coordinates of the missing towns[6]. At the end, the list contains 111 coordinates (latitude and longitude) of *magical towns.*

4.1 Experimental Setup

Experiments were run in a Personal Computer with Windows 10 Pro 64-bits, with processor i7-4790@3.6 GHz and 8 GB RAM. The number of experiments for all algorithms was 30. Parameters for the AS and PSO algorithms are shown in Table 1 and Table 2, respectively, according to the most used parameters in the state of the art.

Table 1. Algorithm parameters for AS.

Parameter	m	ρ	α	β	Iterations
Value	111	0.1	1	2	100000

[4] http://www.gpstec.mx/coordenadas-gps-de-los-pueblos-magicos-de-mexico-2014/

[5] http://hellodf.com/la-lista-definitiva-de-todos-los-pueblos-magicos-que-hay-que-conocer-en-mexico/

[6] Available at (1)http://iarp.cic.ipn.mx/∼magicaltowns and (2)https://www.dropbox.com/sh/dtqodj41x29tmxz/AABQPfo3xHD8M4n19wKkF-ha?dl=0

Table 2. Algorithm parameters for PSO.

Parameter	Particles	Neighborhoods	Velocity	Iterations	c_1	c_2	w
Value	333	5	56	500	2	2	1

In Table 2, the velocity is the number of swap operators (SOs), i.e. the length of swap sequence (SS), in this case 56 random values between 1 and 111, which represent the indexes to change in the corresponding tour.

5 Results and Discussion

In this section we present our results of applying five techniques to solve TSPs using optimization metaheuristics. Table 3 shows the average performance of 30 experiments of *MAX-MIN* AS, PSO, Greedy, LK and an hybrid (PSO and LK) algorithms.

Table 3. Cost function values of algorithms.

Algorithm	Best	Worst	Average	Time [s]
*MM*AS	134.14	228.68	136.81	37.27
PSO	129.53	138.70	134.87	28.04
Greedy	140.83	168.06	154.39	0.32
LK	121.81	693.22	138.67	0.31
PSO and LK	**120.65**	**129.53**	**124.21**	0.28

It is interesting to note that distance between best and worst tour in AS is more significant than PSO, it shows that PSO has a faster convergence behavior than AS, maybe because ant system has an explorative tendency with the parameters selected and in PSO particles for a swarm in which, at the end, all have similar solutions. In Figure 1(a) the obtained tour using *MM*AS algorithm is shown.

Moreover, in PSO the initial velocity, i.e. the length of SS, influences the population diversity, values between 10 and 111 were tested along the experiments, the higher the value of population diversity, the more influence has the particle velocity. In Figure 1(b) the obtained tour using PSO algorithm is shown.

Furthermore in Figure 1(c) and 1(d) we shown the obtained tour using the greedy and LK ($k = 2$) algorithms, respectively.

The optimal tour was found by applying an hybrid heuristic with PSO and LK algorithms, with a final cost function of 120.65[7]. In Figure 2 the obtained tour using hybrid heuristic is shown.

[7] Available at (1) http://iarp.cic.ipn.mx/∼magicaltowns, and (2) https://www.dropbox.com/sh/dtqodj41x29tmxz/AABQPfo3xHD8M4n19wKkF-ha?dl=0

