Optimization multiple traveling salesman problem by considering the learning effect Function in skill and workload balancing of salesman with using the FireFly Algorithm

Mostafa Mohammadi
Master of science in in industrial engineering ،Mazandaran University of Science and Technology
Mostafa.mohammadi989@Gmail.com

Golman Rahmanifar
Master of science in in industrial engineering ،Mazandaran University of Science and Technology
Golman_Rahmanifar@yahoo.com

Ghasem Kaveh Garna
Master of Science in in industrial engineering ،Islamic Azad University of Noor
Kaveh.garna@Gmail.com

Abstract

The Multiple Travelling Salesman Problems (MTSP) are one of the main scopes of discontinuous optimization problems and is the extension of the well-known travelling salesman problem.
The (MTSP) can in general be defined as follows: Given a set of nodes (cities), let there be m salesmen located at a single depot node. The remaining nodes that are to be visited are called intermediate nodes. Then, the MTSP consists of finding tours for all my salesmen, who all start and end at the depot, such that each intermediate node is visited exactly once and the total cost of visiting all nodes is minimized.
The purpose of this paper is following a set of paths for the salesman with the aim of balancing the workload between them. So that number of nodes don't consider predetermined interval for each salesman. But visit time of each node for each salesman follows oriented position learning effect function that depends on the skill of each salesman, and these nodes are determined according to the seller's acquisition skills using a number of repetitions. to the best of our knowledge, this paper presents the first study that brings learning effect in the distribution and in order to achieve optimum quality and local escape answer using a meta-heuristic algorithms. In this study, a Meta heuristic algorithm, called Fire Fly Algorithm (FA), has been presented for solving the MTSP.

Keywords:
Multiple Travelling Salesman Problems (MTSP), optimization, Fire Fly Algorithm (FA)، learning effect

1. Introduction

The MTSP is defined as: In a given set of nodes (cities), let there are located m salesmen in a single depot node. The remaining nodes that are to be visited are intermediate nodes. Then, the MTSP consists of finding tours for all my salesmen, who all start and end at the depot, such that each intermediate node is visited exactly once and the total cost of visiting all nodes is minimized. The metric cost can be defined as terms of distance, time, etc. Possible variations of the problem are as follows: Single vs. multiple depots: In the single depot, all salesmen finish their tours at a single point while in multiple depots the salesmen can either return to their initial depot or can return to any depot keeping the initial number of salesmen at each depot remains the same after the travel. Number of salesmen: The number of salesmen in the problem can be fixed or a bounded variable. Cost: When the number of salesmen is not fixed, then each salesman usually has an associated fixed cost incurs whenever this salesman is used. In this case, minimizing the requirements of salesman also become an objective. Timeframe: In this case, some nodes need to be visited in a particular time period that are called time windows which is an extension of the MTSP, and referred as multiple traveling salesman problem with specified time frame (MTSPTW). The application of MTSPTW can be very well seen in the aircraft scheduling problems. Other constraints: Constraints can be on the number of nodes that each salesman can visit, maximum or minimum distance a salesman travels or any other constraints. The MTSP is generally treated as a relaxed vehicle routing problem (VRP) where there are no restrictions on capacity. Hence, the formulations and solution methods for the VRP are also equally valid and real for the MTSP if a large capacity assigned to the salesmen (or vehicles) However, when there is a single salesman, then the MTSP reduces to the TSP (Betas, 2006).
There is some study that considers MTSP load balancing objective for instance T. Lee, J. Ueng (1999), developed an integer programming model for vehicle routing problems with two objectives, the first objective is minimizing the total distance and second is balancing the workload among employees. as much as developed a heuristic algorithm for solving the problem. Rita Ribeiro(2001), presented a new extension of the basic vehicle routing problem with three objectives and multiple periods. The first objective is cost minimization, the second is balancing work levels and the third is a marketing objective. P. Lacomme (2015), addressed the vehicle routing problem with route balancing, with two purposes, the first objective is minimizing the total routing cost and the second is balancing between the routs by minimizing the difference between the largest and smallest route cost. As it was considered in literature background, "balancing" defined is treated as "the number of visits customers, total length of the tour, total travel time for each salesman and etc. But In the real world, each salesman usually has limited time to visit customers and the important degree of store is different together so that stores (nodes) are classified according to criteria as average purchase, average weight of purchase, diversity of buying goods and etc. Tour time duration for each salesman consists of two time component: first component is traveling time between two nodes and the second is servicing time at each node. It seems that make associate the time duration of the tour to the number of visits nodes and balancing the routes based on number of visited nodes can't be sufficient because the factors as salesman skill and importance degree of stores (nodes) are effective. Servicing time of each customer is different regard to its type also salesman skill influence on service time of each nodes. Also in this research, to approach real world, in addition to mentioned factors, the learning effect has been expressed, too. Namely, time of servicing customers isn’t regarded as traditional methods during a fixed time. In this paper, a position-based learning function has been used in which learning is influenced by the number of conducting servicing. To the best of our knowledge, this paper presents the first study that brings learning effect in the distribution system.
This research determines the optimum route for salesmen regard to their skill and quality level of stores (nodes), such that it will lead to reality considering learning effect for salesmen.

