Advanced Logistic Systems, Vol. 7, No. 2 (2013), pp. 95–102.

PRELIMINARY STUDIES ON THE FIXED DESTINATION MMTSP SOLVED BY DISCRETE FIREFLY ALGORITHM

LÁSZLÓ KOTA[1]–KÁROLY JÁRMAI[2]

Abstract: The fixed destination MmTSP (multi-depot multiple travelling salesman problem) is an np hard problem, which can't be solved in polynomial time. Against the traditional TSP problem here there are more travelling salesmen which seek out the cities. Every city is visited by once by any of the salesmen and after the round route the salesman go back to its home location. But unlike at the TSP the salesman does not start from the same location. The firefly algorithm is a member of the swarm optimizations family. Originally it was developed to solve continuous state space problems but with discretization it is capable to solve combinatorial problems also. In this article we will show a potential discretization variant. In the firefly algorithm every firefly represent a solution. In our algorithm the salesmen using a multi chromosome model, where there are a separate list for every salesman for the cities to visit.

Keywords: supplier selection, optimization, firefly algorithm, MS Excel solver

1. The problem

The problem of the MmTSP has an important role in logistics. The problem is the generalization of the TSP problem [1], in the MTSP there are more salesmen instead of one which are visiting the cities starting from the same location. The agents starting from the same location is the MmTSP problem. MmTSP means multi depot multiple salesman problem, which represents more complexity compared to the original problem. We can differentiate fixed destination and non-fixed destination cases. In case of fixed destination problem the salesman have to return to the same location where he started. In case of non-fixed destination problem this constraint is not present; the salesmen can arrive at any locations [2]. The fixed destination case suits to a huge amount of logistic problems like the large scale technical inspection and maintenance systems, like the elevator maintenance networks [3] or water quality monitoring systems or natural gas transfer station networks.

2. Literature

The pure MSTP problem is widely discussed in the literature, the range of the solution methods and the application is extensive [4][5][6]. But the literature of the special cases are small sometimes even not exists. The authors found a solution of this problem which uses Ant Colony Algorithm [7]. In our former articles we show the optimization problems of large scale technical inspection and maintenance systems, where we solved the general fixed destination multiple depot multiple salesman problem with multiple tours with the very special constraints of these type of systems, there are no known another solution so far [8][9]. In this early stage of the research we examine and judge the usability of the firefly

[1] PhD, University of Miskolc
altkota@uni-miskolc.hu
[2] DSc, University of Miskolc
altjar@uni-miskolc.hu
H-3515 Miskolc-Egyetemváros, Hungary

algorithm to solve this type of problems, in this article the multiple tour model is not examined yet.

3. FireFly algorithm

The firefly algorithm developed by Xin-She Yang [10]. The effectivity of the algorithm can be compared to the newest metaheuristic algorithm like the harmony search [10], or the PSO based [12] new algorithms. The fireflies attract the other fireflies with light signals. The artificial fireflies defined in the algorithm are:

- unisexual: one firefly will attract all the other fireflies,
- attractiveness is proportional to their brightness, and for any two fireflies, the less brighter one will be attracted by the brighter one.
- If there are no fireflies brighter than a given firefly, it will move randomly.
- The brightness of the fireflies based on the target function [1].

The pseudo code of the firefly algorithm:

```
1.  target function: f(x); X=($x_1$, $x_2$, …. $x_d$)
2.  generate an initial population of fireflies: $x_i$, (i=1….n)
3.  Formulate light intensity (I) so that it is associated with I=f(x)
4.  define absorption coefficient: $\gamma$
while (t < maxgeneration)
    for i=1:n (all fireflies)
            for j = 1:n (all fireflies)
                    if ($I_j$ > $I_i$)
                            move firefly i towards j
                    endif
            define attractiveness based on the (r) distance exp(-$\gamma$·r)
            evaluate new solutions and update light intensity
            end for
    end for
    find the best firefly
    end while
```

The absorption coefficient (γ) defines how much the attractiveness is decreased by the range, if $g \to 0$, then the algorithm corresponds to the normal PSO [12] (Particle Swarm Optimization) algorithm.

