İSTANBUL ÜNİVERSİTESİ
İKTİSAT FAKÜLTESİ
EKONOMETRİ VE İSTATİSTİK
DERGİSİ

Ekonometri ve İstatistik Sayı:13 (12. Uluslararası Ekonometri, Yöneylem Araştırması, İstatistik Sempozyumu Özel Sayısı) 2011 114–124

A DISCRETE PARTICLE SWARM OPTIMIZATION ALGORITHM FOR BI-CRITERIA WAREHOUSE LOCATION PROBLEM

Fehmi Burcin OZSOYDAN[1] **Tugba SARAC**[2]

Abstract

The uncapacitated warehouse location problem (UWLP) is one of the widely studied discrete location problems, in which the nodes (customers) are connected to a number (*w*) of warehouses in such a way that the total cost, yields from the dissimilarities (distances) and from the fixed costs of the warehouses is minimized. Despite *w* is considered as fixed integer number, the UWLP is NP-hard. If the UWLP has two or more objective functions and *w* is an integer variable, the UWLP becomes more complex. Large size of this kind of complex problems can be solved by using heuristic algorithms or artificial intelligent techniques. It's shown that Particle Swarm Optimization (PSO) which is one of the technique of artificial intelligent techniques, has achieved a notable success for continuous optimization, however, PSO implementations and applications for combinatorial optimization are still active research area that to the best of our knowledge fewer studies have been carried out on this topic. In this study, the bi-criteria UWLP of minimizing the total distance and total opening cost of warehouses. is presented and it's shown that promising results are obtained.

Keywords: Warehouse Location Problem, Particle Swarm Optimization, Discrete Location Problems, Bi-criteria.
Jel Classification: C61, C63

Özet

Kapasitesiz Depo Yeri Belirleme Problemi, açılacak "w" adet deponun toplam açma maliyetlerinin ve düğümlerde bulunan müşteriler ile açılan depolar arasındaki uzaklıklardan kaynaklanan maliyetlerin toplamının en küçüklendiği, literatürde yaygınca bilinen bir kesikli yer belirleme problemidir. "w" sabit bir sayı olmasına rağmen bu problem Np-Hard sınıfında yer almaktadır. Eğer birden fazla amaç fonksiyonu aynı anda ele alınır ve "w" sayısı sabit yerine değişken kabul edilirse problem daha da zorlaşmaktadır. Büyük boyutlu örnekleri ise ancak sezgisel tekniklerle ele alınabilmektedir. Öte yandan Parçacık Sürüsü Optimizasyonu' nun (PSO), sürekli eniyilemede ciddi bir başarıya sahip olduğu gösterilmiştir. Fakat Kombinatoriyel Problemlerde uyarlama ve uygulama alanı hala aktif bir araştırma alanıdır ve bilindiği kadarıyla, bu başlık altında daha az çalışma yürütülmüştür. Bu çalışmada İki Kriterli Kapasitesiz Depo Yeri Belirleme Probleminin çözümü için bir Parçacık Sürüsü Optimizasyonu Algoritması önerilmiştir.

Anahtar Kelimeler: Depo Yeri Belirleme Problemi, Parçacık Sürüsü Optimizasyonu, Kesikli Yer Belirleme Problemleri, İki-Kriter.
Jel Sınıflaması: C61, C63

[1] Arş.Gör. Eskişehir Osmangazi Üniversitesi Meşelik Kampüsü Mühendislik Mimarlık Fakültesi Endüstri Mühendisliği Bölümü, e-posta: fbozsoydan@ogu.edu.tr

[2] (Sorumlu Yazar), Yrd. Doç. Dr. Eskişehir Osmangazi Üniversitesi Meşelik Kampüsü Mühendislik Mimarlık Fakültesi Endüstri Mühendisliği Bölümü, e-posta[2]: tsarac@ogu.edu.tr

1. INTRODUCTION

UWLP is one of the most widely studied discrete location problems (Cornuejols, et al., 1990; Gao, at al., 1994) and this problem is known to be NP-hard (Krarup and Pruzan, 1983). There are many surveys related to this topic, in that respect examining all essential contributions is beyond the scope of this paper (Dearing, 1985; Francis 1983).