The remainder of this paper is organized as follows. Definition of problem and mathematical model are presented in section 2, section 3 deals with the proposed FA algorithm. The computational results are provided in Section 4 and the conclusions are presented in Section 5.

2. Problem definition:

This paper presents a model for multiple traveling salesman problem by considering the position-oriented learning effect Function in skill and workload balancing of salesman. Salesman routing problem determines assigned customers and visit sequence of each salesman. In this research the object determines a set of routs in order to balance the workload of salesmen noticing their skill and quality level of customers, and finally it causes salesmen satisfaction. Traditionally a service process time of each salesman for each customer assumes to be fixed and independent of visit sequence. Since most of the time repeating visit increases salesman's ability and then decreases visit process time, in this paper an oriented learning function is applied to determine salesmen visit rout, so that visit time for each customer depends on customer's location in visit sequence.
In this section, we first define the assumption that is used throughout this paper, followed by the description of the problem.

2.1. Problem assumption:

In this model, a number of assumptions have been considered in regard to practical situations:
- Every seller at per time service, just to one store.
- Every store is only allocated to no seller.
- The time of passing routes by all sellers from one group to other is considered as fixed time.
- After finishing the works, it is not necessary for sellers to return depot.
- All sellers start their work from the depot.
-Weight work load per group is in conformity with stores quality level and skill of per sellers.
- service time to every store by per seller, referred to repeating that activity and its situation in visit sequence effected by situation-oriented learning function, is determined as follow: $pr_{ip} = p_{im} * p^{-\alpha}$
Where p_{im} is standard visit time of consumer j by ith seller and pr_{ip}represents actual visit time of consumer I at pth place and p indicates visit place of ith consumer in a sequence and a is "learning impact" coefficient.

2.2. The mathematical model

The issue in this case is defined as a set of n distinct Customer n = {1 ... n} that should by V salesman v = {1 ... v} are meanwhile meeting each customer only by one of the visitor, service and gets every single vendor at any moment to be able to service a store and each customer during a period of time, that business volume would be applicable only once the service gets, target The issue is at least off the travel time to the seller according to the mentioned factors all clients are assigned to any business vendor to work with vendors they satisfaction homogenization time.
In this section, we first define the index, input parameters and variable of model that is used throughout this paper, followed by the description of the problem.

