OPTIMIZATION OF THE SUPPLIER SELECTION PROBLEM USING DISCRETE FIREFLY ALGORITHM

LÁSZLÓ KOTA[1]

Abstract: In this article I show a firefly optimization based algorithm which helps to choose the appropriate suppliers in a case of given order quantity of a given product. The developed algorithm takes account of the minimum and maximum order quantities at the different suppliers as constraints. It also takes account of the quantity discounts offered by the different suppliers, which can be described as step function. The algorithm takes account of the capacity and the cost of the used transport vehicles too. The article describes the operation of the algorithm and the penalty functions applied. In the last part the firefly algorithm and the solution given by the MS Excel solver's general reduced gradient and the evolutionary algorithm is compared.

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

1. Problem

The supplier selection is a crucial problem of the logistics. In this article I will describe that case when one product ordered in a given quantity but only one supplier can't provide all of the ordered quantity because of lack of capacity and more suppliers have to involve.

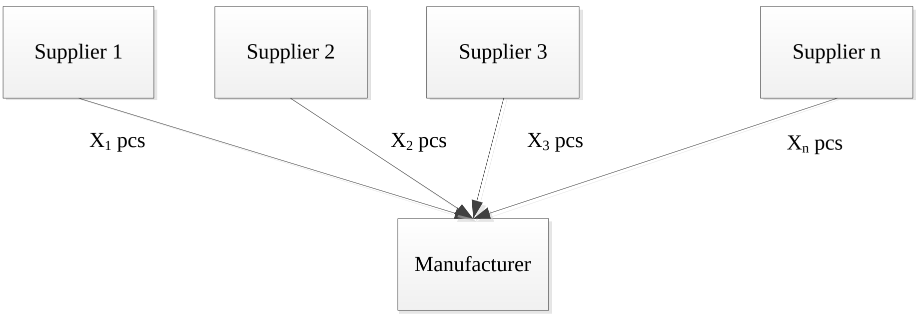


Fig. 1. Supplier problem

The required quantity of the product is given: Q. The minimum and maximum quantities are also given for every supplier:

$$O_i^{min}|_{i=1..n},\, O_i^{max}|_{i=1..n}. \quad (1)$$

[1] research assistant, University of Miskolc
altkota@uni-miskolc.hu
H-3515 Miskolc-Egyetemváros, Hungary

The capacity of the transport vehicles are: P_i, which is constant in this model for every transport vehicle at the given supplier. The transportation cost is also defined for every vehicle: tr_i.

The total transportation cost can be defined with the:

$$C_i^{TR} = RoundUp\left(\frac{X_i}{P_i}\right) \cdot tr_i \tag{2}$$

formula, where the RoundUp is the function of round up a real number up to the next integer.

The price of the ordered product is a function of the ordering quantity, because at large quantities usually the suppliers give reasonable amount of discounts, which described as a step function in this model.

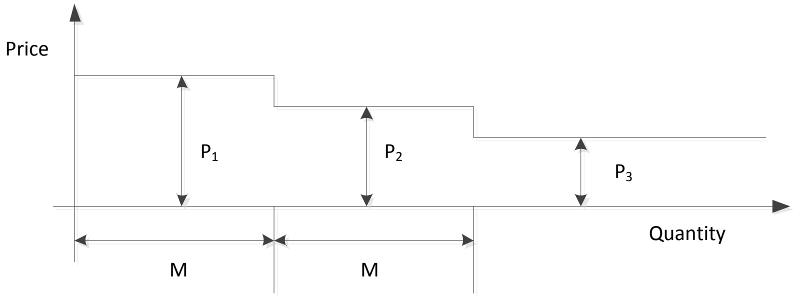


Fig. 2. The relation of the price and the quantity as a step function

As the *Fig. 2.* shows the price can be described with a step function where:

- M : is the quantity step of the given price, in this model the quantity steps are constants, so every step is equal,
- P_j : is the price of the given quantity step.