The UWLP is gone under different names in the literature. The uncapaciteted facility location problem, simple plant location problem can be considered as some of them. Despite the NP-hardness of the problem it was shown by (Grishukhin, 1994) that, there exist some polynomially solvable special cases.

In UWLP, a number of "*w*" warehouses is tried to be located at some candidate points on an Euclidian graph with fixed costs so as to minimize the total cost arising from locating a warehouse and the distances between "*n*" customers (nodes) to the nearest possible located warehouse. In addition there does not exist a capacity constraint for all of the warehouses. "*w*" can be defined either a fixed integer number or an integer variable number, however problem becomes more complex whether "*w*" is chosen an integer variable. In this study, "*w*" is an integer variable number and the bi-criteria UWLP of minimizing the total distance and total fixed cost of warehouses to be opened is presented. The mathematical formulation of the considered problem is given below.

Let $I = \{1, \dots, m\}$ be the set of possible locations to establish a warehouse, $J = \{1, \dots, n\}$ be the set of customers, $c_{i,j}$ be the distance between warehouse *i* and customer *j*,b_j be the demand of customer *j* and a_i be the opening cost of warehouse i.

Let binary variable y_i be equal to 1 if warehouse *i* is opened and decision variable $x_{i,j}$ denotes amount of the transportation from warehouse *i* to customer *j*.

We have two objective functions:

- Total distance:

$$F_1(x,y) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \quad (1)$$

- Total opening cost of warehouse:

$$F_2(x,y) = \sum_{i=1}^{m} a_i y_i \quad (2)$$

In respect of the notation given above, we can now formulate the mathematical model as:

$$min = [F_1(x,y), F_2(x,y)] \quad (3)$$

st.

$$\sum_{i=1}^{m} x_{ij} = b_j \qquad \forall j \in J \quad (4)$$

$$y_i \leq x_{i,j} \leq \left(\sum_{j=1}^{n} b_j\right) y_i \quad and \quad y_i \in \{0,1\}, \qquad \forall i \in I \quad (5)$$

$$y_i \in \{0,1\}, \qquad \forall i \in I \quad (6)$$

$$x_{ij} \geq 0, \qquad \forall i \in I, \quad j \in J \quad (7)$$

Equation 1 is the minimization of the total distance whereas equation 2 is the minimization of the costs of opening of warehouses. Equation 3 is the scalarized form of equations 1 and 2. Equation 4 ensures that demands of all customers are met. Equation 5 is provides distribution from warehouse *i* to any other possible customers if warehouse *i* is opened. Equations 6 and 7 are the integrality constraints.

2. SOLUTION APPROACHES

Solution approaches for UWLP can generally be divided into three categories: heuristic methods, exact methods, and hybrid approaches. In this paper we used both exact and Particle Swarm Optimization (PSO) which is one of the artificial intelligent techniques, solution approaches for solving the UWLP.

Because of the multiobjective nature of the model, a solution process of these kinds of problems has been considered in two stages: the scalarization of the given problem, and the solution of the scalarized problem. Scalarization and the solution approaches for UWLP were explained in Section 2.1 and Section 2.2 respectively.

2.1. Scalarization

Scalarization means combining different objectives to a single one such that the obtained single objective optimization problem allows finding efficient solutions of the initial multiobjective problem. There are many scalarization methods for combining different objectives to a single one (see, for example, Luc (1989), Chankong and Haimes (1983), Ehrgott (2005)). However some of these methods such as weighted sum method are not appropriate to find every nondominated solution. Tchebycheff metric based scalarization method can be applied as an efficient method for finding supported and unsupported nondominated solutions for multiobjective programming with a nonconvex feasible region. In this study we used augmented Tchebycheff function for scalarization and it is given below;

$$\min_{x \in X} \max_{l=1..m} [w_l(f_l - y_l^U)] + \rho \sum_{l=1}^{m} (f_l - y_l^U) \qquad (8$$

where ρ is very small positive number, f_l is l th. objective function, y_l^U is a reference value of l th. objective function m is the number of objective functions.

