

New Approach for Solving Dynamic Travelling Salesman Problem with Hybrid Genetic Algorithms and Ant Colony Optimization

Farhad Soleimanian
Gharehchopogh
Computer Engineering
Department, Science and
Research Branch, Islamic Azad
University,
West Azerbaijan, Iran

Isa Maleki
Computer Engineering
Department, Science and
Research Branch, Islamic Azad
University,
West Azerbaijan, Iran

Masoum Farahmandian
Computer Engineering
Department, Science and
Research Branch, Islamic Azad
University,
West Azerbaijan, Iran

ABSTRACT

Dynamic travelling salesman problem (DTSP) is one of the optimization issues which it is not solvable with classical methods. To solve this problem, various solutions in the literature can be seen that each has advantages and disadvantages. Genetic Algorithm (GA) and Ant Colony Optimization (ACO) have been good to solve the DTSP. In this paper, we highlight a new algorithm by combining genetic and ACO which gives us a better solution for DTSP. In hybrid algorithm, suitability of algorithm and travelled distance for DTSP has been considered. Obtained results suggest that Hybrid algorithm does not establish easily in the local optimum and possesses a good speed in convergence for comprehensive answer.

Keywords

DTSP, Optimization, GA, ACO, Hybrid algorithm.

1. INTRODUCTION

DTSP is the most important hybrid optimizing problem. In the past two decades, new computational methods such as refrigeration algorithms [1], ACO [2], GAs [3, 4] and Artificial Neural Networks [5, 6] to solve the problem of DTSP have been suggested. Of these methods, which both GAs and ACO is based on repetitive, in finding solutions close to the optimal solution for problems, hybrid optimization has been considered mostly. ACO system was presented for the first time by Dorigo et al. [7] as a Multi-Agent Solution for difficult problems of DTSP. ACO is one of collective intelligence algorithms [7, 8]. That is the algorithms in which people collaborate to achieve a final goal. The main idea of collective intelligence algorithms is to use the real life of creatures that without using complex mechanisms run their social life in all its aspects as best as possible. ACO is one of the optimization algorithms which are inspired from observations and studies on the ACO.

GA is an iterative process that includes a population of individuals or chromosomes which are randomly or heuristic [9] that each one by a string of symbols as a gene in possible solution on the issue has been coded. In this search space, all possible solutions to the problem are examined. GAs usually are used in search spaces which are very large. People can select an operation, integration and mutation to obtain a

higher fitness values to search for the best solution to the problem.

The hybrid algorithm that we presented, the objective function, is intended to minimize the travelled distance. To minimize the route, the travelled distance by ants as a suitability value, is used. Therefore, it is possible to choose the suitability according to travelled, in such a way that for shorter distances more amount of suitability is considered.

We have organized the general structure of this paper as following: after the introduction we have focused on the literature review; in the third section we introduce the DTSP; in the fourth section of this paper, we describe the proposed solution; in section five we explain the results and evaluation of the proposed method; and finally in the sixth section, we will conclude. In this paper we have reached to good results compared to the optimal algorithm of ACO and GA. Using a hybrid algorithm, path length of DTSP is close to optimum.

2. LITERATURE REVIEW

In recent years various smart optimization methods have been used to solve the DTSP. Solving DTSP based on ACO was first introduced by Guntsch and Middendorf [10]. One of the other classes of hybrid optimization methods are evolutionary algorithms. According to Huang [11] Evolutionary algorithms are a good choice for solving the DTSP.

2.1 Genetic Algorithms

GAs according to the type of mutation operators and used combinations are different with each other. GA proposed by Fogel has shown a good performance in solving DTSP [12]. This algorithm uses the method of exclusion as the combination operator and the reverse mutation operator. In this algorithm, distance of the route is considered as suitability.

GAs were proposed for the first time by John Holland based on Darwin's theory of evolution [13]. GA is one of repetitive accidental process that does not necessarily guarantee convergence; however, with regard to the possible solutions it increases this probability attempted in the method of proposed solution in this paper to be done. The Condition for stopping repetitions of this algorithm can be proved by some fix predetermined amounts, such as the number of generations or the suitability level which should be acceptable.