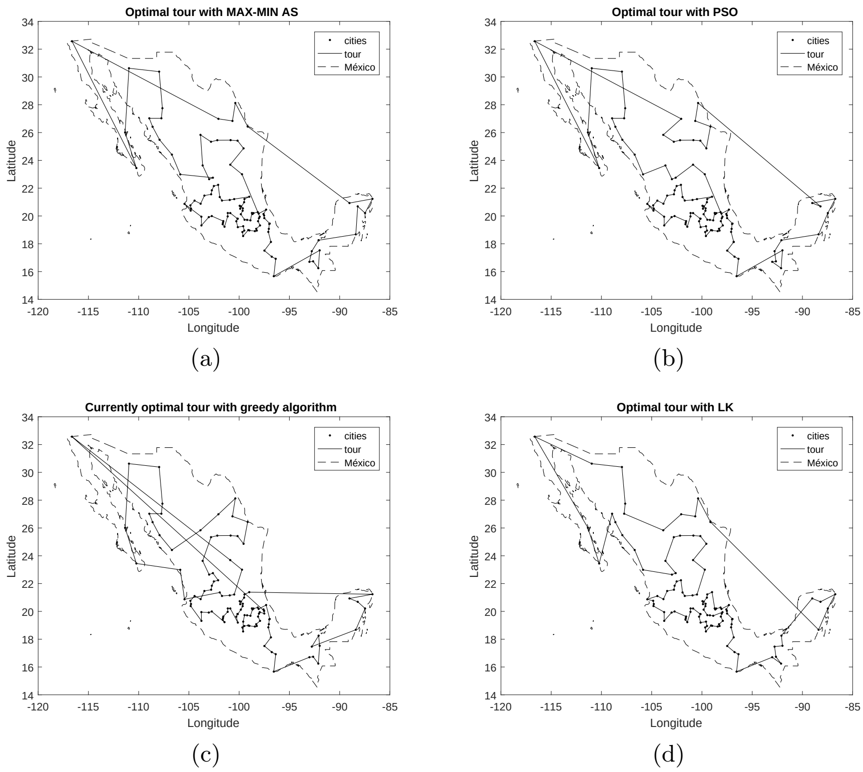

Fig. 1. Best tours using metaheuristics: (a) *MM*AS, (b) PSO, (c) Greedy, (d) LK.

6 Future Work and Conclusions

The experimental results show that the Mexican *Magical Towns* Traveling Salesman Problem could be solved applying different optimization metaheuristics like *MM* AS, PSO, greedy, LK and an hybrid technique.

However, using two metaheuristic results in a powerful combinations that finds the optimal tour for the 111 Mexican *Magical Towns*, with a final cost function value of 120.65. We think this results came from combining explorations and explotation from PSO algorithm and then add LK algorithm which performs locally search and tries to improve the best solution from PSO algorithm.

As future work, the *MM*AS, PSO, LK and greedy algorithms can be analyzed for future enhancement such that new research could be focused to investigate which parameters are the best according to the Mexican *Magical Towns* TSP. A series of hybrid AI algorithms still needs to be applied to the dataset, like genetic algorithm, artificial bee colony, simulated annealing, etc.

On the other hand, the dataset only includes the euclidean distance between cities and it is important to take into consideration the real distance according

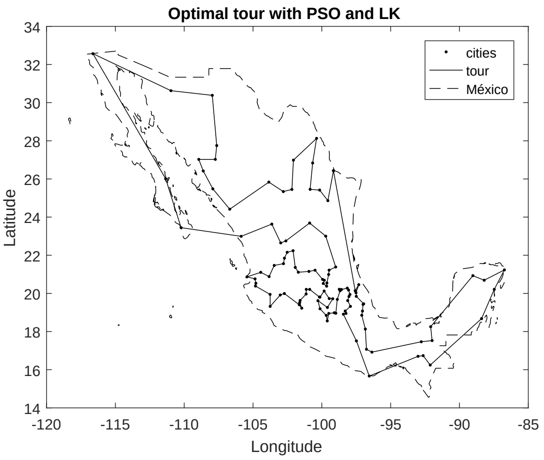

Fig. 2. Best tour using an hybrid metaheuristic.

to routes marked in a map. Nowadays, it is even possible to known the different routes, the traffic and estimated time through the different apps that use a GPS.

Acknowledgments. Work done under support Government of Mexico (Instituto Politécnico Nacional, SNI, SIP-IPN, COFAA-IPN, BEIFI-IPN and CONACyT) for providing necessary support. The authors would like to thank the reviewers for the useful comments, their time, effort, and engagement to making this manuscript better.

