World Academy of Science, Engineering and Technology
International Journal of Industrial and Manufacturing Engineering
Vol:3, No:12, 2009

Application of Ant colony optimization for Multi-objective Production Problems

Teerapun Saeheaw, Nivit Charoenchai, and Wichai Chattinnawat

***Abstract*—**This paper proposes a meta-heuristic called Ant Colony Optimization to solve multi-objective production problems. The multi-objective function is to minimize lead time and work in process. The problem is related to the decision variables, i.e.; distance and process time. According to decision criteria, the mathematical model is formulated. In order to solve the model an ant colony optimization approach has been developed. The proposed algorithm is parameterized by the number of ant colonies and the number of pheromone trails. One example is given to illustrate the effectiveness of the proposed model. The proposed formulations; Max-Min Ant system are then used to solve the problem and the results evaluate the performance and efficiency of the proposed algorithm using simulation.

***Keywords*—**Ant Colony Optimization, multi-objective problems

I. INTRODUCTION

ANT colony optimization (ACO) is a recent family member of the meta-heuristic algorithms and can be used to solve complex optimization problems with few modifications by adding problem-dependent heuristics. ACO is a biological inspiration simulating the ability of real ant colony of finding the shortest path between the nest and food source. It is one of the successful applications of swarm intelligence which is the field of artificial intelligence that study the intelligent behavior of groups rather than of individuals such as the behavior of natural system of social insects like ants, bees, wasps, and termites. Swarm intelligence uses stigmergy which is a form of indirect communication through the environment.

The class of complex optimization problems called combinatorial optimization problems are found in many areas of research and development. Traveling Salesman Problem (TSP), Quadratic Assignment Problem (QAP), Vehicle Routing Problem (VRP), Graph Coloring Problem (GCP), Sequential Ordering Problem (SOP), Job Scheduling Problem (JSP) and Network Routing Problem (NRP) are some examples of these problems. Combinatorial optimization problems arise when the task is to find the best out of many possible solutions to a given problem, provided that a clear notion of solution quality exists. In contrast to other optimization problems, combinatorial problems have a finite number of candidate solutions. Therefore, an obvious way to solve these problems is to enumerate all candidate solutions by comparing them against each other. Unfortunately, for most interesting combinatorial optimization problems, this approach proves to be impractical since the number of candidate solutions is simply too large. The only way to tackle the problems is to apply heuristic search that delivers no guarantee of finding the optimum solution [1, 2]

The main element of ACO success is the use of a combination of priori information (heuristics) about the quality of candidate solutions (also called greedy strategy) and posteriori information (pheromone) about the goodness of the previously obtained solutions (also called positive feedback or autocatalytic process). This seems reasonable since many researches that study the characteristics of some well known optimization problems show that there is a correlation between the solution quality and the distance from the optimal solution [3, 4]. Several well known ACO examples are Ant System [5], Ant Colony System [6, 7], Max-Min Ant System [8], Ranked Ant System [9] and Best Worst Ant System [10, 11, 12]. These algorithms show interesting performance and are competitive with other state of the art optimization methods. However, more research work is needed to enhance the ACO algorithms performance especially on large volume of combinatorial problems. Increasing the number of ants used to tackle a large problem almost yield to a worse algorithm performance. The key element is the organization of the ants' population and the coordination of their works in such a way that yields to a good exploration of the large search space in a strong coupling with the exploitation of the search history.

In this paper, a ACO meta-heuristic algorithm is proposed. The algorithm uses ant colonies and can be efficiently used to multi-objective production problems. The rest of this paper is organized as follows. Section 2 describes the ant colony optimization. The proposed formulations of the multi-objective production problems are reviewed in section 3. Section 4 proposes applied max-min ant system. The computational results of the algorithm testing are presented in section 5 and section 6 presents the conclusion and suggested future work.