GA method is an efficient method that has gathered the positive characteristics of the random and ultra-creativity

methods. But some changes always according to the characteristics of the problem, in the standard GA can be proposed.

The first step in GA is to create the initial population as randomly or heuristically. Each member of the population is called a chromosome which suggests a solution for the problem [14]. Chromosomes evolve in repeated periods and each period is called a generation. In each period, the population changes and creates a new generation that is more effective in reaching the optimal answer. In order to maintain the best answer of each generation and prevent the destruction of them, elitist techniques can be used.

Evolution of chromosomes is done in two ways. In the first stage, some the chromosomes are randomly selected from the existing population and hybrid together, so new elements are created. The second stage is called mutation. In this stage one or some chromosomes are selected randomly in every repetition and one of the chromosome's genes is selected and according to special mechanism changes. Thereupon, new chromosomes are obtained. In the final stage of the initial population of expanded populations of the members shall be elected and shall be considered as a new generation. There are different methods for selecting stage like roulette wheel selection. GAs in the kind of combination, mutation and selection and/or order of applying differs.

One reason for the popularity of GA is the loss of requiring to high level and professional mathematical model. In general, the ideas of evolutionary and population improvement are used in these algorithms. Other applications of GAs to optimize the schedule of courses can be used for a hypothetical school. Erin and Kepler designed and implemented optimum tables for weekly lessons using GAs [15].

2.2 Ants Colony Optimization

ACO which has been used is exactly the algorithm that was presented by Doryegu in 1996 [9, 16]. In this algorithm, m ants by memory are created in the first phase; these ants are placed randomly on n nodes. There is an initial amount of pheromone at each node. In DTSP, the aim is to find an order with minimum distance for the salesman that is implemented by the help of optimizing ACO [2, 17].

Ants while walking leave a chemical substance called pheromone. Although this substance will be evaporated soon, it remains in the short term on the floor as the ants footprints [9]. Ants are capable of producing pheromones that help them to find the shortest path to food. The ants that select the shortest path create stronger pheromone trail rather than the ones select longer routes. Since more pheromones will attract ants better, more and more ants choose the shorter route to that extent that all ants have chosen the shortest route and move in that route. For further investigation, we assume, for example, there are two paths to food sources that are different in lengths. Ants select both directions with equal probability. The ants that has gone the shorter route and returned produce the most pheromones earlier than the rest. Consequently, other ants choose this path sooner and strengthen the pheromone on this path. Finally, all the ants travel the shortest path to food.

The probability of moving from city i to city j for ant k at time t based on equation (1) is expressed. In this relationship, edge of visibility is $\eta_{i,j}$ and equals $1/d_{i,j}$ (nearer cities have more possibility of selection) and the amount of $\tau_{i,j}$ existing pheromone on edge is at the time t . Allowed is the set of cities

that ant k has not met yet and if it is possible can meet them in the next step. α and β parameters also affect the amount of pheromones that has been shed on the edge and the edge of visibility.

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{j \in allowed_k} [\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta} & \text{if } k \in allowed_k \\ 0 & \text{otherwise} \end{cases}$$

Equation (1)

The amount of pheromone on edge (i, j) at time t is given with $\tau_{i,j}$. Each ant at time t visits the next town and meets it at the time $t+1$. So if in each step (in the time interval t to $t+1$) m movements by m ants are done, in every n step (which is called a round), each ant's algorithm has completed a tour. At this time the amount of pheromones has been shed on the edge - according to relationship (2) will occur.

$$\tau_{ij}(t+n) = (1-\rho) \times \tau_{ij}(t) + \Delta\tau_{ij}$$

Equation (2)

That $1-\rho$ specifies the pheromone evaporation rate from t distance to $t+n$. To prevent excessive accumulation of pheromone on an edge, the limits for ρ are considered to be $0 < \rho < 1$. The greater ρ value is, the more pheromone evaporation rate goes up.

