Finding the Optimal Path in 3D Spaces Using EDAs – The Wireless Sensor Networks Scenario

Bo Yuan, Maria Orlowska, Shazia Sadiq

School of Information Technology and Electrical Engineering,
The University of Queensland, QLD 4072, Australia
{boyuan, maria, shazia}@itee.uq.edu.au

Abstract. In wireless sensor networks where sensors are geographically deployed in 3D spaces, a mobile robot is required to travel to each sensor in order to download the data. The effective communication ranges of sensors are represented by spheres with varying diameters. The task of finding the shortest travelling path in this scenario can be regarded as an instance of a class of problems called Travelling Salesman Problem with Neighbourhoods (TSPN), which is known to be NP-hard. In this paper, we propose a novel approach to this problem using Estimation of Distribution Algorithms (EDAs), which can produce significantly improved results compared to an approximation algorithm.

1. Introduction

In wireless sensor networks where sensors are geographically deployed in 3D spaces (e.g., at different depths below the sea surface), it may not be practical to require sensors to directly coordinate with each other to form a communication network to transfer the data back to the server. A mobile robot is needed in this situation to travel to all sensors to collect the data and come back to its starting location. In order to communicate with each sensor, the robot must be physically within its effective range, which is specified by a sphere. The diameter of the sphere is determined by the current battery level of the sensor and is likely to be different among sensors. The optimization problem in this scenario is to design a path for the robot so that it can collect the data from all sensors while the overall travelling distance is minimized.

In its generic form, this routing problem can be regarded as an instance of a class of problems known as the Travelling Salesman Problem with Neighbourhoods (TSPN) [1] where the neighbourhoods are disjoint spheres. In TSPN, a salesman needs to delivery products to a group of clients living in different places. Instead of waiting at home, each client is willing to meet the salesman within a certain region near their houses. The objective here is to find the shortest trip along which the salesman can meet all clients and come back to the starting location. It is easy to see that TSPN is a generalization of TSP, which is known to be NP-hard [10].

Depending on the properties and the connectivity of the regions, a number of approximation algorithms have been proposed. Arkin and Hassin [1] present the first approximation algorithms for a number of special cases, including parallel unit-segments, translates of a convex region, translates of a connected region and parallel

segments where the ratio between the longest and the shortest segments is bounded by a constant. These algorithms can find a valid tour in polynomial time and its quality is guaranteed to be within a constant factor of the optimal tour. The general idea is to carefully pick up a representative point for each of the regions and then apply a TSP algorithm on this set of points. For the general situations, Mata and Mitchell [9] and Gudmundsson and Levcopoulos [6] present some O (log n)-approximation algorithms where n is number of regions. Dumitrescu and Mitchell [4] give an O (1)-approximation algorithm for the case of arbitrarily connected regions with comparable diameters and a PTAS (Polynomial Time Approximation Scheme) for the case of disjoint unit disk regions. For problems closely related to our case (i.e., disjoint spheres with possibly varying sizes), de Berg et al. [3] give a constant factor algorithm with approximation factor $12000\alpha^3$ where α=4 for disk regions and α=8 for 3D spheres. Most recently, Elbassioni et al. [5] show a considerably improved $(9.1\alpha+1)$-approximation algorithm.

One of the major issues of these approximation algorithms is that, despite their polynomial running time, they are often based on some deterministic procedures and there is an inherent lack of global optimization ability. Also, approximating Euclidean TSPN within $(2-\varepsilon)$ has proven to be NP-hard [11]. After all, the performance of these algorithms has only been characterized theoretically in terms of the approximation factors in the worst case, which are often several times worse than the optimal solutions. In real-world situations, simply knowing such a loose bound is obviously of little practical value.

In this paper, we approach this problem from a new perspective by taking advantage of advanced Estimation of Distribution Algorithms (EDAs) [8], which refer to a relatively new paradigm of Evolutionary Algorithms (EAs) [2]. The unique strength of EDAs compared to many other traditional EAs is that they can explicitly capture the dependences among problem variables and use this structural information to conduct more efficient searching, which may often lead to significantly faster convergence speed on challenging optimization problems [8].