2.2.1. Index:

N: number of customer points or demand centers
I: counters related to points of demand
J: counters related to points of demand
M: the number of the visitor

m: counters related to the visitor
P: number of processing locations
p: counters related to processing locations

2.2.2. Input parameter:

p_{im}: The normal processing time (visit time) for node *i* by salesman
w_i: Grade of each customer
IS_i: travel time from depot to each customer
a: The learning rate (for all customer)
S_{ij}: The travel time from customer *I* to customer *j*

2.2.3. Variable of model:

pr_{im}: The actual processing time or visit time of node *i* by salesman
X_{ipm} : If customer I is assigned on situation *p* at salesman m
c_i: Completion time of service to customer *i*

In the fowling, non- liner integer programming formulations for the MTSP can be written as follows:

$$Min = Max\{c_1, \dots, c_j,\} \qquad (1)$$

S.t

$$\sum_{m=0}^{m}\sum_{p=1}^{p} x_{ipm} = 1 \qquad i = 1, \dots, n \qquad (2)$$

$$\sum_{i=0}^{n} X_{ipm} - \sum_{j=0}^{n} X_{j,p-1,m} \le 0 \qquad m = 1..m \;;\; p = 2..p \qquad (3)$$

$$\sum_{i=1}^{n} x_{ipm} \le 1 \qquad m = 1..m \;;\; p = 1..p \qquad (4)$$

$$pr_{ip} = \frac{\left(\left(\sum_{i=0}^{n} p_{im} X_{ipm}\right) * p^{-a}\right)}{w_i} \qquad i = 1, \dots, n;\; p = 1..p \qquad (5)$$

$$C_j - C_i + L\left(2 - x_{ipm} - X_{j,p-1,m}\right) \ge pr_{ip} + S_{ij} \qquad (6)$$

$$i = 1 \dots n), (i \ne j = 1..n), (m = 1..m), (p = 1..p)$$

$$C_i \ge \left(IS_i + pr_{ip}\right) * x_{i1m} \qquad i = 1, \dots, n \quad m = 1, \dots, m \qquad (7)$$

$$c_i \ge 0 \qquad (8)$$

$$\{0,1\} x_{ipm} \in \qquad (9)$$

Eq (1) Minimize the makespan. Constraint (2) Ensure that each client is visited on one of the existing position on the salesman tour. Constraint (3) ensures that until one position on a salesman's tour is empty, clients are not assigned to subsequent positions and clients assigned to empty positions on each salesman's tour, respectively. Constraint (4) guarantees that on each existing position, at most one job can be assigned. Constraint (5) measures actual processing time that if client j assign to the position j of salesman m. Constraint (5) measures the actual processing time that if client j assign to the position j of salesman m. Constraint set (6) ensures that the completion time of service to a client in sequence on a salesman's tour is at least equal to the sum of the completion time of the preceding clients service, the travel time between clients and the visit time of the present client. Constraint (7) measures the completion time for each service of each salesmen Constraint (8) defines the type of decision variables.

3. Solution approach

The multiple traveling salesman problem (MTSP) is a generalization of the well-known traveling salesman problem (TSP) (Ahmadvand et al., 2012) which is an obviously NP-hard problem (Evelyn et al., 2007). MTSP is more difficult than TSP, because it involves finding a set of Hamilton circuits without sub-tours for $m>1$ salesmen to serve a set of $n>m$ nodes so that each one is visited by exactly one salesman. The Because of the fact TSP belongs to the class of NP-hard problems (Russell, 1977), it is obvious that MTSP is an NP-hard problem. This means that the MTSP solution time grows exponentially increase in distribution points, thus exact algorithms are not capable of solving problems with large dimensions. Therefore, meta-heuristic are thought to be more efficient for complex MTSP and have become very popular among researchers .In this section, first proposed FA algorithm as name as meta-heuristic that is used throughout this paper.

3.1. Firefly Algorithm:

FA is one of the recent swarm intelligence methods developed by Yang in 2008 and is a kind of stochastic, nature-inspired, meta-heuristic algorithm that can be applied for solving the hardest optimization problems (also NP-hard problems). This algorithm belongs to stochastic algorithms. Swarm Intelligence (SI) belongs to an artificial intelligence discipline (AI) that became increasingly popular over the last decade .It is inspired from the collective behavior of social swarms of ants, termites, bees, and worms, flock of birds, and schools of fish. Although theses warms consist of relatively unsophisticated individuals, they exhibit coordinated behavior that directs the swarms to their desired goals. It is possible to formulate optimization algorithms because the flashing light can be formulated in such a way that it is associated with the objective function of problems considered, in order to obtain efficient optimal solutions. By idealizing some of the flashing characteristics of fireflies, the firefly algorithm was developed by Yang (2008).