II. ANT COLONY OPTIMIZATION

Ant colony optimization (ACO) meta-heuristic, a novel population-based approach was recently proposed in 1992 by Marco Dorigo et al. to solve several discrete optimization problems [13]. The ACO mimics the way real ants find the shortest route between a food source and their nest. The ants communicate with one another by means of pheromone trails and exchange information about which path should be followed. The more the number of ants traces a given path, the more attractive this path (trail) becomes and is followed by other ants by depositing their own pheromone. This auto catalytic and collective behavior results in the establishment of the shortest route.

Ants find the shortest path from their nest to the food source with the help of pheromone trail. This characteristic of ants is adapted on ant colony optimization algorithms to solve real problems with using exactly some characteristics of ants and some new addition.

The method improved by modeling real ants use exactly the same specifications taken from real ants are below [14]:

- The communication established with ants through pheromone trail.
- Paths deposited more pheromone preferred previously.
- Pheromone trail on short paths increase more rapidly.

Addition of new specifications to this new technique is below:

- They live in an environment where time is discrete.
- They will not be completely blind, they will reach the details about the problem.
- They will keep information formed for the solution of the problem with has some memory.

As shown in Figure 1-a, ants start from their nest and goes along a linear path through the food source.

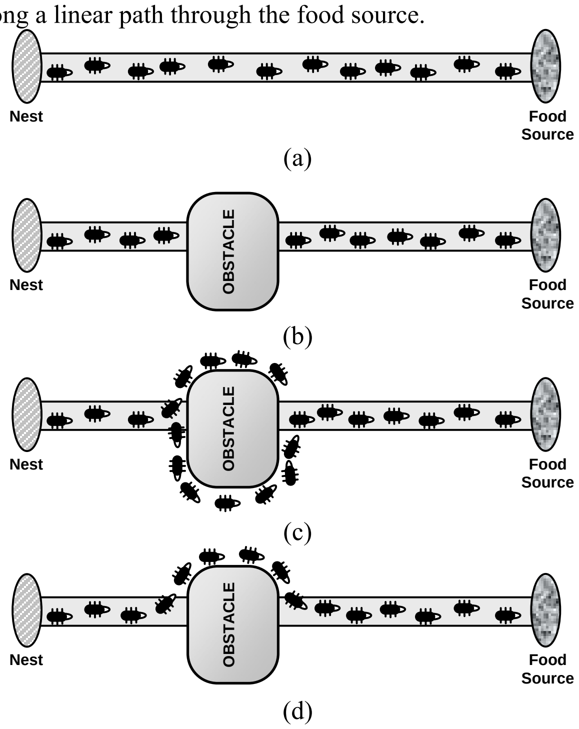

Fig. 1 Behaviors of real ants between their nest and food source

(a) Ants following a path between their nest and food source

(b) Encountering a obstacle of ants

(c) Selection of ants

(d) Finding the shortest path of ants

Actually, if there exists a difficulty on the path while going to the food source (Figure 1-b), ant lying in front of this difficulty cannot continue and has to account a preference for the new outgoing path. In the present case, selection probability of the new direction alternatives of ants is equal. In other words, if ant can select anyone of the right and left directions, the selection chance of these directions is equal (Figure 1-c). Namely, two ants start from their nest in the search of food source at the same time to these two directions. One of them chooses the path that turns out to be shorter while the other takes the longer path. But it is observed that following ants mostly select the shorter path because of the pheromone concentration deposited mostly on the shorter one.

The ant moving in the shorter path returns to the nest earlier and the pheromone deposited in this path is obviously more than what is deposited in the longer path. Other ants in the nest thus have high probability of following the shorter route. These ants also deposit their own pheromone on this path. More and more ants are soon attracted to this path and hence the optimal route from the nest to the food source and back is very quickly established. Such a pheromone-meditated cooperative search process leads to the intelligent swarm behavior.