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k$$

Equation (3)

$\Delta\tau_{ij}^k$ Is the amount of pheromone that ant k leaves on path (i, j) and the time interval t to $t+n$ which is expressed as following:

$$\Delta\tau_{ij}^k = \begin{cases} Q/L_k & \text{if } k \text{ ant uses edge } (i, j) \text{ at time } (t, t+n) \\ 0 & \text{otherwise} \end{cases}$$

Equation (4)

That Q is a fix amount and L_k is the travelled route by ant k .

One of the other applications of ACO is positioning of computer networks. Most operator algorithms in positioning computer networks, due to the distributed network nodes, have been inspired by life of colony of ants. Finding the shortest path is very important for computer networks. A new positioning algorithm based on this method presented in 1997 [18]. He used operators the same as ants used in his own method. Caro and Dorigo based on ACO suggested AntNet for positioning the computer networks [19,20].

3. DTSP

DTSP is a modified version of the travelling salesman. DTSP first was introduced in 1988 by Psaraftis [21,22]. In the DTSP, a salesman starts his trip from a city and after a complete trip comes back to his own city again and pass each city for once and of course he should pass all cities. In DTSP

cities can be deleted or added [23]. The main purpose for this trip is to find the shortest path. The goal is to find the shortest route for this trip. Consider a set of n cities where is the problem of travelling salesman to find the shortest path for all n the city met only once. The problem as a graph (N, E) is shown where N is the set of cities (graph nodes) and E the set of routes between cities (edge graph). If the Euclidean distance $C_{i,j}$ between cities is i and j it will be calculated as following [24]:

$$C_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Equation (5)

Each ant selects the next town to meet with the potential function which its distance from the city and the amount of pheromone on the edge is considered. To prevent from creation of around in the chosen route by an ant the selection of edges leading to a visited city is forbidden. When an ant meets all the cities, puts all of its own pheromones on all passed edges.

4. PROPOSED SOLUTION

Among the methods for solving DTSP, meta-heuristic methods [25] have high capacity in achieving close solutions to optimum. In recent decades, many efforts to improve the quality of the answer - obtained by this algorithm are applied. Hybrid algorithms are part of these activities. We have had a special look at ACO due to the similar structure to GA to combine a new algorithm.

In the hybrid algorithms, initial answers of ACO among the obtained data from routes for the mutations are selected by GAs. GAs answers have been used for a wide range of ants search. In the GA there are optimal solutions to achieve a better answer in every generation. Consequently, the GA using efficient routes of ACO, explores the most efficient way in search space till in a new searching point moves towards the better solutions. GA as a computational algorithm for optimization with regards to a set of answer points in each iteration computation searches effectively the different parts of answer. In hybrid method unlike ACO all the all-round solution space is searched; therefore, there will be less possibility for convergence to local optimum. You see the overall view of the hybrid algorithm in Fig 1. As the figure implies the algorithm has three basic concepts: ACO, the GA and comparing answers part which is explained in the following.

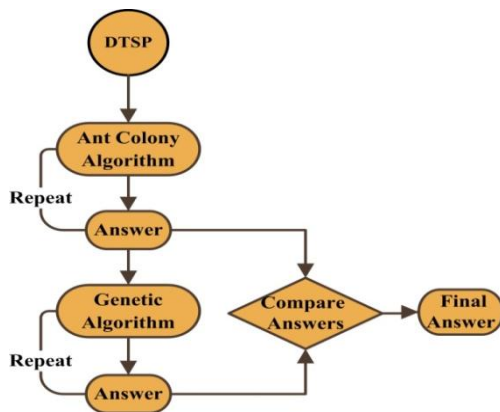


Fig 1: Proposed Hybrid Flowchart

Quasi proposed algorithm code includes the Fig 2. based on ACO and GA:

1. Initialization
2. Set all ants at the starting city
3. Repeat until all cities have been visited
4. Update Pheromone
5. Terminating condition
6. Create initial population with Shortest-Route ACO
7. Repeat (Until terminated)
- Calculate Fitness for each chromosome evaluate fitness, Selection, Crossover operator, Mutation operator
8. Check for termination criteria
9. Output the best individual found
10. Compare (Solution Best ACO & GA)
11. Repeat (Compare)
- Length Tour Until best Solution

Fig 2: Quasi Code of Proposed Algorithm

4.1 Chromosome Designing

To solve the problem by GA, a chromosome containing the gene information is to be considered. Classical GAs use binary strings to create a chromosome which is not suitable for problems like dynamic travelling salesman. In this method, chromosomes are represented as a string of natural numbers which each of these numbers is related to a special parameter in the problem space. Table 1 shows a case of these chromosomes for a problem with 6 cities. The application of this type is encoded in solving DTSP. Here the chromosome represents the order of cities that DTSP should pass.