FA uses the following three primary rules:
- All fireflies are unisex which means that they are attracted to other fireflies regardless of their sex.
- The degree of the attractiveness of a firefly is proportional to its brightness, thus for any two flashing fireflies, the less bright one will move toward the brighter one and their attractiveness will decrease as their distance increases. If there is no brighter one than a particular firefly, it will move randomly.
- The brightness or light intensity of a firefly is affected or determined by the landscape of the objective function to be optimized. For a maximization problem, the brightness can simply be proportional to the objective function.

3.2. Structure of the firefly algorithm:

In order to design FA properly, two important issues need to be defined: the variation of light intensity algorithms in such a manner that they are best suited to the demands of the problems to be solved. In and the formulation of attractiveness. These two issues enable developers to tailor different firefly the standard firefly algorithm, the light intensity I of a firefly representing the solution s is proportional to the value of fitness function $I(s) \propto f(s)$, while the light intensity $I(r)$ varies according to the following equation:

$I(r) = I_0 e^{-\gamma r2}$ (10)

Where I_0 denotes the light intensity of the source, and the light absorption is approximated using the fixed light absorption coefficient γ .The singularity at $r=0$ in the expression $I/r2$ is avoided by combining the effects of the inverse square law and an approximation of absorption in Gaussian form. The attractiveness β of fireflies is proportional to their light intensities $I(r)$. Therefore, a similar equation to Eq. (10) can be defined, in order to describe the attractiveness β

$\beta = \beta_0 e^{-\gamma r2}$ (11)

Where β_0 is the attractiveness at $r=0$. The light intensity I and attractiveness β are in some way synonymous. While the intensity is referred to as an absolute measure of emitted light by the firefly, the attractiveness is a relative measure of the light that should be seen in the eyes of the beholder sand judged by other fireflies .The distance between any two fireflies s_i and s_j is expressed as the Euclidean distance by the base firefly algorithm, as follows:

$$r_{ij}=||s_{i-}s_j||=\sqrt{\sum_{k=1}^{k=n}(s_{ik}-s_{jk})^2} \qquad (12)$$

Where n denotes the dimensionality of the problem. The movement of the i th firefly is attracted to another more attractive firefly j. In this manner, the following equation is applied:

$$s_i=s_i+\beta_0 e^{-\gamma r^2{}_{ij}}\left(s_j-s_i\right)+\alpha\varepsilon_i, \qquad (13)$$

Where ε_i is a random number drawn from Gaussian distribution. The movements of fireflies consist of three terms: the current position of i-th firefly, attraction to another more attractive firefly, and a random walk that consists of a randomization parameter α and the random generated number from interval [0, 1]. When $\beta_0 = 0$the movement depends on the random walk only .On the other hand, the parameter γ has a crucial impact on the convergence speed. Although the value of this parameter can theoretically capture any value from interval$\gamma \in [0,\infty)$, its setting depends on the problem to be optimized. Typically, it varies from 0.1 to 10. In summary, FA is controlled by three parameters: the randomization parameter α, the attractiveness β, and the absorption coefficient γ. According to the parameter setting, FA distinguishes two asymptotic behaviors. The former appears when $\gamma \to 0$ and the latter when $\gamma \to \infty$ .If$\gamma \to 0$, the attractiveness becomes $\beta = \beta_0$. That is, the attractiveness is constant anywhere within the search space. This behavior is a special case of particle swarm optimization (PSO). If$\gamma \to \infty$, the second term falls out from Eq. (13), and the firefly movement becomes a random walk, which is essentially a parallel version of simulated annealing. In fact, each implementation of FA can be between these two asymptotic behaviors.