The instrument of ants uses to find the shortest path is pheromone. Pheromone is a chemical secretion used by some animals to affect their own species. Ant deposit some pheromone while moving, they deposit some amount of pheromone and they prefer the way deposited more pheromone than the other one with a method based on probability. Ants leave the pheromone on the selected path while going to the food source, so they help following ants on the selection of the path (Figure 1-d).

There are many algorithms derived from ant colony meta-heuristic and they are used on solution of many problems. These algorithms are derived from each other as formulation but all use the common specifications of ant colony meta-heuristic.

Generally, in ant colony optimization algorithms, operations described above are iterated in main loop until a certain number of iterations are completed or all ants begin to generate the same result. This situation is named as stagnation behavior, because after a point, algorithm finishes to generate alternative solutions. The reason of this situation is, after a certain number of iterations, ants generate continuously the same solutions because pheromone amount intensifies in some points and the difference between pheromone concentrations on paths become very huge.

Most ant colony optimization algorithms use this algorithmic diagram demonstrated below [15]:

Initiation of the parameters which determines the pheromone trail

While (until result conditions supplied) **do**

    Generate Solutions
    Apply Local Search
    Update Pheromone Trail

End

ACO for multi-objective problem

Recently, different papers have introduced ACO algorithms for multi-objective problems. These algorithms mainly differ with respect to the three following points.

Pheromone trails. The quantity of pheromone laying on a component represents the past experience of the colony with respect to choosing this component. When there is only one objective function, this past experience is defined with respect to this objective. However, when there are several objectives, one may consider two different strategies. A first strategy is to consider a single pheromone structure, as proposed in [16, 17, 18, 19]. In this case, the quantity of pheromone laid by ants is defined with respect to an aggregation of the different objectives. A second strategy is to consider several pheromone structures, as proposed in [20, 21, 22, 23, 24]. In this case, one usually associates a different colony of ants with each different objective, each colony having its own pheromone structure.

Solutions to reward. When updating pheromone trails, one has to decide on which of the constructed solutions laying pheromone. A first possibility is to reward solutions that find the best values for each criterion in the current cycle, as proposed in [21, 22, 24]. A second possibility is to reward every non-dominated solution of the current cycle. In this case, one may either reward all the solutions in the Pareto set, as proposed in [19] or only the new non-dominated solutions that enter in the set in the current cycle, as proposed in [20].

Definition of heuristic factors. When constructing solutions, at each step a candidate is chosen relatively to a transition probability which depends on two factors: a pheromone factor and a heuristic factor. The definition of the pheromone factor depends on the definition of the pheromone trails, as discussed in the first point. For the definition of the heuristic factor, two different strategies have been considered. A first strategy is to consider an aggregation of the different objectives into a single heuristic information, as proposed in [17, 21, 23]. A second strategy is to consider each different objective separately, as proposed in [16, 20, 22, 19, 24]. In this case, there is usually one different colony for each objective.

III. PROBLEM STATEMENTS AND FORMULATIONS

In many real-life optimization problems there are several objectives to optimize. For such multi-objective problems, there is not usually a single best solution but a set of solutions that are superior to others when considering all objectives [25]. In this article, we propose an approach based on Ant Colony Optimization (ACO), and describe the main features of ACO algorithms for solving multi-objective production problems. The problem can be mathematically formulated as follows:

Lead time equation

$$f_1 = \sum_{i=1}^{n}(S_i + P_i) + \sum_{j=1}^{m} T_j + \sum_{k=1}^{l} W_k + \sum_{x=1}^{y} O_x \qquad (1)$$

