DISCRETIZATION OF THE FIREFLY ALGORITHM FOR THE TRAVELLING SALESMAN PROBLEM

László Kota[1], Károly Jármai[2]
[1]PhD, research assistant, [2]DSc, Dr. *habil* university professor

University of Miskolc, Insitute of Logistics

INTRODUCTION

The aim of this research is developing a discrete firefly optimization algorithm to solve the multiple travelling salesmen problem (MTSP) and later the fixed destination multiple route multiple travelling salesmen problem. In this article we will introduce the first steps of this research. We will introduce one proposed discretization of the firefly algorithm at this special problem, which heavily based on the characteristic of the problem and state space.

FIREFLY ALGORITHM

The firefly algorithm developed by Xin-She Yang [1.]. The effectivity of the algorithm can be compared to the newest metaheuristic algorithm like the harmony search [2.], or the PSO based [3.] 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 [4.].

The pseudo code of the firefly algorithm:
1. target function: f(x); $X=(x_1, x_2, \ldots x_d)$
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: γ

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(-γ·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 $\gamma \to 0$, then the algorithm corresponds to the normal PSO [5.] (Particle Swarm Optimization) algorithm.

The movement of the firefly describes mostly by the

$$x_i = x_i + \beta_o e^{-\gamma r_{ij}^2}(x_j - x_j) + \alpha(rand - \tfrac{1}{2}), \tag{1}$$

formula or by the

$$\beta = \beta_0 \cdot e^{-\gamma r}$$
$$x_i = x_i \cdot (1 - \beta\ ) + x_i \cdot \beta\ + \alpha(rand - \tfrac{1}{2}), \tag{2}$$

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 [6.].

THE MODEL

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

FireFly 1

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

Fig. 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 (Fig. 2.).

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   D(F$_1$,F$_2$) = 1

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

Fig. 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:

$$M(F_i) = random(1, d(F_i, F_j)) \tag{3}$$

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.

MOVING THE FIREFLIES

The normal movement of the fireflies based on the distance of the fireflies. The tested movement algorithm is the following:

1. Checking the chromosome lengths of the fireflies
   a. if a chromosome length is bigger the first gene will be moved from the chromosome of the next expert to the end of this chromosome
   b. if a chromosome length is lesser the last gene will be moved from the end of the chromosome to the first location of chromosome of the next expert.
   Summarizing this process; one gene will be shifted to forward or back depending on the difference in the length of the chromosomes.
2. If there are no length difference in any chromosome the algorithm checking the genes one by one in the two fireflies
   a. if any genes differ, that gene will be swapped
      i. the algorithm search the gene of the second firefly in the first firefly
      ii. and the gene swapped with that exact gene.
3. If there are no differences no gene swap happen.
4. If the firefly is not moved during this process it is flagged for the random movement process.

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 tuck in local optima, thru providing high degree of change of the actual permutation. The random operators operate similarly like the evolutionary algorithm mutation operators [7.] 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 (Fig. 3.).
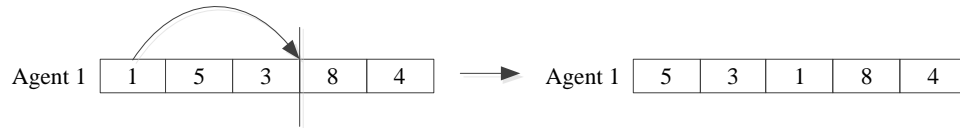
Fig. 3.
Local node move

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

Fig. 4.
Local node swap

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

Fig. 5.
Local sequence order swap

Global movement operators:
- Node swap:
  Two randomly selected nodes are swapped between two randomly selected salesmen (Fig. 6.).