3.3. Implementation of FA for MTSP:

Originally firefly algorithm designed to solve continuous optimization problems (Lukasik and Zak, 2010); Yang, 2008). However, the FA can be discretized to solve a permutation problem. In discrete problems, such as combinatorial, binary, and categorical, it is necessary to reduce the number of possible states to feasible solutions. This is done by the discretization of the continuous space by transforming the values into a limited number of possible states. There are several discretization methods one of them is random key.

The random key (RK) encoding scheme can be used to transform a position from a continuous space into an integer/combinatorial space (Chen et al., 2011; Li et al., 2010). To decode the position, the nodes are visited in ascending order for each dimension. For example, the continuous solution vector $\bar{x} = (0.9,0.2,0.03,0.35,0.5)$ can be decoded as$\bar{x} = (5,2,1,3,4)$.

4. Experimental design and analysis

4.1. data generation

In this section, performance of the proposed FA algorithms for MTSP is presented. The goal is to test the effectiveness of the proposed FA algorithm and comparative effectiveness between them. For comparing between Algorithms a set of test problem is needed. For producing our instance, proper values should be created. Each of models parameters, generate in the following Process time are generated from the discreet uniform distribution [10-20].

1. Setup time = S * Average Pi ; s= (0.5,1,1.5)
2. Salesman 's skill $\delta = (0.8,0.9,1,1.1,1.2)$
3. Level of quality each node:1,1.15,1.3)

According to model's parameters distribution created 70 problem in three size that show at table 1.

Table 1: model's parameters distribution

Size problem	Sales men	node
small	2	5
	2	6
	2	8
	3	5
	3	6
	3	8
medium	4	40
	4	50
	5	40
	5	5
	6	40
	6	50
large	7	70
	8	100

3.2. Parameter setting:

In this section, the results of the computational experiments are used to evaluate the performance of the FA algorithm for MTSP problem. There are five instances for each problem size. At this point, some information about parameter analysis would be useful. Initially, several experiments were conducted on test problems in order to determine the tendency for the values of parameters. In this pervious works, to determine the tendency for values of parameters FA, that show at Table 2 one test problems were used for this purpose. Test problems were made of 40 of nodes and 4 salesmen. The SN ratio plots are shown in Figure1, suggesting that the main factors including that (Max it* N pop , gamma, beta0 , alpha) best level for FA algorithm are(100*40,1,0.5,0.5) .

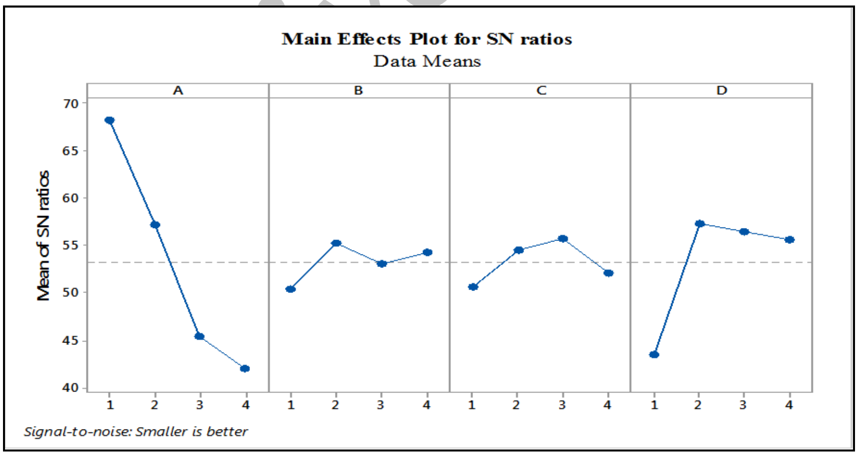

Figure 1: The SN ratio plot of FA

Table2: FA's parameters and their levels considered.

parameters	level1	level2	level3	level4
maxit* npop	100*40	80*50	50*80	40*100
gamma	0.1	1	5	10
beta0	0.05	0.1	0.5	1
alpha	0.1	0.5	0.9	0.99