Work in Process equation

$$f_2 = \sum_{c=1}^{d} E_c = DOS \times plan = DOS \times 21 \qquad (2)$$

Multi-objective function

$$Minimize \quad F = f_1 \cdot \alpha + f_2 \cdot (1-\alpha) \qquad (3)$$

Constraints

$$\sum_{c=1}^{d} E_c \le \max WIP \qquad (4)$$

$$T_j = s/v \qquad (5)$$

$$\sum_{i=1}^{n} P_i = \min{}_{P_i} \le x_{P_i} \le \max{}_{P_i} \qquad (6)$$

$$\sum_{i=1}^{n} S_i = \min{}_{S_i} \le x_{S_i} \le \max{}_{S_i} \qquad (7)$$

Where

S_i	=	set up time
P_i	=	process time
T_j	=	transfer time between each station
W_k	=	process wait time
O_x	=	other time such as communication
c	=	machine or work station that has WIP
E_c	=	total WIP in the system
DOS	=	days of supply
	=	ability to support parts in a day
$plan$	=	working hours per day in this case is 21 hours
α	=	weight of factor to balance lead time and work in process variation and $0 \le \alpha . \le 1$
v	=	transference velocity
s	=	distance between each station or machine
$\min_{P_i}$	=	longest process time
$\max_{S_i}$	=	shortest process time
x_{P_i}	=	standard process time
$\min_{S_i}$	=	longest set up time
$\max_{S_i}$	=	shortest set up time
x_{S_i}	=	standard set up time

The lead time variation equation in (1) computes total variations of lead time in the system. Equation (2) calculates the amount of work in process. The objective function (3) is to minimize lead time variation and WIP moves based on the weights $\alpha.$ and 1-$\alpha.$ given for both objectives. The constraint (4) and (5) ensure that total WIP must not exceed max WIP to store in the stock area and transfer time can calculate from distance between each station divided by velocity, respectively. The constraint (6) indicates that standard process time must be between both shortest and longest process time, respectively. Similarly, the constraint (7) assures that standard set up time must be between both shortest and longest set up time, respectively.

IV. MAX-MIN ANT SYSTEM

Max-Min Ant System (MMAS) suggested by Stutzle and Hoos [26] is yet another method which employs the idea of elitism to introduce exploitation into the original ant system. The provision of exploitation is made in MMAS by the addition of pheromone to only the iteration-best ant's path at the end of each iteration. To further exploit good information, MMAS uses the global-best solution to update the pheromone trail every T_{gb} iterations. The MMAS updating scheme is then given by:

$$\Delta\tau_{ij}(t) = \Delta\tau_{ij}^{ib}(t) + \Delta\tau_{ij}^{gb}(t) I_N\{t / T_{gb}\} \quad (8)$$

Where, N is the set of natural numbers and $\Delta\tau_{ij}^{ib}(t)$ and $\Delta\tau_{ij}^{gb}(t)$ are the pheromone addition given by the iteration-best and global-best ants, respectively.

Premature convergence to sub-optimal solutions is an issue that can be experienced by all ACO algorithms, especially those that use an elitist strategy of pheromone updating. To overcome this problem whilst still allowing for exploitation, Stutzle and Hoos [26] proposed the provision of dynamically evolving bounds on the pheromone trail intensities such that the pheromone intensity on all paths is always within a specified range. As a result all paths will have a non-trivial probability of being selected and thus wider exploration of the search space is encouraged. MMAS uses upper and lower bounds to ensure that pheromone intensities lie within a given range which is calculated based on some analytical reasoning. The upper pheromone bound at iteration t is given by [26]:

$$\tau_{\max}(t) = \frac{1}{1-\rho} \frac{R}{f(\varphi)^{gb}} \quad (9)$$

This expression is equivalent to the asymptotic pheromone limit of an option receiving pheromone addition of $R / f(\varphi)^{gb}$ and decaying by a factor of $1-\rho$ at the end of each iteration. The upper bound as defined in Eq. 6 was found to be of lesser importance while the lower limit played a more decisive role. Stutzle and Hoos [26] introduced the following formula for the calculation of the lower trail strength limit based on some analytical arguments:

$$\tau_{\min} = \frac{\tau_{\max}.(1-p^{dec})}{(J_{avg}-1).p^{dec}} \;,\; p^{dec} = (p^{best})^{1/n} \quad (10)$$