2. Methodology

2.1 Problem Specification

The formal definition of the problem is given below. The neighbourhood region corresponding to each sensor is represented by a sphere specified by two parameters: centre e and radius r. A problem instance is then fully specified by the starting position s along with a set of n spheres: $\{s, (e_1, r_1), ..., (e_i, r_i), ... , (e_n, r_n)\}$. Note that the radius of each sphere can be significantly different from others, depending on its own power status. Since sensors are supposed to be geographically distributed, here it is assumed that spheres are disjoint from each other. However, as will be shown later, the applicability of the proposed approach is not directly influenced by this factor. Also, we assume that the starting position is not within any sphere as the robot would otherwise have immediate access to the data contained in the sensor. In practice, such sensors may be removed from the problem at the beginning.

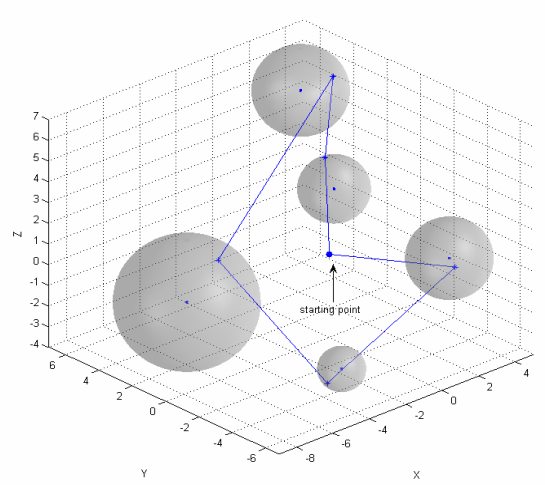

Fig. 1. An example of the layout of the wireless sensors in 3D spaces. There are totally five sensors with each one represented by a sphere of varying sizes. A random path is also shown, which connects all spheres with the starting position.

Fig. 1 shows an example of the optimization problem with five sensors as well as a valid tour intersecting with all spheres. Since the robot can start communication with each sensor as long as it reaches the surface of the sphere, the first intersection point between the path of the robot and each sphere is of the major importance. When the data acquisition from a sensor is finished, the direction of movement of the robot will be solely determined by the location of the intersection point on the next sphere. In other words, each TSPN tour is constructed by sequentially connecting the starting position *s* and *n* first intersection points on the surfaces of *n* spherical regions, referred to as hitting points from now on.

Suppose *p* is the set of *n* hitting points and π is the permutation over *p*. A valid tour is then uniquely specified by a tuple <*s*, *p*, π>. It is easy to see that there are two groups of parameters to be optimized: the locations of hitting points and their order of appearance (permutation) in the tour. Since the hitting points are distributed on the surface of each sphere, the location of each hitting point can be fully specified by two angles: $\theta \in [0, 2\pi]$ and $\varphi \in [0, \pi]$. The actual coordinate values in the xyz-plane can be calculated by:

$$\begin{aligned} x &= x_0 + r \cdot \sin\varphi \cdot \cos\theta \\ y &= y_0 + r \cdot \sin\varphi \cdot \sin\theta \\ z &= z_0 + r \cdot \cos\varphi \end{aligned} \qquad (1)$$

In Eq. 1, (x_0, y_0, z_0) and *r* are the centre and radius of the sphere respectively. In this spherical coordinate, φ is the angle between the z-axis and the line connecting the centre (x_0, y_0, z_0) and the point (*x*, *y*, *z*) while θ is the angle between the x-axis and the projection of this line on the xy-plane. Note that there are different ways to define the ranges of θ and φ, which are functionally equivalent.

2.2 Problem Decomposition

According to the problem specification in the last section, for a problem instance with n sensors, there are totally $3n$ parameters to be optimized, including $2n$ continuous parameters specifying the location information and n discrete parameters specifying the permutation. In this section, three different schemes on how EDAs could possibly be applied to this problem will be analysed, taking into account the trade-off between time complexity and global optimization ability.

In the first scheme, all parameters are encoded into the individuals of EDAs, which means that the entire search space is under exploration and the true global optimum is thus guaranteed to be found in theory. The major issue is that the first part of the individual represents a continuous problem while the second part of the individual represents a combinatorial problem. As a result, it would be quite difficult for EAs/EDAs to efficiently handle such individuals and a significant amount of customization may be required to design problem-specific search operators. On the other hand, EAs/EDAs have not been shown to be the most competitive methods as far as TSP is concerned. In other words, they are not particularly good at solving the combinatorial part of the optimization problem. Instead, there exist some dedicated algorithms that are generally more efficient and effective, especially on large TSP instances.