2.2. The solution of the scalarized problem

The UWLP is solved both by exact and PSO methods. They have been explained in Section 2.2.1 and Section 2.2.2 respectively.

2.2.1. Exact solution of the UWLP

Exact solution approaches used for solving UWLP type problems generally have two main difficulties. The first of them is related to the solution time, which increases exponentially with the number of integer decision variables and the second one is the nonconvexity of the problem which again is a result of the existence of integer variables, even though the objective and constraint functions in such models are all linear. So we can solve exactly only the small size UWLP instances by using the solvers of GAMS, in this study.

2.2.2. Particle Swarm Optimization Algorithm

Firstly introduced by James Kennedy and Russel C. Eberhart, PSO is one the swarm intelligence based (Kennedy and Eberhart, 1995; Eberhart and Kennedy, 1995; Eberhart and et.al., 1996; Kennedy, 1997; Angeline, 1998; Kennedy and et.al., 2001) algorithms that simulates the behavior of the social organisms such as bird flocking and fishing schooling. As collecting food or searching rich food sources of a population in nature, PSO similarly uses the communication and coordination between the particles (individuals) to converge to optimum solution. What is meant here by communication and coordination is, a comparative of the fitness values (value of objective function) and utilization of the positions among population, respectively. These two phenomenons are related to two fundamental terms of PSO: The position and the velocity. In PSO, each particle in a population has a position and a velocity vector. Derived from utilization of other particles', individual best's and global best's positions which will be clarified later, a velocity vector is generated. That particle then modifies its own position due to the velocity vector for which the achievement of coordination and communication with the experience of other individuals is not ignored.

A PSO includes three basic steps: evaluation of the fitness values of each particle, update the values and positions of individual and global best, and update the velocities and positions of each individual in the population according to the equations 9 and 10.

Before introducing these steps, it may be more beneficial to explain the terms in equations 9 and 10. As can clearly be seen from equation 9, a velocity vector is composed of three additive components which can be thought as the fundamental characteristics frame of PSO. These components are known as inertia component, cognitive component and social component, respectively. $V_{i(t)}$ and $X_{i(t)}$ mean the velocity and position of the i.th particle at iteration *t*, respectively. Similarly $X^*_{iind(t)}$ and $X^*_{gl(t)}$ mean the individual best position of the i.th particle and globally best position obtained until iteration *t,* respectively. w (inertia coefficient), c1 (cognitive coefficient) and c2 (social coefficient) are user-supplied parameters and they are usually accepted between the intervals of

$0 \leq w \leq 1.2, 0 \leq c1 \leq 2$ and $0 \leq c2 \leq 2$. Ω_1 is a uniformly distributed random number between 0 and 1, and affects the speed of cognitive component whereas Ω_2 performs the same task for social component. Similarly if the coefficient w is chosen relatively smaller, this may dampen particle's inertia velocity or conversely if it is chosen a greater value it may accelerate the inertia component. As a result of the accelerating or dampening parameters, the situation, "What if the lower or upper bounds of any variable are violated?" comes into question. To prevent it, considering the domain of the variable a lower X_{min} and upper bound X_{max} for $X_{i(t)}$ can be defined. Similarly because of the same reason, for not to converge too quickly and for not to miss any promising region and solutions, as for $X_{i(t)}$, a lower V_{min} and upper bound V_{max} for $V_{i(t)}$ can also be defined. Despite bounds of the velocity vector is a user-supplied parameter, considering robustness of the algorithm the relation between V_{max} and domain of X can be defined as in equation 11 where k is a continuous coefficient, usually accepted in the interval $0.1 \leq k \leq 1$.