Where $\tau_{\min}$ represents the lower limit for the pheromone trail strength; p^{dec} is the probability that an ant constructs each component of the best solution again; p^{best} is the probability that the best solution is constructed again and J_{avg} is average number of options available at decision points of the problem. MMAS as formulated in Stutzle and Hoos [26], also incorporates another mechanism known as pheromone trail smoothing (PTS). This mechanism reduces the relative difference between the pheromone intensities, and further encourages exploration. The PTS operation performed at the end of each iteration is given by

$$\tau_{ij}(t) \leftarrow \tau_{ij}(t) + \delta(\tau_{\max}(t) - \tau_{ij}(t)) \quad (11)$$

where $0 \le \delta \le 1$ is the PTS coefficient. If $\delta = 0$ the PTS mechanism has no effect, whereas if $\delta = 1$ all pheromone trails are scaled up to $\tau_{\max}(t)$. In addition to these additional mechanisms, MMAS uses the same decision policy as AS.

The max-min ant system is apply to multi-objective function of lead time and work in process that can show in pseudo code as follow this picture

```
Start
   Define original pheromone
        For iteration=1 to i = (1,2,3,...n)
             For ant=1 to i = (1,2,3,...n)
                  Random start node
                       Do while node < i
                            Select next node
                       Loop
             Next 'End loop ant
             Calculate Multi-objective function
             For ant=1 to i=(1,2,3,...n)
                       Do while node <i
                            Update pheromone
                       Loop
                       Checking pheromone upper and pheromone lower
             Next 'End loop ant
        Next 'End loop iteration
```

Fig. 2 The pseudo code of the max-min ant system for machine cell formation

V. EXPERIMENT RESULTS AND DISCUSSION

Plots of the residuals, which are the differences between the observed and predicted values of the response variable, are very useful to check the quality of the fit. Graphical analysis of the residuals is the single most important technique for determining the need for model refinement or for verifying that the underlying assumptions of the analysis are met. Further residual diagnostic plots are shown below.

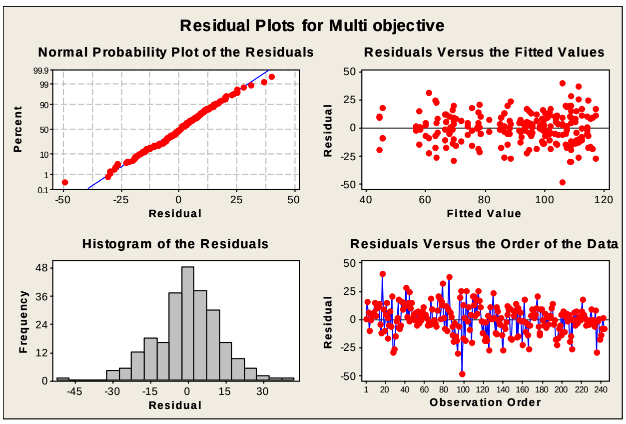

Fig. 3 Residual Plots of multi-objective production problem

Use residual plots, available with many statistical commands, to check statistical assumptions:

- Normal probability plot—to detect nonnormality. An approximately straight line indicates that the residuals are normally distributed.

• Histogram of the residuals—to detect multiple peaks, outliers, and nonnormality. The histogram should be approximately symmetric and bell-shaped.

• Residuals versus the fitted values—to detect nonconstant variance, missing higher-order terms, and outliers. The residuals should be scattered randomly around zero.

• Residuals versus order—to detect time-dependence of residuals. The residuals should exhibit no clear pattern.

For the production data, the four-in-one residual plots indicate no violations of statistical assumptions. The one-way ANOVA model fits the data reasonably well.