The discount price is applied only above the give amounts, this is the reason why a step function is formed.

In this case the problem is described with the following model:

- At all supplier the starting value of the price have to be defined: P_1 .
- It have to be defined what is the quantity after that the price is decreased: M.
- It have to be defined how much the price is decreased reaching the decrease steps, constant in this model.
- The decreasing step value have to be defined also, which defined how long the prices decreased lasts: U, after this step the price isn't decrease any more so it can't reach very small values or zero.

The purchasing price of the product (C_i^T) can't be described with an exact formula. The step function of the price can be described with the following flowchart (*Fig. 3.*).

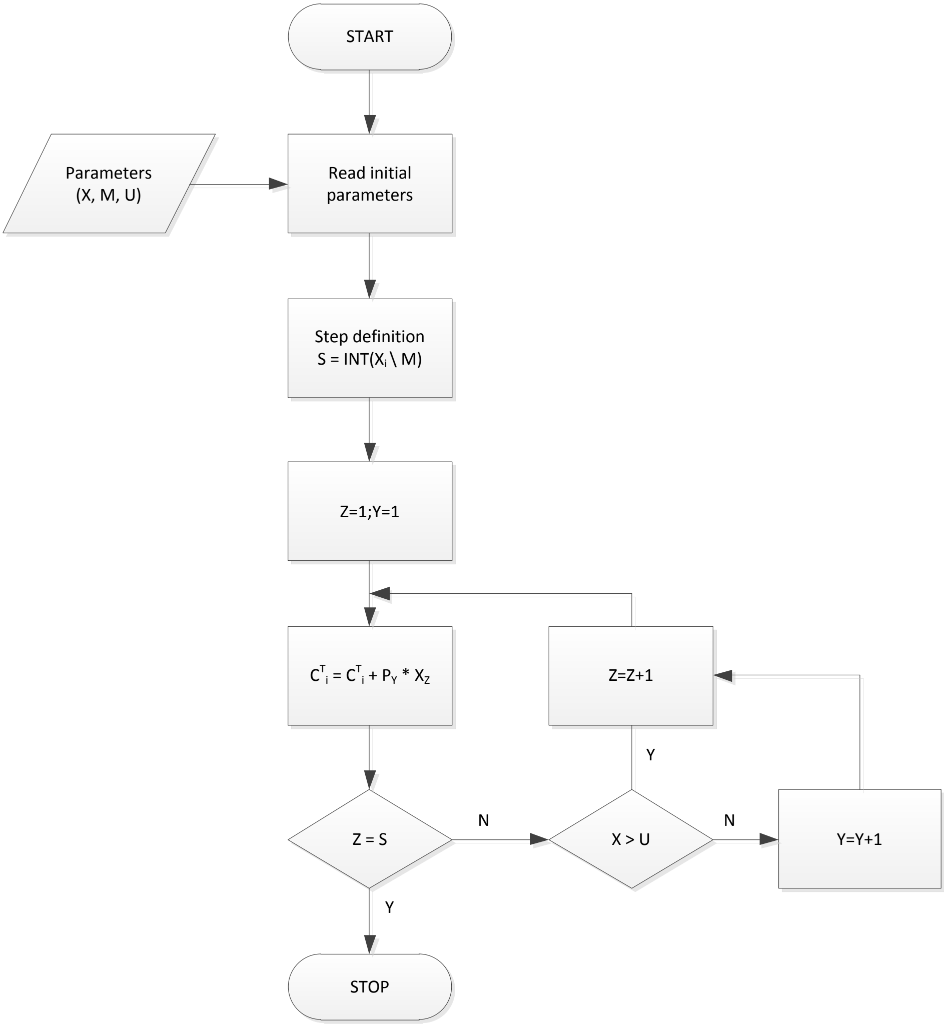


Fig. 3. Calculation of the price function

Constraints:

The ordered quantity from each supplier must not exceed the minimum and maximum quantity determined by the suppliers:

$$O_i^{min} < X_{Ii} < O_i^{max}. \tag{3}$$

The sum of the ordered quantities has to be equal to the required quantity:

$$sum(X_i\ ) = Q|_{i=1..n} \tag{4}$$

The ordered quantity has to be integer:

$$\forall X_i \ = Integer. \tag{5}$$

The target function of the optimization:

$$C = \sum_{i=1}^{n} (C_i^T + C_i^{TR}) \to min. \tag{6}$$

2. FireFly algorithm

The firefly algorithm developed by Xin-She Yang [1] is a metaheuristic algorithm based on the social behavior of the fireflies. 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=(x1, x2, …. xd)
2.  generate an initial population of fireflies: xi, (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 (Ij > Ii)
                            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 [4] (Particle Swarm Optimization) algorithm.

The movement of the firefly describes mostly by the

$$x_i = x_i + \beta_o e^{-\gamma r_{ij}^2}(x_j - x_j) + \alpha(rand - \tfrac{1}{2}), \quad (7)$$

formula or by the

$$\beta = \beta_0 \cdot e^{-\gamma r}$$
$$x_i = x_i \cdot (1 - \beta\ \ ) + x_i \cdot \beta\ \ + \alpha(rand - \tfrac{1}{2}), \quad (8)$$

formula which is equivalent.

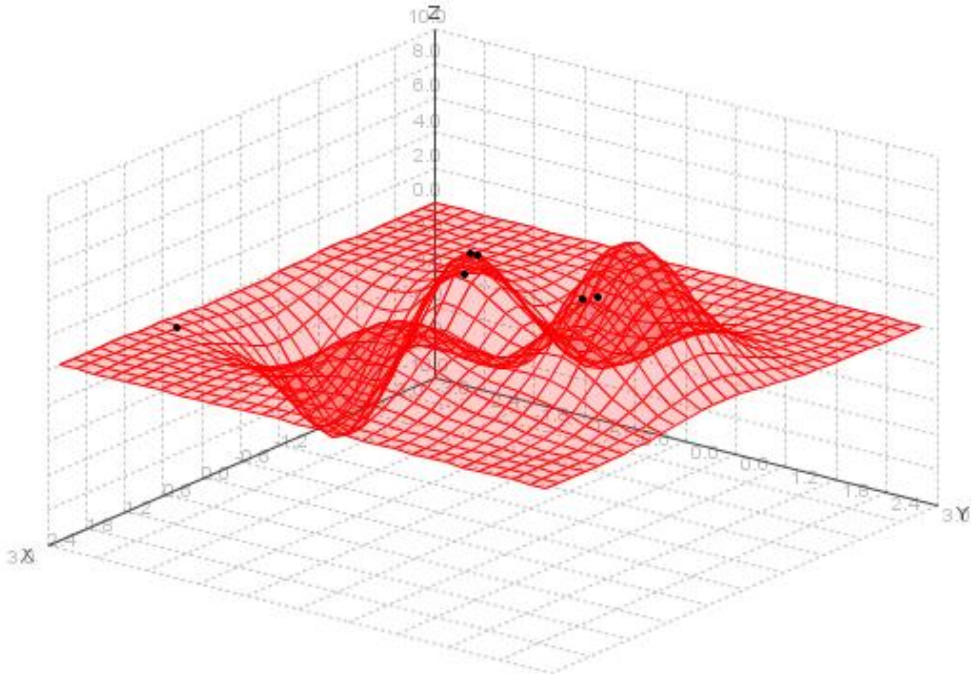


Fig. 4. Firefly algorithm searching for the maximum value in the Peaks function

The firefly algorithm was developed to solve continuous problems (*Fig. 4.)*, but the algorithm can be discretized so it can be used to solve non continuous permutation problems too [2].

3. Problem solution

The solution of the supplier problem is realized through the discretization of the firefly algorithm. The single fireflies represent a solution of the problem, just like the chromosomes of the genetic algorithm. The goodness of the individuals is evaluated with penalty functions.

The initial population is generated according to the formula (4).