An alternative scheme is to only encode the location information into individuals and EDAs are only responsible for selecting the set of hitting points (i.e., a continuous optimization problem). In this situation, each individual is simply a vector of angles. The quality or fitness of each candidate set of hitting points is measured by the length of the optimal tour based on these points, which is assumed to be given by an external TSP algorithm. Under the assumption that such effective TSP algorithms are available, this scheme can still guarantee finding the global optimum in theory while the dimensionality of the search space to be explored by EDAs can be significantly reduced ($2n$ vs. $3n$). However, since the evaluation of each individual involves solving a TSP instance, this scheme could become very time consuming when there are a relatively large number of sensors.

In order to avoid the above issues, the scheme adopted in our work is based on the following heuristic: find the optimal TSP tour based on the sensor centres and use the same permutation of sensors in the evaluation of each set of hitting points. In other words, we assume that the permutation of spherical regions π^r in the optimal TSPN tour is always in consistence with the permutation of sensor centres π^c in the optimal TSP tour. In practice, this heuristic is certainly not expected to guarantee optimality but to provide a close estimation of π^r. An important question is to determine when this assumption is valid. Although a rigorous analysis is unavailable at this stage, intuitively, if the diameters of spheres are relatively small compared to the distances among spheres, it is very likely that π^r is equal to π^c. After all, when the diameters approach zero, TSPN also gradually approaches TSP.

One of the major advantages of this problem decomposition heuristic is that the dimensionality of the original problem can be reduced from $3n$ to $2n$ and only a single TSP instance needs to be solved. Another advantage is that, given the knowledge of π^r (estimated by π^c), it is now straightforward to precisely evaluate the quality of each set of hitting points and advanced optimization techniques such as EDAs can be conveniently applied. By contrast, although the idea of problem decomposition is also in-

corporated in many approximation algorithms, the selection process of candidate hitting points is usually conducted without any clear quality information.

Note that, although the permutation of spheres is determined in advance and fixed in the evaluation of all individuals, the quality of the TSPN tours may still vary significantly with regard to different sets of hitting points. To demonstrate this point, a sampling experiment was conducted with 100,000 random sets of hitting points. The quality (tour length) of each set of hitting points was evaluated based on the problem instance with five sensors (10D search space) shown in Fig. 1. The optimal permutation of sensor centres π^c was found using a brute force search due to the small number of sensors in this problem. The distribution of the quality of the 100,000 samples is shown in Fig. 2 from which it is clear that the quality of TSPN tours was sensitive to the locations of hitting points, even if the permutation was fixed.

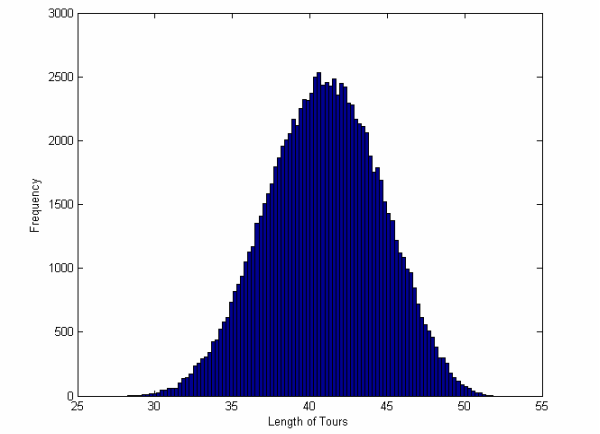

Fig. 2. The distribution of the quality of 100,000 sets of randomly generated hitting points on the five-sensor problem. It shows that there was a large variation on the quality of different sets of hitting points even if the permutation was fixed.

3. Estimation of Distribution Algorithms

The general mechanism of EDAs is to build a statistical model based on a set of promising individuals in each generation. All new individuals are then sampled from the generated model (Table 1). The role of the statistical model is to estimate the structure of the search space where high quality individuals are likely to be found. A continuous EDA named EDA$_{mvg}$ based on a multivariate Gaussian distribution with full covariance structure is used in this paper [8, 12]. In this EDA, the model is specified by the mean vector and the covariance matrix where off-diagonal elements represent the dependence information and all new individuals are generated by sampling from this Gaussian distribution. The advantage of EDA$_{mvg}$ is that it can be implemented very efficiently and still has the capability of capturing complex problem structure. EDAs in this class have shown comparable performance compared to EDAs based on more complex models.