The experiments have been run on Intel Core 2, 1.66 GHz and 1.50 GB of RAM. In order to evaluated the performance of the algorithm. The proposed max-min ant system algorithm contains four parameters namely, the number of ants/ the number of iterations (A/I), Pheromone weight (α), Heuristic information weight (β), and Pheromone evaporation weight (ρ). Experiments based on design of experiment approach conducted to find the suitable values for parameters. The results of the performances of the algorithm that in paper is used design of experiment in 3 levels. The results are showed in this picture.

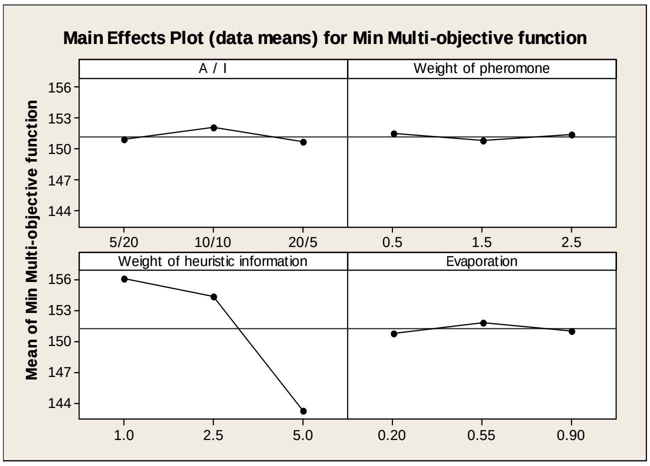

Fig. 4 Main effect plots of the computational result obtained from max-min ant system

Based on these experiments, appropriate values for these parameters were A/I= 10/10, α =1.5, β =5 and ρ = 0.55 that it consider with interaction factor.

VI. Conclusion

In this paper, a meta-heuristic method namely, max-min ant system algorithm is proposed to solve multi-objective production problems. The results from computational experiment of test problem show that develop the lead time and work in process. Further research direction will analyze algorithm performance in greater detail in order to get a better understanding of how benefits are attained from using the adaptive strategy.

Acknowledgement

This study was conducted with the Graduate School at Chiang Mai University (CMU), Thailand in collaboration with the faculty of engineering Chiang Mai University, Chiang Mai, Thailand and supported by National Electronics and Computer Technology Center, National Science and Technology Development Agency and Industry/University Cooperative Research Center (I/UCRC) in HDD Component, the Faculty of Engineering, Khon Kaen University.