FireFly 1	X_1	X_2	X_3		X_d
FireFly 2	X_1	X_2	X_3		X_d
.	.	.	.	.	.
FireFly n	X_1	X_2	X_3		X_d

Fig. 5. Population and data structure of the fireflies

In the next step the light intensity is evaluated to all the fireflies. The light intensity is a sum of the target function (6) and the penalty values.

The penalty functions:

The penalty value under the minimum ordering quantity is:

$$B^{min} = \sum_{j=1}^{d} \left(X_j^{min} - X_j\right)^2 \mid if\ X_j < X_j^{min} \tag{9}$$

where:

- d: is the number of the suppliers,
- X_j^{min}: is the minimum quantity can be ordered from the supplier *j*.

The penalty value above the maximum quantity is:

$$B^{max} = \sum_{j=1}^{d} \left(X_j - X_j^{max}\right)^2 \mid if\ X_j > X_j^{max} \tag{10}$$

where:

- d: is the number of suppliers,
- X_j^{min}: is the maximum quantity can be ordered from the supplier *j*.

The light intensity of an individual is:

$$I = C + B^{min} + B^{max}. \tag{11}$$

In the next phase the individuals move towards the brighter individuals. The brightest individual moves randomly.

The definition of the movement is really simple in a continuous state space but in a discrete state space the movement and the distance function have to be defined.

The determination of the distance of the individuals is performed by the following function:

$$D(F1, F2) = \sum_{j=1}^{d} abs(X_j^{F1} - X_j^{F2}) \tag{12}$$

where:

- F1: firefly 1, is the first operand of the distance function,
- F2: firefly 2, is the second operand of the distance function,

- X_j^{F1}: is the ordering quantity from the supplier j at the first firefly,
- X_j^{F2}: is the ordering quantity from the supplier j at the second firefly.

The movement of the individuals is the discretized variant of the continuous movement function [3].

Because the sum of the ordering quantities is a given constant value which must not altered in either way, so only even variances can happen among the ordering quantities (*Fig. 6.* marked quantities). So, if a given quantity is differing then another quantity has to differ at alternate sign.

	X_1	X_2	X_3	X_4	X_5	Total
FireFly 1	100	100	100	100	100	500
FireFly 2	100	99	100	101	100	500

Fig. 6. The individuals have the same total amount

The first step of the movement phase is to define the more (*M*) and the less (*L*) differences at every single firefly.

$$M_i = \begin{cases} 0 \; if \; X_i^{F1} < X_i^{F2} \\ 1 \; if \; X_i^{F1} > X_i^{F2} \end{cases} \tag{13}$$

$$L_i = \begin{cases} 0 \; if \; X_i^{F1} > X_i^{F2} \\ 1 \; if \; X_i^{F1} < X_i^{F2} \end{cases} \tag{14}$$

The step value has to be defined based on the distance of the fireflies (12).

$$S = RoundUp\left(\frac{D^2}{R^2}\right) \tag{15}$$

where:

- R^2: is the range of the ordering quantity, the range of the state space: from 0 to Q

The randomly choose one more (*M*) and one less (*L*) difference and the more value needs to be decreased the less value needs to be increased, so the distance of the fireflies is decreasing.