Table 1. View of Chromosome

1	5	3	2	4	6
---	---	---	---	---	---

4.2 Creating Initial Population

After determining the coding system and identifying any response to chromosome conversion method, the initial population of chromosomes can be produced. In most cases, the initial population is generated randomly, but sometimes innovative methods for enhancing the speed and quality of the algorithm is used to generate the initial population. The numbers of algorithms created according to ACO with regard to length of route are considered as members of the population. Members of the initial population using the travelled routes by the ants are created, so that each member represents an answer for problem. Assuming that m is the number of members, $m-1$ member of the population with random permutations are produced. For creating the last member of the population, we use the nearest not-met neighbor method. We call this permutation approximate permutation. The last member added to the population with the greatest similarity is the final answer.

For example, in Fig 3. how the initial population are created for the visited cities, by assuming $n = 6$, is shown. The first five members of the population by five random permutations $\langle 2,4,5,1,6,3 \rangle$, $\langle 4,5,6,2,3,1 \rangle$, $\langle 5,6,2,4,1,3 \rangle$, $\langle 3,6,2,5,4,1 \rangle$ and $\langle 2,4,3,1,5,6 \rangle$ are created. For the sixth permutation we use not-met nearest neighbor method. Assuming the starting city is 1, sixth permutation will be $\langle 1, 5, 4, 6, 3, 2 \rangle$. Then the created population is ordered according to created value of distance length. The less the length of the route is, the more suitability answer will be and the probability of participation to produce the next generation will also increase.

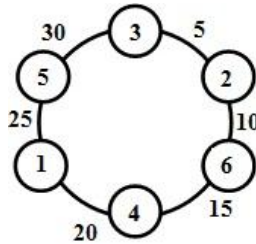


Fig 3: View of Permutation of Cities

4.3 New Operator Integration

So far, common operations such as integration PMX [26], for solving DTSP are provided, which can be interesting in their selves. In this method, two numbers randomly are chosen as cut-off points, then the two cut-off points between the two chromosomes are exchanged and then both parts are initialized so that in none of the two chromosomes the repetition does not occur. Integration operator with probability P_c is done. Unlike the other integration methods, PMX gives important information on a tour of nearby towns. We choose two chromosomes of P1 and P2. Child chromosomes are produced as following:

1. The first gene in the chromosomes of P1 and P2 is selected as the city of origin. This city is called C.

2. The two parent chromosomes are selected and the first two genes means and j in one of two parent chromosomes are selected. Then the same genes are selected. In other two parent chromosomes. Distance between P1 and P2 in C and cities that chromosomes are references to them, is calculated. Obtained distance from the genes of first and second chromosomes P1 and P2, if chromosome P1 in the distance is less than or equal to P2 chromosome, then the city P1 is inserted in Child1 and chromosomes P1 and P2 instead of visited cities, become zero. But if the chromosome P1 is larger than the chromosome P2, then the next town for visiting is selected from the chromosome P2. Cities for the next visits should be equal to the first gene of each chromosome. For this purpose, the exchange action should be done between the genes. The condition of stop will be the time when the number of non-zero cities of chromosomes P1 and P2 are equal to zero. To produce Child 2 the above steps are performed, with this difference that counting of the chromosomes will be from last to beginning. With this action, two new chromosomes are produced which are called the children of two parent

chromosomes for example a DTSP problem with 6 cities as been considered. Matrix of distances between cities is as follows (Table 2.):

Table 2. Cost Matrix

City	1	2	3	4	5	6
1	0	25	50	20	25	65
2	25	0	5	65	40	10
3	50	5	0	50	30	82
4	20	65	50	0	27	15
5	25	40	30	27	0	56
6	65	10	82	15	56	0

In Fig 4. Select two chromosomes with the hypothetical, the results of the proposed method to create child chromosomes are shown.