$$V_{i(t+1)} = wV_{i(t)} + c_1\Omega_1\left[X^*_{iind(t)} - X_{i(t)}\right] + c_2\Omega_2\left[X^*_{gl(t)} - X_{i(t)}\right] \quad (9)$$

$$X_{i(t+1)} = X_{i(t)} + V_{i(t+1)} \quad (10)$$

$$V_{max} = k(X_{max} - X_{min}) \quad (11)$$

The first step is fairly clear. The fitness values of each particle at iteration t are evaluated according to the objective function of the problem. At the second step a comparison is performed between recently obtained results and individual best values which mean the best values obtained so far for each particle. If a better result is obtained here for a particle, then that particle forgets its latest position and value and memorizes recently obtained position and value as new individual best of its own.

It's pretty clear that equations 9 and 10 can be used for continuous optimization; however, apart from the modifications mentioned in the upper paragraph, there also exists binary integer programming versions of PSO. Kennedy and Eberhart (1997) firstly introduced a binary version of PSO. PSO has been applied to a wide range of applications including discrete and continuous optimization. In this study, a new discrete modification of PSO with cross-over (Feng et.al., 2007; Pant et.al., 2009) is proposed. Hence, for a comprehensive survey for PSO and its applications, please check (Frans, 2001; Kennedy et.al., 2001).

In the surveys of Fang et al. (2007), a cross over operator was adapted for information transmission among particles. According to this approach a particle crosses with its individual best and global best obtained so far, respectively and the new particle is created (Fang et.al., 2007). Apart from original cross over, this cross over operator produces one chromosome. But there may occur some drawbacks of this approach which may crucially affect the performance of the algorithm.

Developed in this study, we introduce a novel technique to handle with both cross over sequence drawback and much processing requirement. We tried to orient the search by orienting the whole population and we now introduce B1, B2 and B3 as three new user-supplied parameters all between [0,1]. These bounds are used for determining the cross over type of the i.th particle. To perform this, a random number generation for all particles in the swarm is needed. A pseudo code for this procedure, scale for cross-over type and flow diagram of the proposed PSO are given below in Figure1, Figure2 and Figure3, respectively.

i=0

Do until i = SS (SS is the swarm size.)

i=i +1

RN (random number) =Rnd( )

If $RN \leq B1$ *and* $RN \geq 0$ *then*

Do not perform cross over for particle(i)

End if

If $RN \leq B2$ *and* $RN > B1$ *then*

Cross the particle(i) with a random particle chosen from swarm.

End if

If $RN \leq B3$ *and* $RN > B2$ *then*

Cross the particle(i) with its individual best obtained so far.

End if

If $RN \leq 1$ *and* $RN > B3$ *then*

Cross the particle(i) with global best of the swarm obtained so far.

End if

Loop

Figure1. Pseudo code for the proposed cross-over technique.

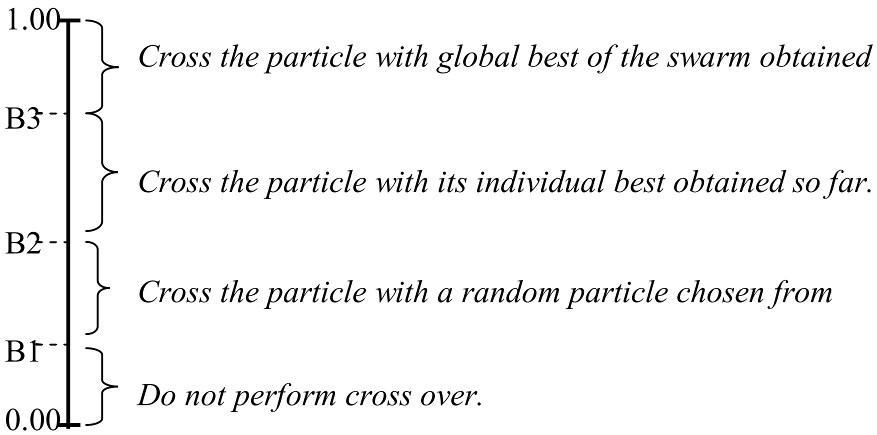

Figure 2. The scale for cross over type.