References

[1] Blum, C. & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparisons. ACM Computing Surveys, 35(3), 268-308.
[2] Dorigo M., Birattari M. & Stützle T. (2006) Ant Colony optimization: Artificial Ants as Computational Intelligence Technique. IEEE Computational Intelligence Magazine, Nov. 2006.
[3] Blum, C. & Dorigo, M. (2005). Search bias in ant colony optimization: On the role of competition-balanced systems. IEEE Trans. on Evolutionary Computation, 9(2), 159-174.
[4] Dorigo, M. & Stützle, T. (2002). The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances. In: F. Glover and G. Kochenberger (Eds.), Handbook of Metaheuristics. Kluwer Academic Publishers.
[5] Dorigo, M., Maniezzo, V. & Colorni, A. (1996). The ant system: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B, 26(1), 29-41.
[6] Dorigo, M. & Gambardella, L. (1996). Ant Colonies for the Traveling Salesman Problem. Technical Report IRIDIA/1996-3, Universit´e Libre de Bruxelles.
[7] Dorigo, M. & Gambardella, L. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Trans. on Evolutionary Computation, 1(1), 53-66.
[8] Stützle T., & Hoos H. (2000). Max-Min Ant System. Journal of Future Generation Computer Systems, 16, 889 – 914.
[9] Bullnheimer, B., Hartl, R. F., & Strauss, C. (1999). A new ranked-based version of the ant system: a computational study. Central European Journal for Operations Research and Economics, 7(1), 25-38.
[10] Cordón O, Herrera F, Fernández de Viana I, Moreno L. A new ACO model integrating evolutionary computation concepts: The best–worst ant system. In: Proc Second IntWorkshop on Ant Algorithms, Brussels, Belgium, 2000. pp 22–29.
[11] Cordón O, Fernández de Viana I, Herrera F. Analysis of the best–worst ant system and its variants on the TSP. Mathware Soft Comput 2002;9:177–192.
[12] Cordón O, Fernández de Viana I, Herrera F. Analysis of the best–worst ant system and its variants on the QAP. In Dorigo M, Di Caro G, Sampels M, editors. Ant algorithms. LNCS 2463. Heidelberg: Springer-Verlag; 2002. pp 228–234.
[13] DI CARO, G., DORIGO, M., "Extending AntNet for Best-effort Quality-of-Services Routing", Ant Workshop on Ant Colony Optimization, htpp://iridia.ulb.ac.be/ants98/ants98.html, 15-16, 1998.
[14] DALKILIÇ, G., TÜRKMEN, F., "Karınca Kolonisi Optimizasyonu", YPBS2002 – Yüksek Performanslı Bilişim Sempozyumu, Kocaeli, Ekim 2002
[15] VRAHATIS, M.N., BOUTSINAS, B., ALEVIZOS, P., PAVLIDES, G., "The new k-windows algorithm for improving the k-means clustering algorithm", Journal of Complexity 18, pp. 375–391,2002.
[16] C. E. Mariano, E. Morales, A Multiple Objective Ant-Q Algorithm for the Design of Water Distribution Irrigation Networks, Technical Report n HC-9904, Instituto Mexicano de Tecnologa del Agua, Mexico, June, 1999.
[17] M. Gravel, W. L. Price, C. Gagn´e, Scheduling Continuous Casting of Aluminium using a Multiple Objective Ant Colony Optimization Metaheuristic, European Journal of Operational Research, vol. 143, n1, 2002, p. 218-229.

[18] P. R. McMullen, An Ant Colony Optimization Approach to Addressing a JIT Sequencing Problem with Multiple Objectives, Artificial Intelligence in Engineering, vol. 15, n3, 2001, p. 309-317.
[19] B. Bar´an, M. Schaerer, A Multiobjective Ant Colony System for Vehicle Routing Problem with TimeWindows, Proc.Twenty first IASTED International Conference on Applied Informatics, Insbruck, Austria, 2003, p. 97-102.
[20] S. Iredi, D. Merkle, M. Middendorf, Bi-Criterion Optimization with Multi Colony Ant Algorithms, First International Conference on Evolutionary Multi-criterion Optimization (EMO'01), vol. 1993, Lecture Notes in Computer Science, 2001, p. 359-372.
[21] K. Doerner, W. J. Gutjahr, R. F. Hartl, C. Strauss, C. Stummer, Pareto Ant Colony Optimization: A Meta-heuristic Approach to Multiobjective Portfolio Selection, Annals of Operations Research, 2004.
[22] L. Gambardella, E.D. Taillard, G. Agazzi, MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows, in F. G. D. Corne, M. Dorigo(ed.), New Ideas in Optimization, McGraw Hill, London, UK, 1999, p. 63-76.
[23] B. Bullnheimer R.F. Hartl C. Strauss, An Improved Ant system Algorithm for the Vehicule Routing Problem, Annals of Operations Research, vol. 89, 1999, p. 319-328.
[24] K. Doerner, R. F. Hartl, M. Teimann, Are COMPETants More Competent for Problem Solving? The Case of Full Truckload Transportation, Central European Journal of Operations Research, vol. 11, n2, 2003, p. 115-141.
[25] Alaya, I. Solnon, C. Ghedira, K., Ant Colony Optimization for Multi-Objective Optimization Problems, 19th IEEE International Conference on Tools with Artificial Intelligence, Vol. 1, 450-457, 2007.
[26] Stutzle, T., and Hoos, H. H. (2000). MAX-MIN ant system. Future Generation Comput. Systems, 16, 889-914.