3.3. Experimental results

The FA algorithms were developed using MATLAB 8.4 and run on a PC Pentium IV, 2.4 GHz speed with 6 GB of RAM. Sixteen test problems with the number of nodes varying from 4 to 100 are generated. There are five instances for each problem size. Name of producing test problems with variable parameters is given in Table3. Each test was repeated for seven runs due to each instance and calculated RPD for each algorithm in the following.
Thus, there were 420 runs in total. Where c* for problems with small size is lingo solution and for other problems with medium and large size is the best solution between the all replication of two algorithms. And c^A is objective value obtained by algorithm A.

$$RPD(A) = \frac{(c^A - c^*)}{c^*} * 100 \qquad (14)$$

The average of RPD for each of set problems are summarized in Table 4 Based on Eq. (14), lower value of ARPD is better than its higher value.

Table 3: Name of problems generated with variable parameters.

Number of client	**Number of Salesman**					
	2	3	4	5	7	8
5	521	531	-	-	-	-
	522	532	-	-	-	-
	523	533	-	-	-	-
	524	534	-	-	-	-
	525	535	-	-	-	-
6	621	631	-	-	-	-
	622	632	-	-	-	-
	623	633	-	-	-	-
	624	634	-	-	-	-
	625	635	-	-	-	-
8	821	831	-	-	-	-
	822	832	-	-	-	-
	823	833	-	-	-	-
	824	834	-	-	-	-
	825	835	-	-	-	-
40	-	-	4041	4051	-	-
	-	-	4042	4052	-	-
	-	-	4043	4053	-	-
	-	-	4044	4054	-	-
	-	-	4045	4055	-	-
50	-	-	5041	5041	-	-
	-	-	5042	5042	-	-
	-	-	5043	5043	-	-
	-	-	5044	5044	-	-
	-	-	5045	5045	-	-
70	-	-	-	-	7071	7081
	-	-	-	-	7072	7082
	-	-	-	-	7073	7083
	-	-	-	-	7074	7084
	-	-	-	-	7075	7085
100	-	-	-	-	10071	10081
	-	-	-	-	10072	10082
	-	-	-	-	10073	10083
	-	-	-	-	10074	10084
			-	-	10075	10085

Table 4: FA ARPD

name problem	521	522	523	524	525	531	532	533	534	535
FA Arpd	1%	4%	4%	4%	7%	14%	5%	5%	7%	4%
name problem	621	622	623	624	625	631	632	633	634	635
FA Arpd	5%	17%	6%	8%	4%	7%	4%	11%	17%	4%
name problem	821	822	823	824	825	831	832	833	834	835
FA Arpd	9%	8%	8%	9%	9%	4%	11%	21%	11%	18%
name problem	4041	4042	4043	4044	4045	4051	4052	4053	4054	4055
FA Arpd	9%	14%	12%	10%	9%	11%	11%	11%	10%	10%
name problem	5041	5042	5043	5044	5045	5051	5052	5053	5054	5055
FA Arpd	14%	13%	11%	14%	13%	12%	10%	12%	17%	12%
name problem	7071	7072	7073	7074	7075	7081	7082	7083	7084	7085
FA Arpd	15%	14%	13%	14%	18%	15%	16%	15%	15%	12%
name problem	10071	10072	10073	10074	10075	10081	10082	10083	10084	10085
FA Arpd	17%	17%	17%	17%	18%	17%	18%	17%	19%	18%

Table 5: Comparison of Objective Value's results of the model solved by Lingo 8 with the proposed FA