A general flow diagram of the proposed PSO is below given in Figure 3.

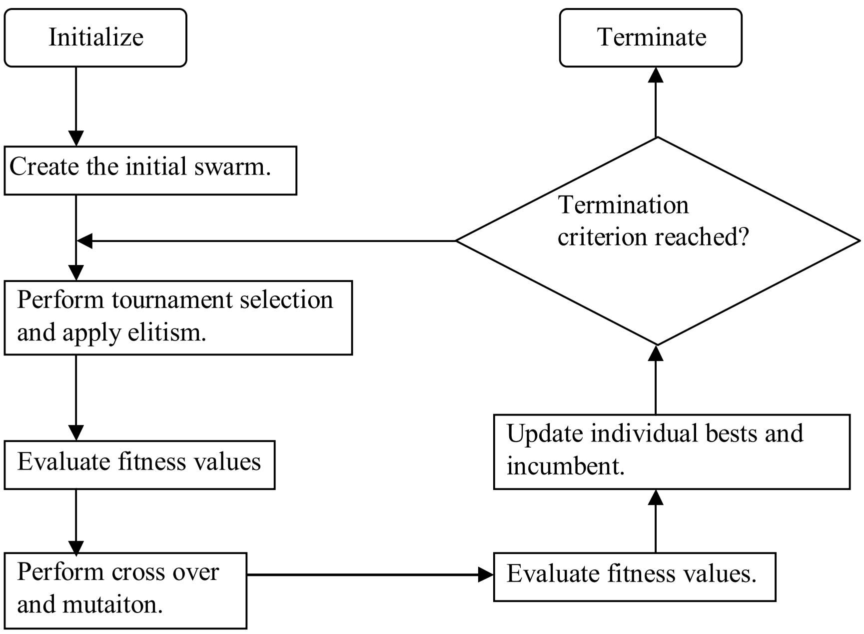

Figure 3. Flow diagram of the proposed PSO.

In this study, an effective chromosome encoding technique is applied. In this technique, all of the genes in a chromosome are shown as a uniformly distributed random number between 0-1. The information of the opened warehouses is memorized in another array called warehouse array created with a size “n” and with alleles of 0 and 1. The values of this array are determined randomly in the initial population but this array is allowed to have at least one gene, with a value of 1. In this manner, the i^{th} warehouse can be connected to the j^{th} customer as follows: Let “k” warehouses be opened and the first gene of the chromosome (denoted by random numbers) be 0.18. a=1+int(0.18*p) where “a” is the a^{th} opened warehouse among the opened warehouses. This connection operation is repeated until all customers are connected to a warehouse according to the information they keep beneath their own gene.

A single point cross-over is applied for cross-over but as can be easily seen, infeasible cases for warehouses arrays with all values of genes is 0 may be created throughout iterations. In such a case, a randomly chosen warehouse is opened.

Mutation operator simply interchanges value of a gene with a new random number unless the gene will be mutated belongs to the warehouse arrays. In such a case the value of a gene in the warehouse array is changed to 1 if it’s 0 and vice versa.

3. COMPUTATIONAL RESULTS

The performance of the algorithm was tested on 8 different test problems for which two of each test problems of dimensions 50, 100, 150 and 200 were created by the authors. The proposed algorithm was coded by using Visual Basic 6.0 and executed on a computer with a hardware of Core2Duo 2.36GHZ and 3GB RAM to compare to the results of GAMS CPLEX Solver derived from the same computer.

Unfortunately GAMS was unable to solve the problems; however to demonstrate that the proposed method is able to obtain optimum solutions, a very small test problem with size n=m=8 was run for both GAMS and proposed PSO and it was observed that both of them could obtain optimum solutions in a few second.

4. CONCLUSIONS

Because of having several parameters in the proposed method, the levels of the parameters may crucially affect the quality of the solutions derived from the proposed PSO which means that en experimental design should be carried out to optimize these parameters; however in this study because of the unavailability of comparing the results, only a group of tests were made to determine a good level for each parameter.