Fig. 6.
Global node swap

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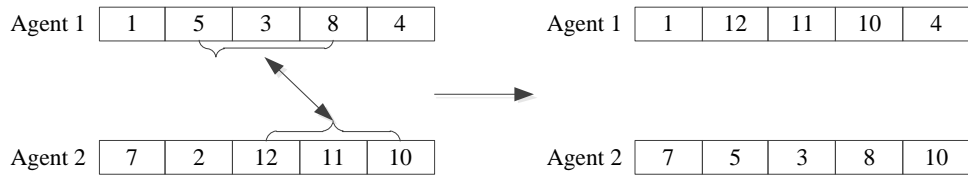
Fig. 7.
Global node sequence swap

- 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 (Fig. 8.).
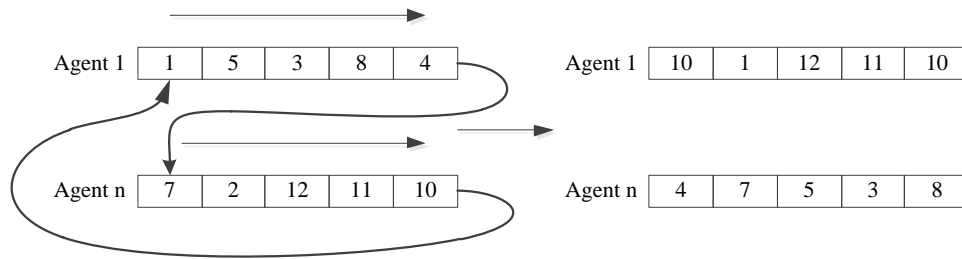

Fig. 8.
Rotation

TEST RESULTS

The first test instance is a small test instance with 3 salesmen and 50 nodes in circular position, with 200 fireflies (Fig. 9., Table 1.).
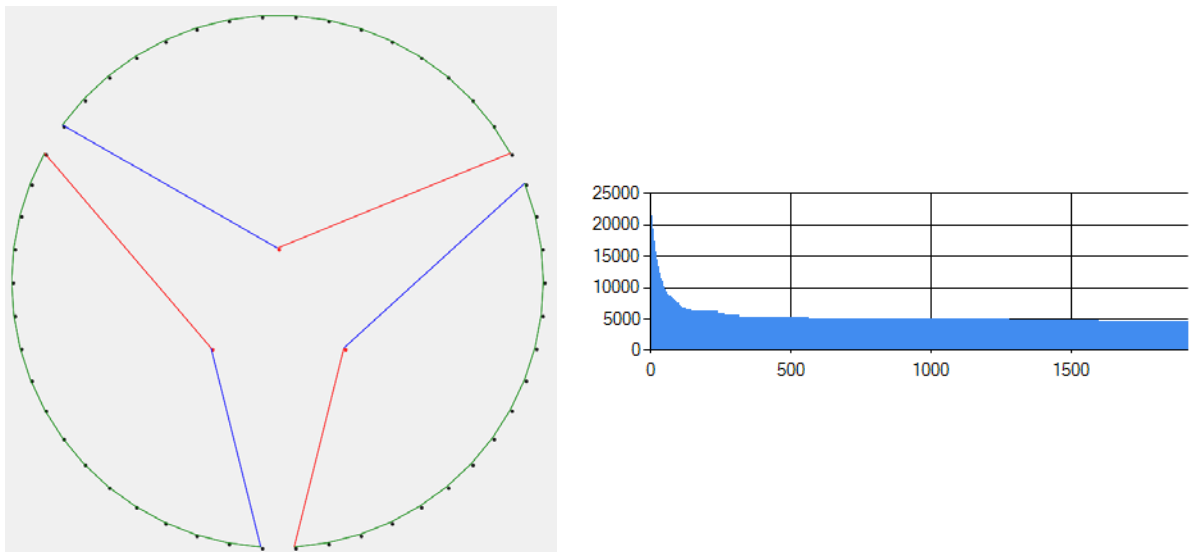