References

1. Akhand, M. A. H., Akter, S., Rashid, M. A. and Yaakob, S. B.: Velocity Tentative PSO: An Optimal Velocity Implementation based Particle Swarm Optimization to Solve Traveling Salesman Problem. International Journal of Computer Science 42(3):221–232 (2015)
2. Dorigo M., Gambardella L.C.: Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Transactions on Evolutionary Computation, 1:1–24 (1997)
3. Karapetyana, D., Gutina, G.: Lin-Kernighan Heuristic Adaptations for the Generalized Traveling Salesman Problem. European Journal of Operational Research 208(3), 221–232 (2011)
4. Chahuan, C., Gupta, R., Pathak, K.: Survey of Methods of Solving TSP along with its Implementation using Dynamic Programming Approach. International Journal of Computer Applications, 52(4):12–19 (2012)
5. Blum, C., Puchinger J., Raidl, G.R., Roli A.: Hybrid metaheuristics in combinatorial optimization: A survey. Applied Soft Computing 11:4135–4151 (2011)
6. Calvo, H., Godoy-Calderon S., Moreno-Armendáriz, M.A., Martínez-Hernández V.M.: Patrolling Routes Optimization Using Ant Colonies. In: 7th Mexican Conference on Pattern Recognition, pp.302–312, Mexico (2015)

7. Calvo, H., Godoy-Calderon S., Moreno-Armendáriz, M.A., Martínez-Hernández V.M.: Forecasting, clustering and patrolling criminal activities, J. Intelligent Data Analysis, 21(3):697–720 (2017)
8. Montiel R.: Un Planificador de Rutas Turísticas para el Estado de Puebla. In: 7th Congreso Internacional de Cómputo en Optimización y Software, pp.221–232, M.A. Cruz-Chávez Press, Estado de México (2011)
9. Reyes, R.: Diseño del programa de recolección de desechos sólidos domiciliarios para el municipio de Atizapán de Zaragoza como aplicación del problema del agente viajero. Undergraduate dissertation, Instituto Politécnico Nacional (2005)
10. Soto, C.: Modelo para optimizar la recolección de tarimas en una empresa arrendadora: Caso de estudio. Masters thesis, Instituto Politécnico Nacional (2008)
11. Raman, V., Singh, N.: Review of different heuristic algorithms for solving Traveling Salesman Problem. International Journal of Advanced Research in Computer Science, 8(5):423–425 (2017)
12. Stutzle, T., Hoos, H.H.: MAXMIN Ant System. Future Generation Computer Systems, 16(8):889–914 (2000)
13. Stutzle, T., Hoos, H.: MAX-MIN Ant System and Local Search for the Traveling Salesman Problem. In: IEEE International Conference on Evolutionary Computation, pp. 309–314, Indianapolis (1997)
14. Dorigo, M., Maniezzo, V., Colorni, A.: Ant System: Optimization by a Colony of Cooperating Agent. In: IEEE Transactions on Systems, Man and Cybernetics-Part B: Cynernetics, 26(1):29–41 (1996)
15. Eberhart, R. C., Kennedy, J.: A new optimizer using particle swarm theory. In: 6th International Symposium on Micromachine and Human Science, pp. 39–43, Nagoya, Japan (1995)
16. Yan, X., Zhang, C., Luo, W., Li, W.,Chen, W., Liu, H.: Solve Traveling Salesman Problem Using Particle Swarm Optimization Algorithm. International Journal of Computer Science, 9(6):264–271 (2012)
17. Zhang, J., Si, W.: Improved Enhanced Self-Tentative PSO Algorithm for TSP. In: 6th IEEE International Conference on Natural Computing, pp.2638–2641, Yantai, Shandong (2010)
18. Gutin, G., Yeo, A., Zverovich, A.: Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP. Discrete Applied Mathematics, 117(1):81–86 (2002)
19. Bang-Jensen, J., Gutin, G., Yeo, A.: When the greedy algorithm fails. Discrete Optimization, 1(2):121–127 (2004)
20. Lin, S., Kernighan, B.W.: An effective Heuristic Algorithm for the Traveling-Salesman Problem. Operations Research, 21(2):498–516 (1973)
21. Helsgaun, K.: An effective implementation of K-opt moves for the Lin-Kernighan TSP heuristic. Doctoral dissertation, Roskilde University (2006)
22. Applegate, D., Cook, W., Rohe, A.: Chained Lin-Kernighan for large traveling salesman problems. INFORMS Journal on Computing, 15(1):82–92 (2003)
23. Helsgaun, K.: Solving the equality generalized traveling salesman problem using the LinKernighanHelsgaun Algorithm. Mathematical Programming Computation, 7(3):269–287 (2015)
24. Blum, C., Puchinger, J., Raidl, G. R., Roli, A.: Hybrid metaheuristics in combinatorial optimization: A survey. Applied Soft Computing, 11(6):4135–4151 (2011)