The movement of the firefly describes mostly by the

$$x_{i+1} = x_i + \beta_o e^{-\gamma r_{ij}^2}\left(x_j - x_i\right) + \alpha(rand() - \frac{1}{2}) \tag{1}$$

formula or by the

$$\beta = \beta_0 \cdot e^{-\gamma r}$$
$$x_{i+1} = x_i \cdot (1 - \beta \quad) + x_i \cdot \beta \quad + \alpha(rand() - \frac{1}{2}) \tag{2}$$

formula which is equivalent.

The firefly algorithm was developed to solve continuous problems, but the algorithm can be discretized so it can be used to solve non continuous permutation problems too [13].

4. A simple mathematical model

There are *n* salesmen and *m* cities. The location of the salesmen (S) and the cities (C) are defined by its coordinates:

$$S_i = \{x_i, y_i\} \\ C_i = \{x_i, y_i\}. \tag{3}$$

The main output variable of the optimization is the assignment matrix:

$$Y_{ij} = [y_{ij}], \tag{4}$$

where

- $y = \begin{cases} 1 \\ 0 \end{cases}$ in case of 1 the salesman *i* is assigned to the city *j* so he visited it on its tour.

We used the Euclidean distance:

$$D(S_i, C_j) = \sqrt{\left(x_i - x_j\right)^2 + (y_i - y_j)^2} \tag{5}$$

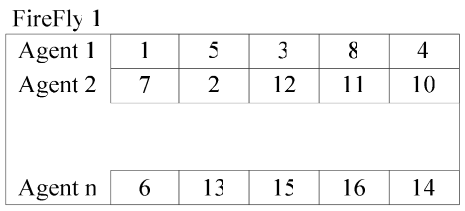
The target function is the optimization is minimal route length of the sum of the routes:

$$C = \sum_{i=1}^{n} \left(D(S_i, C_1) + \sum_{j=1}^{m-1} \left(D(C_j, C_{j+1}) \right) + D(C_m, S_i) \right) \tag{6}$$

for the cities where $Y_{ij} <> 0$.

5. The model

The problem is solved by the discretization of the firefly algorithm, where one firefly represents one solution of the problem (Figure 1).



FireFly 1

Agent 1	1	5	3	8	4
Agent 2	7	2	12	11	10
Agent n	6	13	15	16	14

Figure 1. One firefly represents one solution

The initial population of the fireflies is generated randomly thus the fireflies scattered in the state space. The distance between two fireflies is defined by the swaps between them.

This means the number of swaps has to be performed on the first permutation to get to the second permutation (Figure 2).

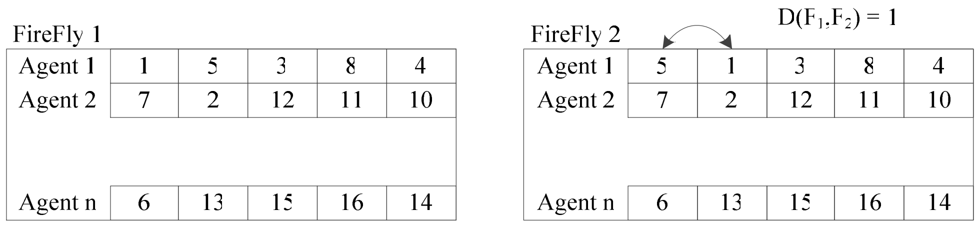

D(F$_1$,F$_2$) = 1
FireFly 1
Agent 1 | 1 | 5 | 3 | 8 | 4
Agent 2 | 7 | 2 | 12 | 11 | 10
Agent n | 6 | 13 | 15 | 16 | 14
FireFly 2
Agent 1 | 5 | 1 | 3 | 8 | 4
Agent 2 | 7 | 2 | 12 | 11 | 10
Agent n | 6 | 13 | 15 | 16 | 14

Figure 2. Distance of two fireflies

In the algorithm the fireflies move toward the brightest firefly. In our case the brightest where the target function is minimal because this problem is a minimization problem. The brightest firefly or fireflies move randomly:

$$M(F_i) = random(1, d(F_i, F_j)). \tag{7}$$

The random movement in the discrete state space is defined by the swap of the cities of the salesmen. In fact the swap of the cities creates new permutations. The new permutations are created by similar functions like the operator function we used before, because the movement in a large dimension state space cannot be defined like the movement in a several (mostly three) dimension continuous state space. However the random movement operators can use special characteristics of the problem, like city swap, rotate.