Small			Medium		Large	
Name Problem	**Best OF FA**	**LINGO Solution**	**Name Problem**	**Best OF FA**	**Name Problem**	**Best OF FA**
521	74.9	74.9	4041	181.52	7071	192.51
522	74.9	74.9	4042	200.2	7072	205.66
523	54.2	54.2	4043	200.12	7073	242.075
524	52.3	52.3	4044	178.54	7074	196.431
525	68.3	68.3	4045	179.93	7075	210.787
531	41.6	39.8	4051	173.39	7081	179.618
532	41	41	4052	133.79	7082	204.831
533	39.8	39.8	4053	147.31	7083	179.541
534	58.1	58.1	4054	162.07	7084	193.834
535	34.4	34.4	4055	151.03	7085	194.695
621	70.3	70.3	5041	226.8	10071	300.82
622	50.9	50.9	5042	235.4	10072	307.236
623	60.9	60.9	5043	246.49	10073	306.78
624	50.3	50.3	5044	258.89	10074	270.87
625	78.3	78.3	5045	234.64	10075	321.518
631	46.1	44.7	5051	228.25	10081	292.905
632	59.1	59.1	5052	181.27	10082	265.798
633	42.6	42.6	5053	219.35	10083	272.899
634	53.2	53.2	5054	199.91	10084	278.084
635	38.7	38.7	5055	203.82	10085	269.155
821	61.9	61.9				
822	49.8	49.8				
823	64.5	64.5				
824	62.8	62.8				
825	85	85				
831	62.2	62.2				
832	54.1	50.3				
833	54.1	54.1				
834	61.5	61.5				
835	77	66				

4. Conclusion

The Multiple Travelling Salesman Problem (MTSP) is one of the main scopes of optimization and is the generalization, of the famous travelling salesman problem. This paper presented a mathematical model for multiple travel salesman problem that minimize the makespan value with the aim of workload balancing between salesmen and time of servicing customers isn't regarded as traditional methods during a fixed time. In this paper, a position-based learning function has been used in which learning is influenced by the number of conducting servicing. So that, each salesman's skill is influenced by a learning effect function. The FA algorithms were planned to solve the mathematical model. Experimental design were carried out to find out the appropriate parameters setting of FA algorithms. This work can be further extended by balancing the workload of the salesmen with varying skill of salesman of different levels.

References:

1. Ahmadvand, M., Yousefi. Khoshbakht, M. and Mahmoodi . Darani, N. .2012, Solving the Traveling Salesman Problem by an Efficient Hybrid Metaheuristic Algorithm, Journal of Advances in Computer Research, 3(3), 75-84
2. Bektas_, T., 2006. The multiple traveling salesman problem : An overview of formulations and solution procedures. Omega: The International Journal ofManagement Science 34, 209–219.
3. Chen, H., Li, S., Tang, Z. 2011. Hybrid gravitational search algorithm with random-key encoding scheme combined with simulated annealing. Int. J. Comput. Sci. Net. Secur.11 (6), 208.217.
4. Evelyn, C. Brown, C. T. Ragsdale, A. Carter, A. E. 2007,A grouping genetic algorithm for the multiple traveling salesperson problem, International Journal of Information Technology & Decision Making, 6(2), 333–347.
5. Lacomme & C.Prins & C.Prodhon & L.Rena ,A Multi Start Split based Path Relinking(MSSPR)approach for the vehicle routing problem with route balancing P. Engineering Applications of Artificial Intelligence38(2015)237–251
6. Li, X., Wang, J., Zhou, J., Yin, M. 2010, An effective GSA based memetic algorithm for permutation flow shop scheduling. In: IEEE Congress on Evolutionary Computation, pp. 1.6.
7. Lukasik, S., ˙ Zak, S . 2010, Firefly algorithm for continuous constrained optimization tasks. Syst. Res. Inst. Pol. Acad. Sci.1-10.
8. R. Ribeiro, H.R. Lourenc_o, A multi-objective model for a multi-period distribution management problem, in: MIC'2001, 2001, pp. 97–102.
9. Russell, R. A. 1977,An effective heuristic for the m-tour traveling salesman problem with some side conditions. Operations Research, 25(3), 517–524
10. T. Lee, J. Ueng, A study of vehicle routing problems with load balancing, International Journal of Physical Distribution and Logistics Management 29 (1999) 646–658.
11. X.S. Yang,2008,Firefly algorithm,Nature-InspiredMetaheuristicAlgorithms20(2008)79–90.