Table 1. The general framework of EDAs

Step 1: Initialize the probability model $P_0(X)$, t=0
Step 2: Sample a population $O = \{X_1, \ldots, X_n\}$ from $P_t(X)$
Step 3: Evaluate individuals in O: $F=\{f(X_1), \ldots, f(X_n)\}$
Step 4: Select a subset of the population $O' \subset O$
Step 5: Update the model: $P_t(X) \rightarrow P_{t+1}(X)$
Step 6: t=t+1
Step 7: Go to Step 2 until stopping criteria are met

In Eq. 1, using the angles θ and φ, it is possible to specify every point on the surface of each sphere. Since we assume that $\pi^r = \pi^c$, it is possible to reduce the original search space to a smaller area compared to the whole surface. Assume that sphere *A* is to be visited immediately after sphere *B*, the distribution of all possible hitting points on sphere *A* can be determined by the diameters of *A* and *B* as well as their relative spatial location (see Fig. 3 for a 2D example). In other words, there is always a "dark side" on each sphere where no hitting points could possibly be located. It is easy to image that, if *A* and *B* are of the same size, the dark side on *A* will account for at least 50% of the surface of *A* (it also depends on the distribution of hitting points on *B*).

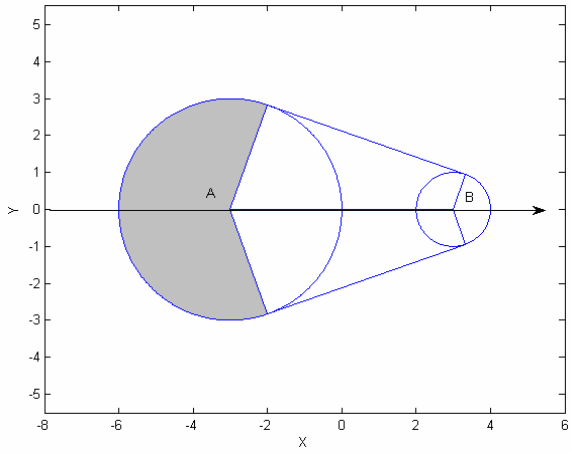

Fig. 3. A 2D example of the "dark side" on sphere A visited immediately after sphere B.

However, it is not convenient to specify the feasible search space analytically in terms of θ and φ and make sure all individuals are generated within this area. Alternatively, each individual is checked before evaluation to make sure that the hitting points are valid. Suppose that the current hitting point on sphere *B* is P_B and the candidate hitting point on sphere *A* is P_A. By connecting P_A and P_B with a line *L*, the intersection point(s) between *L* and *A* can be worked out. If there exists another intersection point P'_A that is closer to P_B, it will replace P_A as the new hitting point on *A* as P_A is not the first intersection point (hitting point) on sphere *A*.

The intersection points can be found by solving for u the following two sets of equations:

$$\text{Line } L: \quad \begin{aligned} x &= P_{B,x} + u \cdot (P_{A,x} - P_{B,x}) \\ y &= P_{B,y} + u \cdot (P_{A,y} - P_{B,y}) \\ z &= P_{B,z} + u \cdot (P_{A,z} - P_{B,z}) \end{aligned} \tag{2}$$

$$\text{Sphere } A: \ (x - x_A)^2 + (y - y_A)^2 + (z - z_A)^2 = r_A^2 \tag{3}$$

4. Experiments

In this section, we empirically evaluated the performance of the proposed techniques for TSPN problems compared to an approximation algorithm called Algorithm A by Elbassioni et al. [5]. The major motivation was to provide some preliminary results to justify the soundness of our methods. More comprehensive experimental studies will be conducted as the future work.

The basic procedure of Algorithm A is to progressively select a hitting point for each sphere and then construct a TSP tour based on the set of selected hitting points. More specifically, all spheres are initially sorted in ascending order based on their diameters. The hitting point on the smallest sphere is selected randomly. In our case, the starting position s is regarded as a zero-diameter sphere and will always be selected as the first hitting point. The hitting point on each subsequent sphere is selected as the point with the minimum distance to the current set of hitting points.

Note that both TSPN algorithms require an external TSP algorithm to either find the optimal permutation of sensor centres (our method) or construct the final tour based on a set of hitting points (Algorithm A). The Lin-Kernighan heuristic [7] was used in this paper, while any other TSP algorithms capable of handling 3D Euclidean spaces will also be appropriate.