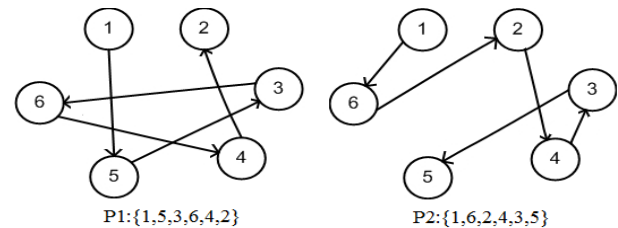


Fig 4: View of Chromosomes P1 and P2

Produced Children of Chromosomes P1 and P2 are shown in the Table 3.

Table 3. View of Chromosomes P1 and P2

P1	1	5	3	6	4	2
P2	1	6	2	4	3	5
Child1	1	5	3	4	2	6
Child2	2	4	6	5	3	1

4.4 New Mutation Operator

Mutation Operator's duty is to prevent from trapping in local optimum points in the algorithm [27]. In the proposed method for implementing the mutation operator, the two genes are randomly selected and exchanged. The new mutation operator considers the relationship between cities in the DTSP and acts in this way that one city is selected randomly and then the nearest city to selected city is considered. The mutation operator is done on the chromosomes with the probability of P_m .

4.5 Way of Comparison of Answers in the Proposed Solution

Hybrid algorithm is designed such that both algorithms pay to finding the most optimal route in problem by the condition to

complete it. In hybrid method, the optimal routes are given to ACO as the initial population of GA. So that answers to ant optimization algorithm as the current population GA are given.

After transferring the answers to the GA, the length of obtained routes by ACOs optimized and compared the shortest obtained route by shortest route algorithm with ACO and finally, the shortest route as the final answer is returned..

5. RESULTS AND DISCUSSION

In this section, experimental results of the proposed algorithm for solving DTSP are shown. These results show the significant improvement of the hybrid algorithm to ants' optimum algorithm and GA. 30 cities and 100 iterations in the algorithm are considered in the done experiment. There is a combination of several parameters which affect the algorithm performance.

In this paper the effect of a combination of these parameters in the algorithm will be analysed. According to conducted tests, ACO algorithm parameters such as number of ants, the amount of pheromone and pheromone evaporation is the percentage change of each of these parameters which is effective on the performance of ACO. Parameter values for implementation of algorithms in Table 4 are shown. The values according to numerous performances of the program are considered to achieve near-optimal solution to increase optimum.

Table 4. Value of Parameters in the Proposed Algorithm

Parameter Name	m	α	β	Pc	Pm
Value	30	1	5	0.9	0.5

Table 5. shows the comparison with two algorithm hybrid algorithm of ACO and GA for solving DTSP after 5 times of the algorithms is given. As is apparent from the results, the hybrid algorithm provides better results along the way.

Table 5. Results Comparison (number of runs = 5)

Algorithms	Average Solution	Best Solution	Worst Solution
ACO	385	340	384
GA	471	351	826
Hybrid Algorithm	382	340	369

As you can see, the proposed method has achieved a better answer compared to the two algorithms of GA, ACO. According to the results of the hybrid algorithm in Table 5. for solutions obtained from ant's colony optimization algorithm GA is better. The proposed algorithm is the best way to answer such as replied, ACO algorithm and the two other cases of this method has achieved a better answer.

Because the implementation of algorithms have more favorable results, Table 6. compares with two algorithm Hybrid algorithm, ACO and GAs to solve the problem of DTSP after 10 times running.

Table 6. Results Comparison (number of runs = 10)

Algorithms	Average Solution	Best Solution	Worst Solution
ACO	385	340	368
GA	464	349	826
Hybrid Algorithm	384	340	358

As you can see along the way with 10 times more running the algorithms the length of routes has been more efficient. So we came to the conclusion that the optimal number of path length affects algorithm performance. A case of implementing optimal way after ten times running for 30 cities, using a hybrid algorithm in Fig 5 is shown.

