Advanced Logistic Systems – Theory and Practice, 13(2), 21-28, https://doi.org/10.32971/als.2020.002

APPLICATION OF A MULTILEVEL FIREFLY ALGORITHM ON A LARGE VARIABLE NUMBER LOGISTIC PROBLEM

LÁSZLÓ KOTA[1]-KÁROLY JÁRMAI[2]

Abstract: During our research and industrial projects, we often meet difficult optimization problems, a lot of variables, a lot of constraints, nonlinear and mostly discrete problems, where the running time can be calculated sometimes in weeks with the usual optimization methods on an average computer. In the most cases in the logistic industry the strongest constraint is the time. The optimizations are running on a usual office configuration and the company accepts the suboptimal solution what the optimization method gives in the appropriate time limit. In this article we will investigate a multilevel method on supply chain problem, to increase the effectivity, improve the solution in a strict time condition.
Keywords: optimization, heuristics, multilevel algorithm, firefly algorithm

1. INTRODUCTION

The aim of this research was to improve our solution on a large variable count logistic problem which we solved with firefly algorithm. It is a large problem with 130 decision variables, the basic was to combine a fast, global search and a slower but more accurate local search algorithm. For the fast, global search algorithm we choose a simple random search algorithm which can generate solution very fast and for the local search the firefly algorithm which is commonly used to solve optimization problems. The main question was can we improve the solution in a limited time frame?

2. LITERATURE

The firefly algorithm is a swarm based heuristic algorithm presented by Xin-She Yang [1], inspired by the mating behavior of fireflies. It has an extensive literature since its appearance. The algorithm is shown on Figure 1.

The firefly algorithm based on three rules:

- All fireflies are attracted by each other.
- Attractiveness is proportioned by brightness, the less bright move towards the brighter one.
- If there is no brighter firefly than the selected one, that will move randomly in the state space.

The control parameters of the firefly algorithm are the absorption coefficient, the randomization control factor, and the firefly population size. The values of these control parameters greatly affect the quality of the achieved solution and the efficiency of the algorithm. It is problem dependent to select suitable control parameters for the actual

[1] assistant professor, University of Miskolc, Hungary
altkota@uni-miskolc.hu
[2] professor, University of Miskolc, Hungary
jarmai@uni-miskolc.hu

algorithm. Hard to deal with complex problems with many local optima where the most algorithms are trapped.

Objective function $f(\boldsymbol{x})$, $\quad \boldsymbol{x} = (x_1, ..., x_d)^T$
Generate initial population of fireflies $\boldsymbol{x}_i$ $(i = 1, 2, ..., n)$
Light intensity I_i *at* $\boldsymbol{x}_i$ *is determined by* $f(\boldsymbol{x}_i)$
Define light absorption coefficient γ
while *(t <MaxGeneration)*
for $i = 1 : n$ *all* n *fireflies*
  for $j = 1 : n$ *all* n *fireflies (inner loop)*
  if $(I_i < I_j)$, *Move firefly* i *towards* j; **end if**
    Vary attractiveness with distance r *via* $\exp[-\gamma r]$
    Evaluate new solutions and update light intensity
  end for j
end for i
Rank the fireflies and find the current global best $\boldsymbol{g}_*$
end while
Postprocess results and visualization

Figure 1. Pseudo code of the firefly algorithm [1]

Although it is highly important, there is no consistent methodology for determining the control parameters of the applied firefly algorithm variant. Mostly, the parameters are fixed throughout a lot of experiments or set arbitrarily within some predefined ranges [2].

The two stage algorithms also have a significant literature. The importance of the two- and multi-phase algorithms are increasing as the complexity of the problems to be solved and we wanted to solve increases. Figure 2 shows the number of articles using the "two stage heuristic algorithms" keywords in the search engine of the ScienceDirect (https://www.sciencedirect.com) are increasing year by year and the time of the publication of this article, the year 2020 is not even closed.