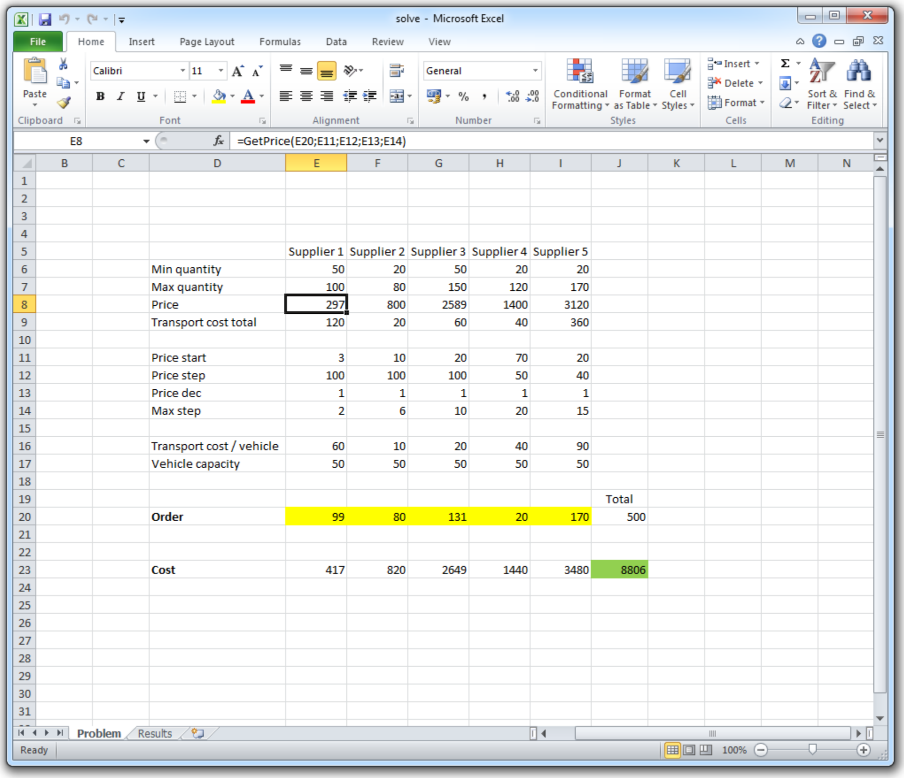
$$\begin{aligned} X_i &= X_i - S \\ X_j &= X_j + S \end{aligned} \tag{16}$$

At the random movement also needs to be change even elements to preserve the total value. So it has to be decreased by randomly selected quantity and the other randomly selected quantity has to be increased with the same amount. The experiments showed that the best convergence can be reached by the value 1 in our case. So the random movement can be described as:

$$\begin{aligned} X_i &= X_i - 1, \\ X_j &= X_j + 1. \end{aligned} \tag{17}$$

4. Comparison with the MS Excel solver

Let's compare the solution of the firefly algorithm with the solution of the MS Excel's solver module. In this example there are 5 suppliers with randomly selected minimum, maximum quantities (*Fig. 7.* Solution with MS).



	Supplier 1	Supplier 2	Supplier 3	Supplier 4	Supplier 5	
Min quantity	50	20	50	20	20	
Max quantity	100	80	150	120	170	
Price	297	800	2589	1400	3120	
Transport cost total	120	20	60	40	360	
Price start	3	10	20	70	20	
Price step	100	100	100	50	40	
Price dec	1	1	1	1	1	
Max step	2	6	10	20	15	
Transport cost / vehicle	60	10	20	40	90	
Vehicle capacity	50	50	50	50	50	
						Total
Order	99	80	131	20	170	500
Cost	417	820	2649	1440	3480	8806

Fig. 7. Solution with MS Excel solver

Because of the nonlinearity of the problem the simplex algorithm of the MS Excel solver can't calculate the optimum values. It gives an error message, „The LP solver cant calculate the values because of the nonlinear constraints". So the nonlinear GRG (general reduced gradient method) and the evolutionary solvers was examined (*Table I.)*.

The step functions which calculate the quantity discounts and the function which calculates the transportation cost needed to be implemented in VBA language (Visual Basic for Applications) in a public module and the functions needed to be declared as public too. So the Excel can use the user implemented function as a normal worksheet function (*Fig. 7.*) during the optimization.

In the first step the output parameters have to be defined (*Fig. 8.*):

- J23: target cell, in this cell will the Excel store the total cost which have to be minimized (*C*),
- E20:I20: variable cells, the order quantities from the suppliers (X_i).