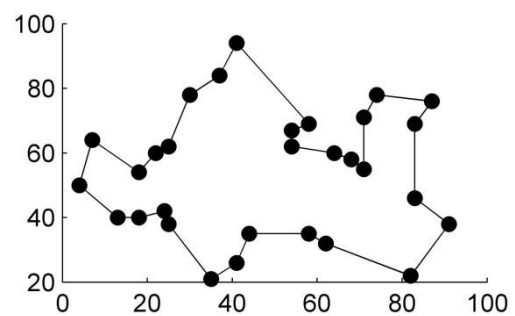


Fig 5: Optimal Path with Hybrid Algorithm

5.1 Combination of Algorithm Parameters

To analyse the effect of parameter α (the amount of pheromones has been shed on the route) on the performance of the algorithm, this parameter equals the numbers 0.1, 0.3, 0.5, 1 and 2. Analysis of results is presented in Table 7., as it is visible, large or small amounts of Parameter alpha influences the increases and decreases along the length of route. If the parameter value α is high, the route length will increase.

Table 7. Impact of Parameter α

Number of the Cities	$\alpha=0.1$	$\alpha=0.3$	$\alpha=0.5$	$\alpha=1$	$\alpha=2$
30	340	340	340	340	342

To investigate the effect of parameter β (the importance of relationship between pheromones and direction) on the performance of the algorithm, we have put this parameter equal to the numbers 1, 2, 3, 4 and 5. Analysis of results in Table 8. is presented, as can be seen, placing the value of beta to 5 has shown a decreasing trend along the way. So by putting the value 5 for the parameter beta is more efficient along the way.

Table 8. Impact of Parameter β

Number of the Cities	$\beta=1$	$\beta=2$	$\beta=3$	$\beta=4$	$\beta=5$
30	618	441	381	356	340

6. CONCLUSION

GA and ACO have speedy convergence; however, they cannot solve the problem of premature convergence of local optimum alone. So to avoid this problem, a hybrid algorithm in this paper has been used to solve these problems. Using a combination of GAs and ACO the search-process to find the optimal path dramatically increases. The experimental results suggest that the proposed algorithm cannot be easily placed in a local optimal solution and can be found close to the optimum. It is worth noting that the implementation of proposed algorithm, the does not have complexity.