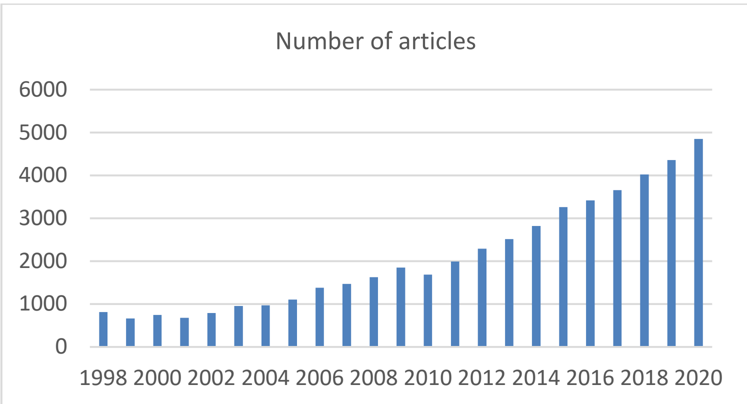

Figure 2. The number of articles found with the "two stage heuristic algorithms" keywords in ScienceDirect

2. OPTIMIZATION PROBLEM

The problem we wanted to optimize is a supply chain optimization problem based on the olive oil production by the company Tariş in Turkey described in [3]. In short: There is an olive oil bottling plant in Izmir. The olive oil supplier of the company collects the olive oil from all over Turkey from the 5 major 5 major olive oil producing region. For the production there are also needed glass bottles and tin cans in different sizes and other packaging materials. These came from different suppliers and of course each bottle and each packaging unit has a different price. The company has a warehouse on site which stores the finished products ready for shipping. The finished products are shipped to the 13 major economic regions in Turkey (Figure 3).

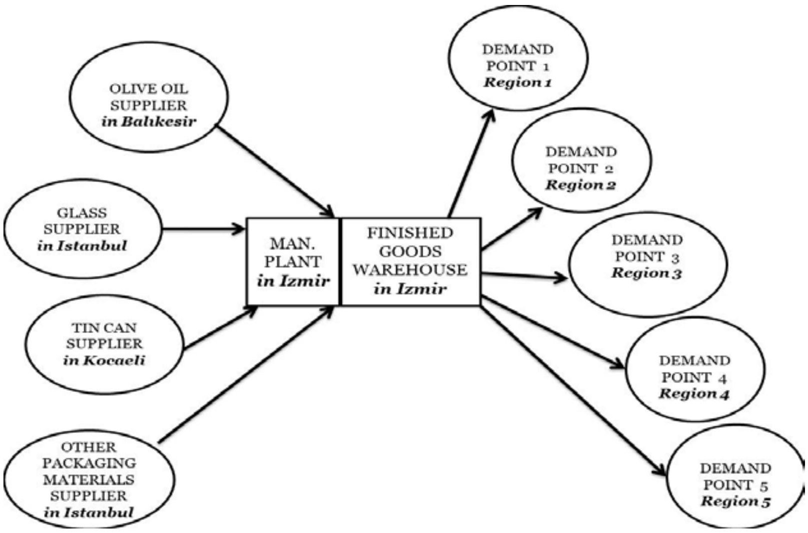

Figure 3. The model of the olive oil bottling supply chain [3]

The whole problem cannot be described within the framework of this article, so we just present the core function (1) needed to be optimized here. The main objective function is the profit maximization, as in many other cases:

$$\sum_{i=1}^{I}\sum_{k=1}^{K}(p_i - c_{ik})Y_{ik} - \sum_{i=1}^{I}(oc * oil_i - pc_i)X_i - t * toil \tag{1}$$

where:

p_i: product price
c_{ik}: transportation cost
Y_{ik}: produced quantity
oc: oil cost (1liter)
oil_i: required oil quantity
pc_i: packaging cost
X: total produced quantity of the given packaging unit
$t * toil$: transportation cost of the oil used

and the constraints:

$$X_i \geq \sum_{k=1}^{K} Y_{ik} \; \forall i \tag{2}$$

$$\sum_{i=1}^{I} oil_i X_i \leq totaloil \tag{3}$$

$$Y_{ik} \leq d_{ik} \; \forall i,k \tag{4}$$

$$X_i \in Z \forall \; i \tag{5}$$

$$Y_{ik} \in Z \forall \; i,k \tag{6}$$

The total produced quantity must less than the total required quantity (1), the oil used must less than the total oil available (2), the produced quantity for each region must be less than the required quantity of the given region(3) and the variables must be integers (5)(6).