5.1. Random movement. It became obvious during the development of the algorithm that the firefly algorithm can easily fall in local optimum in the large multi-dimensional state spaces. So we had to find a method which provides avoiding the stuck in local optima, thru providing high degree of change of the actual permutation. The random operators operate similarly like the evolutionary algorithm mutation operators [9] we developed before. In this case the route length of the salesman is not changed, so the salesman has the same number of cities. The operators must not shorten the chromosome of the salesman and during city swap operations always the same amount of cities can be swapped.

Local movement operators:

- Node move:
  A randomly selected city moved to a randomly selected location at a randomly selected salesman (Figure 3).

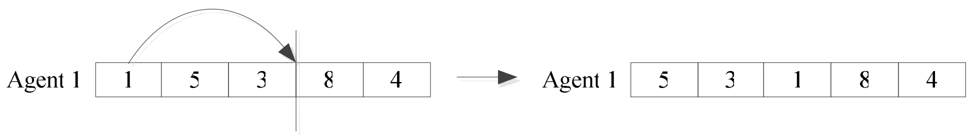


Figure 3. Local node move

- Node swap:
  Two randomly selected nodes are swapped at a randomly selected salesman (Figure 4).

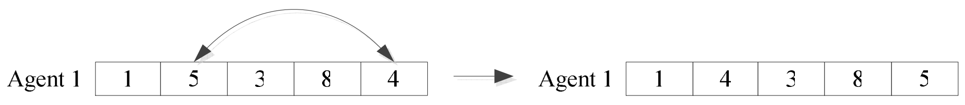


Figure 4. Local node swap

- Node sequence turning:
  A randomly chosen node sequence order is swapped (Figure 5).

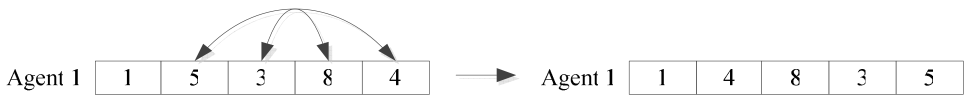


Figure 5. Local sequence order swap

Global movement operators:

- Node swap:
  Two randomly selected nodes are swapped between two randomly selected salesmen (Figure 6).

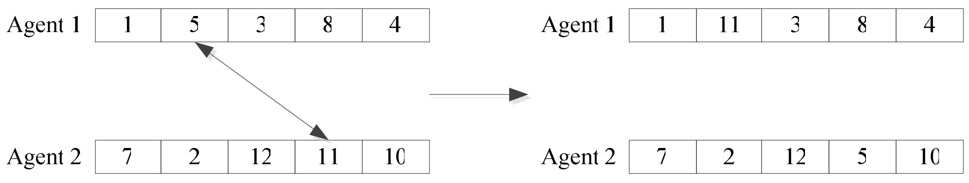


Figure 6. Global node swap

- Node sequence swap:
  Two randomly selected node vectors, by the same length, are swapped between two randomly selected salesmen (Figure 7).

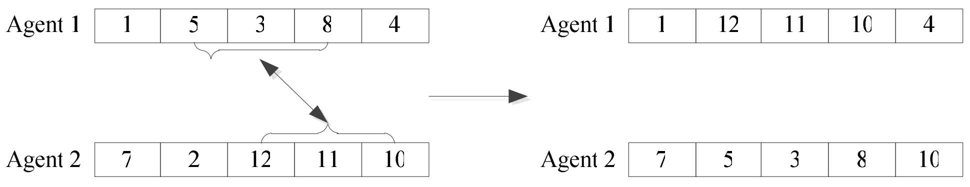


Figure 7. Global node sequence swap

- Rotation:
  The rotation affects all the salesmen. The all nodes shifted to the right by one. The last node of the last salesman shifted to the first node of the first salesman. The last node of a salesman is shifted to the first node of the next salesman (Figure 8).