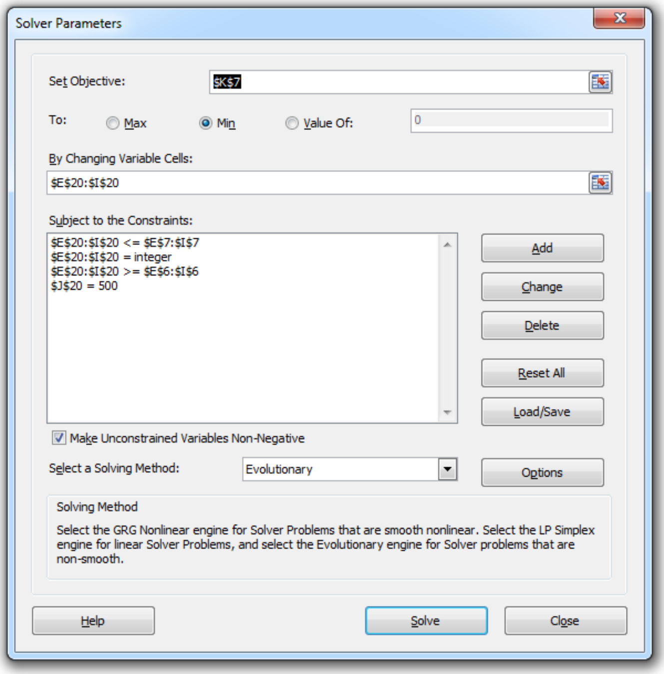


Fig. 8. Parameters of the MS Excel solver

Then the constraints have to be defined, these are the followings:

- the sum of the order quantities must not exceed the required quantity (Q): J20
- the individual quantities can't exceed the largest quantity that supplier can provide (X_i) for eg.: E-I20 < E-I7,
- the individual quantities can't be (X_i) can't be lesser than the minimum quantity that supplier can provide: pl. E-I20 > E-I6,
- and the quantities have to be integer values: E-I20 = Integer.

All the constraints can be defined with group selection of the cells.

Table I. Results

Method	**Run time [s]**	**Relative difference**	**Target function**	**Relative difference**
FireFly	0,085	100,00%	8806	100,00%
Nonlinear GRG (single point, advancing central)	0,109	128,24%	12723	69,21%
Nonlinear GRG (multipoint, advancing)	48,6	57176,47%	9058	97,22%
Nonlinear GRG (multipoint, ventral)	113,6	133647,06%	8873	99,24%
Evolutionary (1 cycle)	64,974	76440,00%	9666	91,10%
Evolutionary (3 cycles)	117,717	138490,59%	88,06	100,00%

5. Summary

The developed firefly based algorithm is very well scalable. The algorithm could handle virtually any number of suppliers, till the expansion of MS Excel's solution is difficult. The run times shows that the MS Excel can't solve this problem with the default settings, although using very large run times it is approximates it well. Only the evolutionary algorithm was able to solve the problem but it is even at the third run. In this case it handles the previous run's optimum as a starting point to refine the solution. But the results show that the firefly algorithm solves the problem in the fragment of the evolutionary algorithm running time.

Acknowledgements

The research was supported by the TÁMOP 4.2.1.B-10/2/KONV-2010-0001 entitled "Increasing the quality of higher education through the development of research - development and innovation program at the University of Miskolc" supported by the European Union, co-financed by the European Social Fund.

Literature

[1] Yang, X. S.: *Nature-Inspired Metaheuristic Algorithms*, Luniver Press 2008, p. 128, ISBN 1-905986-10-6
[2] Jati, G. K.; Suyanto: *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 2011 pp.: 393-403, ISBN: 978-3-642-23856-7
[3] Sayadia, M. K.; Ramezaniana, R.; Ghaffari-Nasaba, N.: *A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems*, International Journal of Industrial Engineering Computations 1 (2010) 1–10, doi: 10.5267/j.ijiec.2010.01.001
[4] Farkas, J.; Jármai, K.: *Design and Optimization of Metal Structures*, Horwood Publishing Limited 2008, ISBN 978-1-904275-29-9, 300 p.