In the model we have 10 packaging units (*p*), and 13 regions (*r*), so the matrix (*Y*) to be optimized, which is the produced quantity matrix, will contain 130 variables.

3. Distance Metric and Movement

We often use swarm methods in our research works like the PSO (Particle Swarm Optimization) [4] or the Firefly Algorithm [5], both are common, widely used methods to solve problems, but used mostly in continuous problems. We did not use these methods on such many variables until now. The firefly algorithm is working well with various test functions, a lot of general problems [6], but how it performs on large variable count problem.

One of the biggest problem was the discrete nature of the problem. The firefly algorithm originally was developed to optimize continuous problems and where the directions can be defined so the “moving toward something” have meaning, so the fireflies can move toward each other. So, first we had to define the distance of the fireflies and define a movement function over a matrix. The distance of two fireflies (F1, F2) is:

$$dst(F1,F2) = \sum_{\substack{i=1..p \\ j=1..r}} Abs(y_{ik}^{F1} - y_{ik}^{F2}) \tag{7}$$

where:

$F1 \; and \; F2$: the two fireflies whose distance we want to define

y_{ik}: decision variable, manufactured quantity of the packaging unit i, in the region k

The movement function:

$$movetoward(F1,F2){:}\; y_{ik}^{F1} = \; y_{ik}^{F1} + \beta * (y_{ik}^{F1} - y_{ik}^{F2}) + \alpha * rnd - 0.5 \tag{8}$$

on every matrix element, where:

$\beta = e^{-\gamma * r}$, gamma was set to 1

$\alpha = range * 0.05$, randomization component where the range is the range of interpretation, this was set to 30000 as the upper limit of the decision variable.

If no brighter firefly the firefly moves randomly:

$$moverandom(F1)\ y_{ik}^{F1} = y_{ik}^{F1} + rnd(\pm 50) \tag{9}$$

We tried a fixed random movement parameter which was selected by guesswork based on the range of the interpretation, the firefly count was 5, based on our previous research, and all run was measured in the same time period. The first run with the usual parameters was mentioned in the literature was given the following result.

Figure 4 shows the convergence of the solution. On the figure we can clearly see when the algorithm jumps out in a local optimum and the function not start to flatten during the time period of the run.

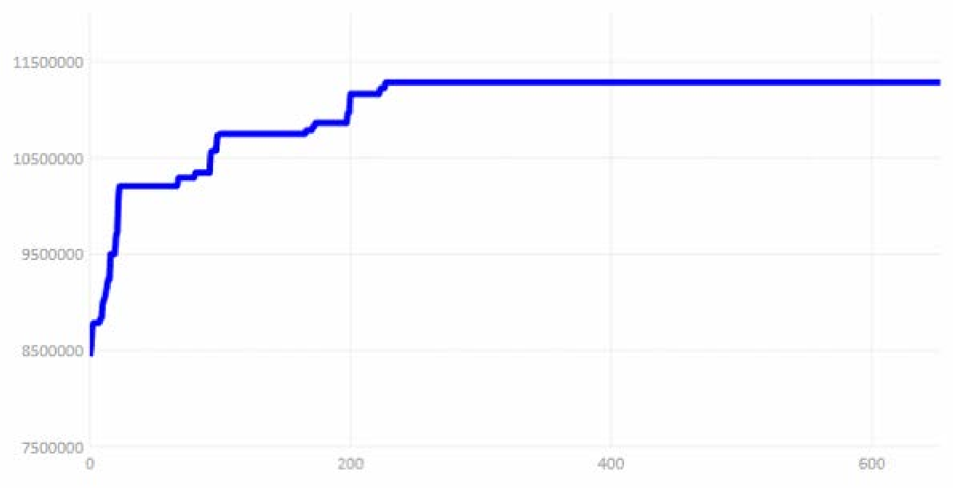

Figure 4. The solution, Firefly Algorithm first run without any adjustment
(X axis iteration, Y axis target function)