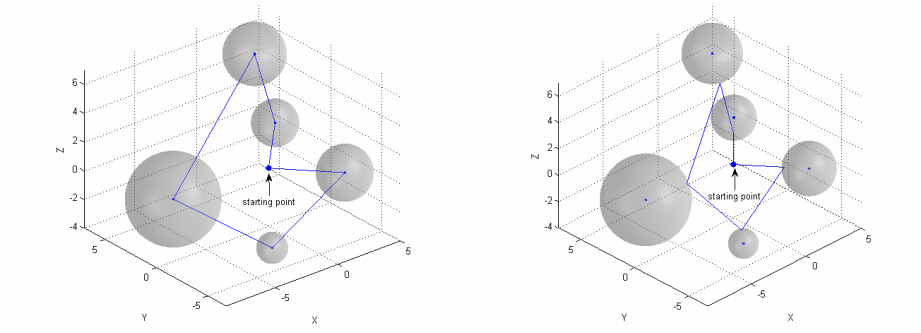

Fig. 4. The optimal TSP tour (length=39.0140) of the five-sensor problem found by a brute force search (left) and the TSPN tour (length=27.2908) constructed by Algorithm A (right).

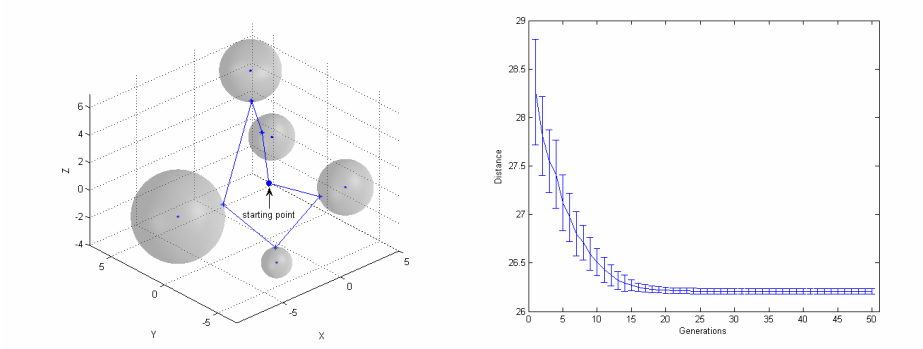

Fig. 5. A typical TSPN tour (length = 26.1830) of the five-sensor problem found by the EDA method (left) and the experimental results averaged over 25 independent trials (right).

The first experiment was based on the five-sensor test problem shown in Fig. 1. Due to its simplicity, a brute force search was used to solve the TSP components in both algorithms. Fig. 4 (left) shows the optimal TSP tour based on sensor centres with length 39.0140. Since the TSP tour is also a valid TSPN tour, it can be used as a benchmark against TSPN algorithms. The corresponding TSPN tour with length 27.2908 constructed by Algorithm A (i.e., an optimal TSP tour found through the brute force search based on the set of hitting points selected by Algorithm A) is shown in Fig. 4 (right).

Next, based on the permutation of spheres in the optimal TSP tour, the EDA was applied to search for the set of hitting points. The parameter values were chosen as population size=100, number of generations=50, truncation selection with selection pressure=0.3 (select the top 30% individuals). No specific parameter tuning was conducted and the above values were largely based on recommended values and a few preliminary trials.

Fig. 5 (left) shows a typical TSPN tour constructed by the EDA approach with length 26.1830. In order to demonstrate the robustness of our method, 25 independent trials were conducted and the mean performance is plotted in Fig. 5 (right), together with error bars showing one standard deviation above and below the mean value. It is easy to see that our approach could reliably find solutions better than the approximation algorithm. Note that there is no parameter to be tuned in Algorithm A and its performance is deterministic.

To further verify the performance of the EDA method, a much more challenging test problem with 25 sensors was randomly generated, as shown in Fig. 6. The starting point was at the origin while sensors were randomly distributed within [-20, 20] in each dimension. The radius of each sphere was randomly chosen within [1, 4] and special care has been taken to make sure that spheres were disjoint from each other. The optimal TSP tour of this problem had length 304 with rounding (the TSP algorithm in use assumes that distances among nodes are integral) while the TSPN tour constructed by Algorithm A had length 271 (with rounding). By contrast, with a larger population size (1000) and more generations (100), the EDA method could again reliably find solutions of significantly improved quality. A typical TSPN tour is shown in Fig. 6 with length 248 (with rounding).