In further studies the same method will be applied for different scalarization techniques and if the availability of a comparison is satisfied, an experimental design is going to be carried out to optimize the level of the parameters.

REFERENCES

A. Kaveh, S. Talatahari. A particle swarm ant colony optimization for truss structures with discrete variables. Journal of Constructional Steel Research 65 (2009) 1558-1568.

Angeline P.J. 1998. Using Selection to Improve Particle Swarm Optimization. In WCCI-98 Proceedings of the IEEE World Congress on Computational Intelligence pp. 84-89. IEEE Press.

Chankong, V., and Haimes, Y.Y., 1983. Multiobjective Decision Making: Theory and Methodology, Elsevier Science Publishing Co, New York.

Cornuejols, G., Nemhauser, G.L., Wolsey, L.A. (1990).: The uncapacitated facility location problem, In: Discrete Location Theory, Eds: P.B.Mirchandani and R.L. Francis, John Wiley & Sons, chap III, pp 120-171.

Dearing, P.M. (1985).: Location problems, OperationsResearch Letters,Vol. 4, pp 95-98.

Eberhart, R. C., Dobbins, R. W., and Simpson, P. (1996), Computational Intelligence PC Tools, Boston: Academic Press.

Eberhart, R. C., and Kennedy, J. (1995). A New Optimizer Using Particles Swarm Theory, Proc. Sixth International Symposium on Micro Machine and Human Science (Nagoya, Japan), IEEE Service Center, Piscataway, NJ, 39-43.

Ehrgott, M., 2005. Multicriteria Optimization, Springer-Verlag, 2nd edition.

Fang L., Chen P. and Liu S. Particle Swarm Optimization with Simulated Annealing for TSP. Proceedings of the 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases, Corfu Island, Greece, February 16-19, 2007.

Francis, R.L., McGinnis, L.F., White, J.A. (1983).:Locational analysis, European Journal of Operational Research, Vol. 12, pp 220-252.

Frans van den Bergh. An Analysis of Particle Swarm Optimizers. PhD thesis, University of Pretoria 2001.

Grishukhin, V.P. (1994).: On polynomial solvability conditions for the simplest plant location problem,In: Selected topics in discrete mathematics, Eds:A.K.

Kelmans and S. Ivanov, pp 37-46, American Mathematical Society, Providence, RI.

Kennedy, J. (1997), The Particle Swarm: Social Adaptation of Knowledge, Proc. IEEE International Conference on Evolutionary Computation (Indianapolis, Indiana), IEEE Service Center, Piscataway, NJ, 303-308.

Kennedy J. and Russel C. Eberhart. Particle swarm optimization. In proceedings of the IEEE International Conference on Neural Networks, volume IV, pages 1942-1948, Piscataway, NJJ,1995. IEEE Press.

Kennedy J., Eberhart R. C. and Shi Y., “Swarm Intelligence”, Morgan Kaufmann, San Mateo, 2001, CA.

Kennedy J. and Eberhart R. C., “A Discrete Binary Version of the Particle Swarm Optimization”, Proc. Of the conference on Systems, Man, and Cybernetics SMC97, pp. 4104-4109, 1997.

Krarup, J., Pruzan, P.M. (1983).: The simple plant location problem: Survey and synthesis, European Journal of Operational Research, Vol. 12, pp 36-81.

Luc, D. T., 1989. Theory of Vector Optimization, Lecture Notes in Economics and Mathematical Systems, Springer Verlag, Berlin.

Pant M., Thangaraj R., and Singh V. P. Particle Swarm Optimization with Crossover Operator and its Engineering Applications. IAENG International Journal of Computer Science. 36:2, IJCS_36_2_02 (2009).

S. He, Q.H. Wu, J.Y. Wen, J.R. Saunders, R.C. Paton. A particle swarm optimizer with passive congregation. BioSystems 78 (2004) 135–147.