3. MULTILEVEL ALGORITHM

We often use swarm methods in our research works like the PSO (Particle Swarm Optimization and the Firefly Algorithm [6], both are widely used to solve a wide range of problems but mostly continuous problems. We did not use it so far on this type of logistical problem where there are so high number of variables present.

The main idea of this algorithm is that: in the first phase, a quick global search algorithm runs and in the second phase came a slower local search algorithm. We choose the random search algorithm for the global search phase, because it is quick and easily implementable. The initial population was generated with randomized fireflies and then the radom search generates solutions. The algorithm of the global search:

- initialize Firefly Population
- for x=0: Population.Count
    - TempFirefly = new FireFly()
    - TempFireFly.GenerateRanomValues()

  - TempFireFly.Fitness = CalculateFitness()
  - if TempFireFly.Fitness > Population[x].Fitness -> Population[x]= TempFireFly
- endfor

In the second phase the firefly algorithm – local search algorithm - gets the population and starts to work on it.

- for i=0:Population.Count
- for j=0:Population.Ccount
  - if Firefly[i].Fitness > Firefly[j].Fitness -> MoveToward Ffirefly[j]->Firefly[i]
- endfor j
- endfor i
- for i=0:Population.Count
  - if Firefly[i] not moved -> MoveRandom Firefly[i]
- endfor i

3. RESULTS

We tested the single-phase algorithm in a limited time period, then a multiphase algorithm with the same time period.

Table I.

Single phase algorithm with different firefly count (best values marked with yellow)

Firefly Count	**1 Phase algorithm**	**Random Search, cycle count: 1000**	**Random Search, cycle count: 10000**	**Random Search, cycle count: 50000**	**Random Search, cycle count: 100000**
5	11284687,39	11431439,27	11327421,58	11354798,83	11222171,48
10	11527232,71	11522102,74	11774711,33	11712414,57	11796566,52
50	11903132,47	11932054,00	11989873,43	11982746,42	11960405,94
100	12012847,18	12064379,69	12054711,29	12050887,65	11991769,25
500	11784064,48	11929456,88	11958241,20	11922737,38	11958241,20
1000	11927385,84	11904804,69	11953134,18	11985269,85	11930145,69

As the results (Table I) shows that, the two-phase algorithm can improve the solution, however the improvement is not that much as in our previous work which was tested in continuous state space [6]. The improvement on the convergence is much greater this time. As Figure 5 shows the solution convergence it is much faster than the original one on the Figure 4. In this problem the large decision variable number problem has a lot of suboptimal solutions and unfortunately near the optimum the convergence is very slow.