Fig. 9.
First test instance and the convergence function

Iteration count	Elapsed time	Target function
1914	29 s	4481,86

The second test instance is also a small test instance with 3 salesmen and random positioned 50 nodes, with 200 fireflies (Fig. 10., Table 2.):
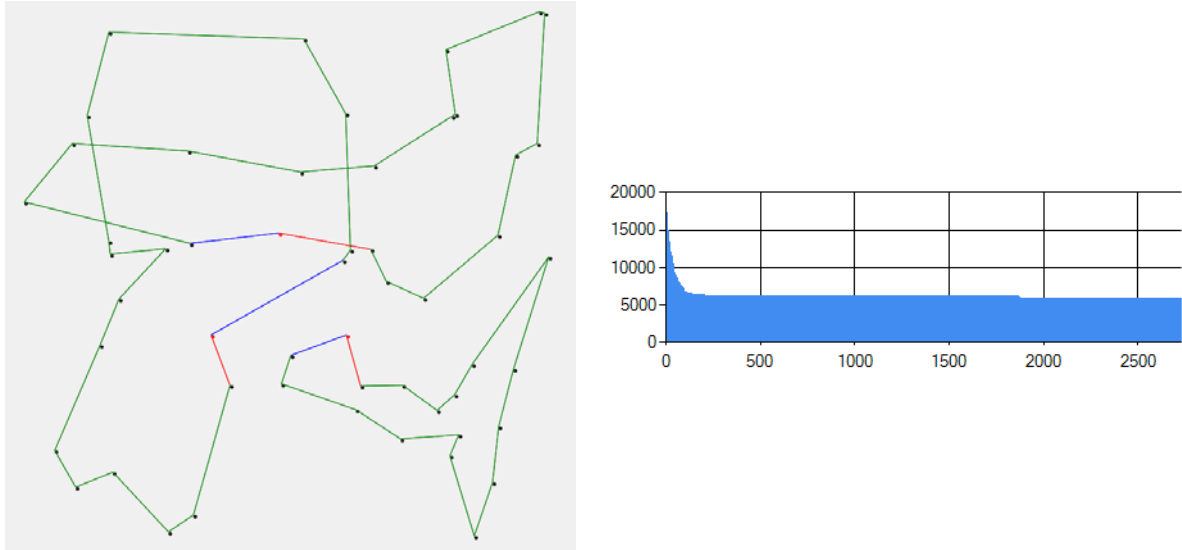

Fig. 10.
First test instance and the convergence function

Table 2.
Results of test run

Iteration count	Elapsed time	Target function
2730	47 s	5747,52

The third test instance is a medium sized test instance with 3 salesmen and random 90 nodes, with 200 fireflies (Fig. 11., Table 3.).

Table 3.
Results of test run

Iteration count	Elapsed time	Target function
7258	3 min 37 s	7549,16

Fig. 11.
First test instance and the convergence function

SUMMARY

The developed algorithm performed well in the test cases. It has good and fast convergence with a reasonable amount of running time. During the tests we saw that the algorithm needs a little tuning of the local and global search ratio. This can be done with applying some adaptive method or using simulated annealing technique which means applying more global search in the beginning of the run and then progressively using more and more local search via decreasing the movement factor.

FURTHER RESEARCHES

However, the main target is to develop this discrete firefly method to be able to handle the problem of large scale maintenance networks mentioned in [7.]. Handling that problem we have to improve this algorithm with a lot of additional capability like:
- the capability of optimizing the number of the salesmen
  - so the algorithm have to be able to move non equal gene sequences,
- we have to apply large number of penalty functions to take the various constraints into account, like the
  - number of salesmen
  - the number of nodes which is visited by one salesman
    - upper and lower limit also,
  - the length of the routes is done by one salesman in one cycle
  - cycle count

This last two constraints will introduce the "multi route" characteristic into this model, so using them it will be no more simple MTSP but it will be the fixed destination multiple travelling salesman problem with multiple routes (mMmTSP).

ACKNOWLEDGEMENTS

LITERATURE

[1.] Yang, X. S. (2008). **Nature-Inspired Metaheuristic Algorithms**, Luniver Press 2008, p. 128, ISBN 1-905986-10-6

[2.] T. Bányai: **Optimisation of multi level supply chain of automatised production systems with harmony search algorithm**, Proceedings of the II. European conference on logistics, pp.: 65-71., ISBN 978-83-61118-67-1

[3.] József Farkas, Károly Jármai: **Design and Optimization of Metal Structures**, Horwood Publishing Limited 2008, ISBN 978-1-904275-29-9, 300 p.

[4.] 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, ISBN: 978-953-307-426-9, pp.: 257-282

[5.] 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

[6.] Kusuma G., Suyanto J.: **Evolutionary discrete firefly algorithm for travelling salesman problem**, ICAIS'11 Proceedings of the Second international conference on Adaptive and intelligent systems, Springer-Verlag Berlin

[7.] Kota, L. (2011): **Optimisation of Large Scale Maintenance Networks with Evolutionary Programming**, DAAAM International Scientific Book, ISSN 1726-9687, ISBN 978-3-901509-84-1, pp.: 495-512, Chapter 40., doi: 10.2507/daaam.scibook.2011.40