7. REFERENCES

- [1] Lo, C. C., & Hus, C. C, "Annealing framework with learning memory", IEEE Transactions on System, Man, Cybernetics - Part A, vol. 28, no. 5, pp. 1–13, 1998.
- [2] Dorigo, M., & Gambardella, L. M, "Ant colony system: a cooperative learning approach to the traveling salesman problem", IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, pp.53–66, 1997.
- [3] Tsai, H. K., Yang, J. M., Tsai, Y. F., & Kao, C. Y, "Heterogeneous selection genetic algorithms for traveling salesman problems", Engineering Optimization, vol. 35, no. 3, pp. 297–311, 2003.
- [4] Albayrak, M., & Allahverdi, N, "Development a new mutation operator to solve the Traveling Salesman Problem by aid of Genetic Algorithms", Expert Systems with Applications, vol. 38, no.3, pp. 1313–1320, 2011.
- [5] Leung, K. S., Jin, H. D., & Xu, Z. B, "An expanding self-organizing neural network for the traveling salesman problem", Neuro computing, vol. 62, pp. 267–292, 2004.
- [6] Masutti, T. A. S., & de Castro, L. N, "A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem", Information Sciences, vol. 179, no. 10, pp. 1454–1468, 2009.
- [7] M. Dorigo, V. Maniezzo, A. Colomi, "Ant system: optimization by a colony of cooperating agents", IEEE Trans. on Systems, Man and Cybernetics, Part B, Vol.26, No.1, pp.29-41, 1996.
- [8] M.Dorigo, L.M.Gambardella,"Ant colony system: a cooperative learning approach to the traveling salesman problem", IEEE Trans on Evolutionary Computation, Vol.1, No.1, pp.53-66, 1997.
- [9] Goldberg. D. E, Genetic Algorithms in Search, Optimization and Machine Learning, Reading, MA: Addison-Wesley, 1989.
- [10] Guntsch, M., Middendorf, M,"A population based approach for ACO". In: Cagnoni, S., Gottlieb, J., Hart, E., Middendorf, M., Raidl, G.R. (eds.) EvoWorkshops 2002. LNCS, vol. 2279, pp. 72–81. Springer, Heidelberg (2002).
- [11] Huang, Z., Hu, X., Chen, S,"Dynamic traveling salesman problem based on evolutionay computation". In: Proc. of the 2001 IEEE Congr. onEvol. Comput., pp. 1283–1288 (2001).
- [12] M.b. Fogel., "The genetic algorithm for TSP", IEEE Transaction on systems, 16:1-13, 1986.
- [13] Whitely D., "A Genetic Algorithm Tutorial", Journal of Statistics and Computing Vol. 4,1994, pp. 65-85.
- [14] Chatterjee, S., Carrera, C., & Lynch, L. A, "Genetic algorithms and traveling salesman problems", European Journal of Operational Research, vol. 93, no. 3, pp. 490-510, 1996.
- [15] Vorac J., Vondrak I., Vlcek K., "School Timetabling Using Genetic Algorithm", Technical Report, VSB-Technical University of Ostrava, Czech Republic, 2002.
- [16] M.Dorigo, G. Di Caro, and L. M. Gambardella." Ant algorithms for discrete optimization. "Artificial Life, 5(2):137–172, 1999.
- [17] Dorigo, M., Birattari, M., &Stutzle, T,"Ant colony optimization: artificial ants as a computational intelligence technique", IEEE Computational Intelligence Magazine, vol. 1, no.4, pp. 28–39, 2006.
- [18] R. Schoonderwoerd, O. Holland, and J. Bruten, "Ant-like agent for load balancing in telecommunications network", in Proc. of the 1stInt. Conf. on Autonomous Agents, pp. 209-216, 1997.
- [19] G. D. Caro and M. Dorigo, "Mobile agent for adaptive routing," in Proc. 31st Hawaii Int. Conf. on System Science, 1998.
- [20] G. D. Caro and M. Dorigo, "AntNet: distributed stigmergetic control for communications networks," J. of Artificial Intelligence Research,vol. 9, pp. 317-365, 1998.
- [21] Psaraftis, H.N,"Dynamic vehicle routing problems". In: Golden, B.L., Assad, A.A. (eds.) Vehicle Routing: Methods and Studies, pp. 223–248. Elsevier, Amsterdam (1988).
- [22] Li, C., Yang, M., Kang, L.: "A New Approach to Solving Dynamic Traveling Salesman Problems". In: Wang, T.-D., Li, X., Chen, S.-H., Wang, X., Abbass, H., Iba, H., Chen, G., Yao, X. (eds.) SEAL 2006. LNCS, vol. 4247, pp. 236–243. Springer, Heidelberg (2006)
- [23] Michael Guntsch, Martin Middendorf, and HartmutSchmeck. "An ant colony optimization approach to dynamic TSP. In Lee Spector et al., editor, Proceedings of the Genetic and Evolutionary Computation", Conference (GECCO-2001), pages 860–867, San Francisco, California, USA, 7-11 July 2001. Morgan Kaufmann.
- [24] Ray, S.S., Pal, S.K., Bandyopadhyay, S., "Genetic Operators For Combinatorial Optimization In TSP And Microarray Gene Ordering", Springer, 2007
- [25] S. N. Sze, "Study on Genetic Algorithms and Heuristic Method for Solving Traveling Salesman Problem," M.S. dissertation, Faculty of Science, UniversitiTeknologi Malaysia, Johor, Malaysia, 2004.
- [26] Goldberg, D.E., and R. Lingle. "Alleles, Loci, and the Traveling Salesman Problem." Proceedings of the First International Conference on Genetic Algorithms and Their Application, edited by Grefenstette J., Lawrence Erlbaum Associates, illsdale, NJ,1985, pp. 154-159.
- [27] M. Bonyadi, M. Azghadi and H. Shah-Hosseini, "Population based optimization algorithms for solving the Travelling Salesman Problem", ITECH publications, ISBN 978-953-7619-10-7, pp: 1-34, 2008.