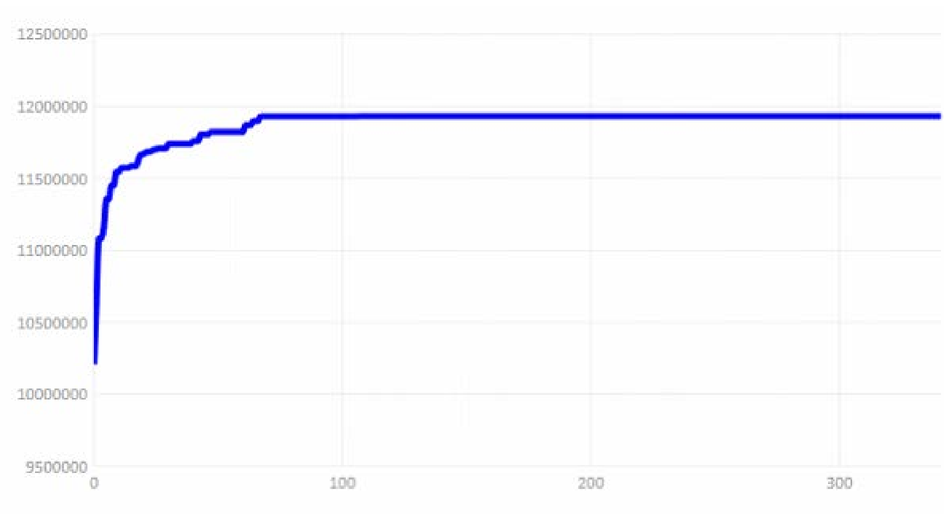

Figure 5. Convergence is improved using 2 phase algorithm
(X axis iteration, Y axis target function)

8. CONCLUSION

In this article we showed and tested an improvement method using a fast, global search and a slower local search on a large variable count logistic problem. Logistic problems mostly discrete and described with lot of decision variables. It is very important to determine the distance metric and the movement function or functions. However, these functions are not specified exactly when to use which, so we used the simplest and fastest functions. There are several distance metrics can be chosen from and the movement also needs to be defined in a discrete state space, mostly described with matrices which is not an easy task. These functions can greatly affect the quality of the solution. Since we are using heuristics, we can't be sure is the global optimum reached or not and can't be sure if the selected movement function for the actual problem is good or not. So, any optimization and the improvement methods need serious testing, to check if it helps or not. Our results shows that the quick global search can improve the solution but on very large scale problems, some methods may be needed which helps to get out of local optima.

ACKNOWLEDGEMENTS

"The described article/presentation/study was carried out as part of the EFOP-3.6.1-16-2016-00011 "Younger and Renewing University – Innovative Knowledge City – institutional development of the University of Miskolc aiming at intelligent specialisation" project implemented in the framework of the Szechenyi 2020 program. The realization of this project is supported by the European Union, co-financed by the European Social Fund."

LITERATURE

[1] Yang, X. S. (2008). *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, ISBN 978-1-905986-10-1.

[2] Cheung, N. J., Ding, X-M. & Shen, H-B. (2014). Adaptive Firefly Algorithm: Parameter Analysis and its Application. *PloS one, 9*(11), e112634. https://doi.org/10.1371/journal.pone.0112634

[3] Yurt, O., Kota, L., Jarmai, K. & Aglamaz, E. (2019). Analysis and Optimization of an Olive Oil Supply Chain: A Case from Turkey, *International Journal of Sustainable Agricultural Management and Informatics, 5(*1*),* https://doi.org/10.1504/IJSAMI.2019.101379

[4] Ghafil, H. N. & Jármai, K. (2018) Comparative study of particle swarm optimization and artificial bee colony algorithms, *Multiscience XXXII. MicroCAD International Multidisciplinary Scientific Conference,* Miskolci Egyetem, Paper: D1, https://doi.org/10.26649/musci.2018.030

[5] Kota, L. (2012) Optimization of the supplier selection problem using discrete firefly algorithm, *Advanced Logistic Systems: Theory and Practice* 6(1), 10-20.

[6] Kota, L. & Jármai, K. (2017) Application of multilevel optimization algorithms, Advances in Structural and Multidisciplinary Optimization, *Proceedings of the 12th World Congress of Structural and Multidisciplinary Optimization*, 710-715, https://doi.org/10.1007/978-3-319-67988-4_54

[7] Kota, L. & Jármai, K. (2014) Discretization of the Firefly Algorithm for the Travelling Salesman Problem, *28th microCAD International Multidisciplinary Scientific Conference, University of Miskolc*, Paper: D30

[8] Surafel, L. T. & Hong, C. O. (2012) Modified Firefly Algorithm, *Journal of Applied Mathematics*, vol. 2012, Article ID 467631, https://doi.org/10.1155/2012/467631

[9] Yu, S., Yang, S. & Su, S. (2013) Self-Adaptive Step Firefly Algorithm, *Journal of Applied Mathematics*, vol. 2013, Article ID 832718, https://doi.org/10.1155/2013/832718