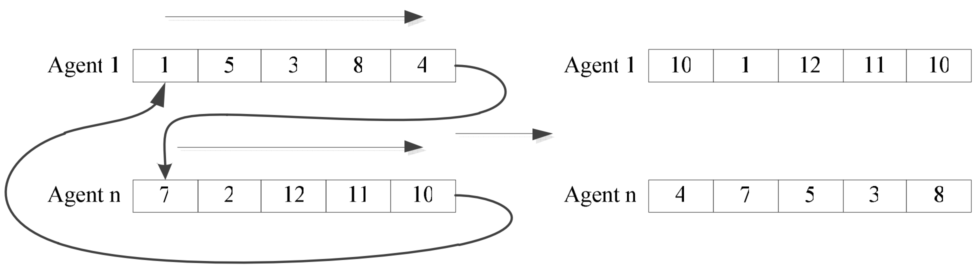
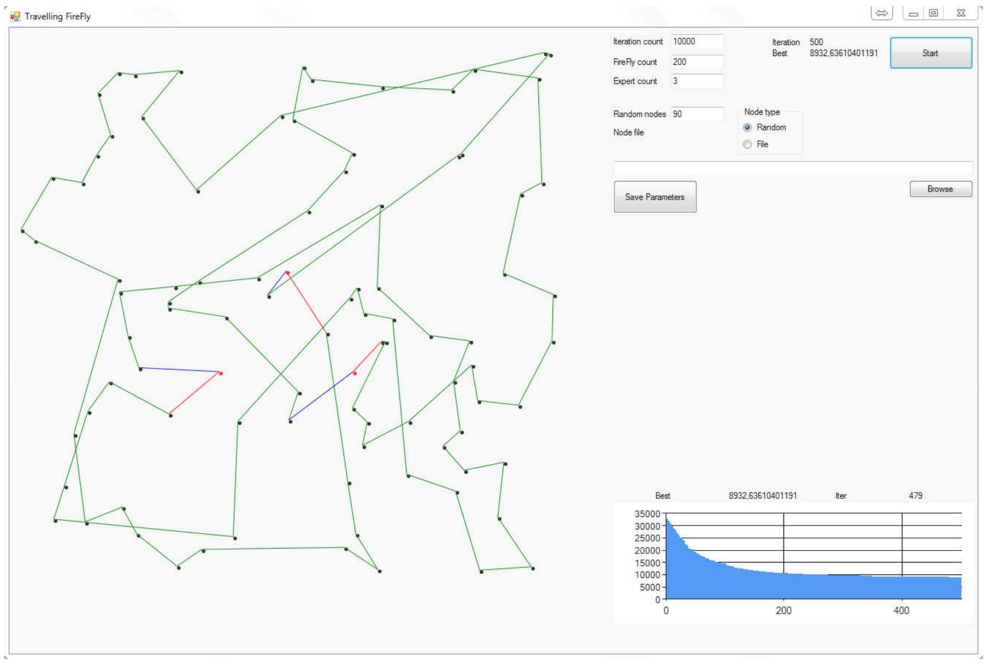


Figure 8. Rotation

6. The solution

A C# application was developed to test the goodness of the algorithm (Figure 9). It was performed well on the test instance and on the random generated instances. The algorithm avoid to stuck on local optimum on the test instances, on the random generated instances it cannot be proven.


Travelling FireFly

Iteration count 10000
FireFly count 200
Expert count 3
Iteration 500
Best 8932,63610401191

Random nodes 90
Node file
Node type
[✓ Random] File

Best 8932,63610401191 Iter 479

Figure 9. Optimization software

The convergence of the algorithm was very fast (Figure 10). During the development as the algorithm stuck on local optimum new random movement operators was introduced. The movement operators shown in this article resulted by numerous test, but there could be new operators implemented. At large scale problems as the dimension of the state space increasing there could be need for operators provide greater permutation distances.

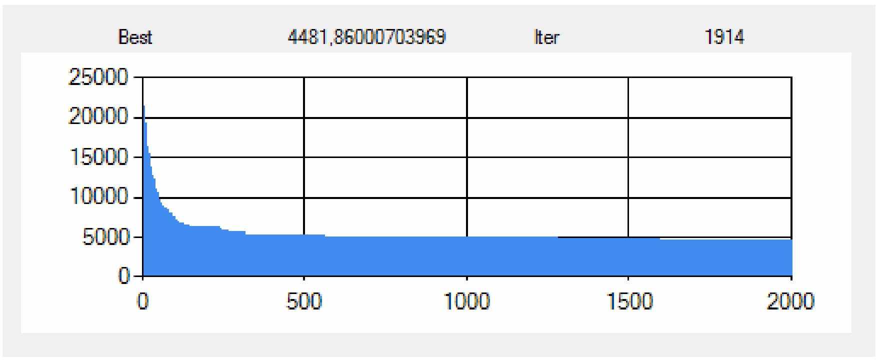


Figure 10. Convergence

7. Further researches

The developed algorithm will be further developed towards the optimization of the large scale technical inspection and maintenance systems what we introduced in [9]. This area requires a lot of special constraints as the minimum and maximum capacity of the experts; these are the salesman in this model. In that problem there are multiple routes performed by an expert so the extensive penalty functions have to be introduced we developed there and the algorithm has to be capable to optimize such large scale problems so the parallelization of the algorithm is also needs a research.