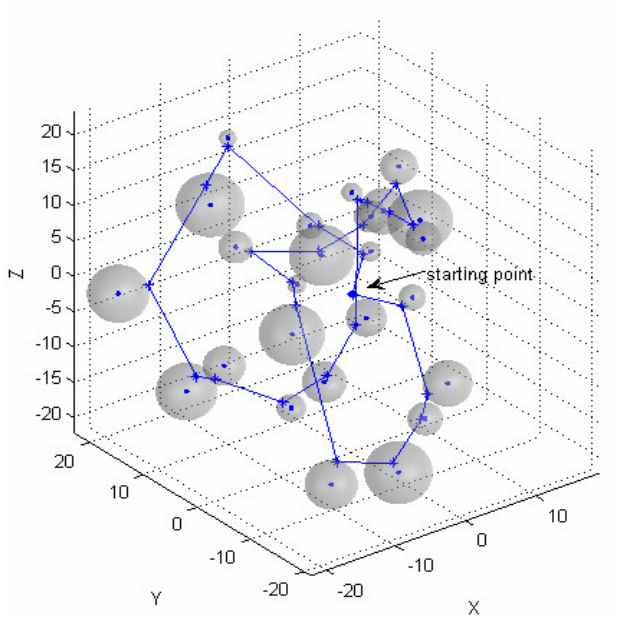

Fig. 6. A random test problem with 25 sensors and a typical TSPN tour (length = 248) found by the EDA method.

5. Conclusions

In this paper, we presented some preliminary work on a novel technique for finding the optimal path in 3D spaces, which was motivated by the data acquisition problem in wireless sensor networks. The core idea is to decompose the original problem into a TSP problem and a continuous optimization problem. In the TSP component, an optimal tour is constructed by an external TSP algorithm based on the sensor centres. The permutation of sensors is then used as the estimation of the permutation of spheres in the TSPN tour. Consequently, the EDA is dedicated to searching for the optimal set of hitting points, represented by a vector of angles.

In the numerical simulations of two test problems, the performance our method has shown to be robust and encouraging and it could significantly improve the quality of the tours constructed by the latest TSPN algorithm.

As to the future work, a major step is to establish some theoretical analysis on the effectiveness of the heuristic (estimation of π^r by π^c) used in our method. It would be favourable to be able to formally investigate the analytical properties of this heuristic in order to understand and quantify its influence on the quality of tours.

Although the EDA performed reasonably well in the experiments, an important task is to further characterize the structure of this type of problems to have a better understanding on what kind of optimization algorithms are mostly applicable and how to incorporate more problem-specific knowledge into these algorithms in order to achieve better performance.

References

1. Arkin, E. M., Hassin, R.: Approximation algorithms for the geometric covering salesman problem. Discrete Applied Mathematics, Vol. 55 (3), (1994) 197-218.
2. Bäck, T., Fogel, D. B.,Michalewicz, Z., Eds.: Handbook of Evolutionary Computation, IOP Publishing Ltd and Oxford University Press (1997).
3. de Berg, M., Gudmundsson, J., Katz, M. J., Levcopoulos, C., Overmars, M. H.,van der Stappen, A. F.: TSP with Neighborhoods of Varying Size. Journal of Algorithms, Vol. 57 (1), (2005) 22 - 36.
4. Dumitrescu, A., Mitchell, J. S. B.: Approximation algorithms for TSP with neighborhoods in the plane. Journal of Algorithms, Vol. 48 (1), (2003) 135 - 159.
5. Elbassioni, K., Fishkin, A., Mustafa, N.,Sitters, R.: Approximation Algorithms for Euclidean Group TSP. In the 32nd International Colloquium on Automata, Languages and Programming, (2005) 1115-1126.
6. Gudmundsson, J., Levcopoulos, C.: A fast approximation algorithm for TSP with neighborhoods. Nordic Journal of Computing, Vol. 6 (4), (1999) 469 - 488.
7. Helsgaun, K.: An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic. European Journal of Operational Research, Vol. 126 (1), (2000) 106-130.
8. Larrañaga, P., Lozano, J. A., Eds.: Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, Kluwer Academic Publishers (2001).
9. Mata, C. S., Mitchell, J. S. B.: Approximation algorithms for geometric tour and network design problems (extended abstract). In the eleventh annual symposium on computational geometry, (1995) 360 - 369.
10. Papadimitriou, C. H.: The Euclidean travelling salesman problem is NP-complete. Theoretical Computer Science, Vol. 4 (3), (1977) 237-244.
11. Safra, S., Schwartz, O.: On the complexity of approximating tsp with neighborhoods and related problems. Computational Complexity, Vol. 14 (4), (2006) 281 - 307.
12. Yuan, B., Gallagher, M.: On the Importance of Diversity Maintenance in Estimation of Distribution Algorithms. In the 2005 Genetic and Evolutionary Computation Conference, (2005) 719-726.