Acknowledgements

The research was supported by the TÁMOP 4.2.4.A/2-11-1-2012-0001 priority project entitled “National Excellence Program – Development and operation of domestic personnel support system for students and researchers, implemented within the framework of a convergence program, supported by the European Union, co-financed by the European Social Fund. The research was supported also by the Hungarian Scientific Research Fund OTKA T 75678 and T 109860 projects and was partially carried out in the framework of the Center of Excellence of Innovative Engineering Design and Technologies at the University of Miskolc.

Literature

[1] Diaby, M. (2010): *Linear Programming Formulation of the Multi-Depot Multiple Traveling Salesman Problem with Differentiated Travel Costs*. Traveling Salesman Problem, Theory and Applications, ed.: Prof. Donald Davendra, pp. 257–282.
[2] Kara, I.–Bektas, T. (2006): *Integer linear programming formulations of multiple salesman problems and its variations*. European Journal of Operational Research, Vol. 174, No. 3, pp. 1449–1458.
[3] Levitt, J. (2009): *The handbook of maintenance management*. Industrial Press Inc, p. 477.
[4] Pang, S.–Li, T.–Dai, F.–Yu, M. (2013): *Particle swarm optimization algorithm for multi-salesman problem with time and capacity constraints*. Applied Mathematics and Information Sciences, Vol. 7, (6), pp. 2439–2444. doi: 10.12785/amis/070637
[5] Bektas, T. (2013): *Balancing tour durations in routing a vehicle fleet*. Proceedings of the 2013 IEEE Symposium on Computational Intelligence in Production and Logistics Systems, pp. 9–16. doi: 10.1109/CIPLS.2013.6595194

[6] Kivelevitch, E.–Cohen, K.–Kumar, M. (2013): *A market-based solution to the multiple traveling salesmen problem*. Journal of Intelligent and Robotic Systems: Theory and Applications, Vol. 72 (1), pp. 21–40, doi: 10.1007/s10846-012-9805-3

[7] Ghafurian, S.–Javadian, N. (2011): *An ant colony algorithm for solving fixed destination multi-depot multiple traveling salesmen problems*. Applied Soft Computing, Vol. 11, pp. 1256–1262. doi: 10.1016/j.asoc.2010.03.002

[8] Kota, L. (2011): *Optimisation of Large Scale Maintenance Networks with Evolutionary Programming*. DAAAM International Scientific Book, pp. 495–512. Chapter 40., doi: 10.2507/daaam.scibook.2011.40

[9] Kota, L.–Jármai, K. (2013): *Efficient algorithms for optimization of objects and systems*. Pollack Periodica (under publication).

[10] Yang, X. S. (2008): *Nature-Inspired Metaheuristic Algorithms*. Luniver Press p. 128

[11] Bányai, T. (2011): *Optimisation of a Multi-Product Green Supply Chain Model with Harmony Search*. DAAAM International Scientific Book, 2011, pp. 15–30. doi: 10.2507/daaam. scibook.2011.02

[12] Farkas, J.–Jármai, K. (2008): *Design and Optimization of Metal Structures*. Horwood Publishing Limited, p. 300.

[13] Kusuma, G.–Suyanto, J. (2011): *Evolutionary discrete firefly algorithm for travelling salesman problem*. ICAIS'11 Proceedings of the Second international conference on Adaptive and intelligent systems, Springer-Verlag Berlin, Heidelberg, pp. 393–403.

[14] Sayadia, M. K.–Ramezaniana, R.–Ghaffari-Nasaba, N. (2010): *A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems*. International Journal of Industrial Engineering Computations, Vol. 1, pp. 1–10. doi: 10.5267/j.ijiec.2010.01.001