

# An Ant Colony System Metaheuristic Algorithm for Solving a Bi-Objective Purchasing Scheduling Problem

José Francisco Delgado Orta<sup>1</sup>, José Antonio Coronel Hernández<sup>1</sup>, Laura Cruz Reyes<sup>2</sup>,  
Alejandro Palacios Espinosa<sup>3</sup>, Christian Ayala Esquivel<sup>1</sup>,  
Isidro Moctezuma Cantorán<sup>1</sup>, and Jorge Ochoa Somuano<sup>1</sup>

<sup>1</sup> Universidad del Mar, San Pedro Mixtepec, Juquila, Oaxaca,  
Mexico

<sup>2</sup> Instituto Tecnológico de Ciudad Madero, Ciudad Madero, Tamaulipas,  
Mexico

<sup>3</sup> Universidad Autónoma de Baja California Sur, La Paz, Baja California Sur,  
Mexico

fdelgado@zicatela.umar.mx, lexmar171@hotmail.com, cruzreyeslaura@gmail.com,  
palacios@uabcs.mx, ceaaa@hotmail.com, moctezumai@hotmail.com,  
ochoa@zicatela.umar.mx

**Abstract.** In this work the Purchasing Scheduling Problem (PSP) is presented, based in study cases of public and private sectors. PSP models the purchasing process in both scenarios through an optimization approach. It is based on the multi-objective formulation of the knapsack problem. Therefore, PSP is defined as a based-graph problem with two objectives: maximization of satisfied demands and minimization of purchasing costs in a supplying task for inventory systems. In order to achieve these goals, the problem is defined as an integer problem, in which, feasibility of solutions is tested using a profit/cost relationship. This permits to solve PSP as a maximization of a single objective through an Ant Colony System Algorithm (ACS), an efficient solver for graph-based problems. Experimental results reveal that ACS reaches 74% of efficiency on solving instances randomly generated; obtaining purchasing plans as a result. This demonstrates the advantages of using heuristics in decision making systems such as Enterprise Resource Planning (ERP).

**Keywords:** Purchasing scheduling problem, ant colony system algorithm, multi-objective optimization.

## 1 Introduction

The purchasing process of materials and goods is very important for companies and organizations around the world. It is a main task for maintaining inventory control in warehouses which arise from customer demands and whose main objective is to

provide goods to warehouses in a timely manner to satisfy all demands in periodic inventory cycles. When a company decides to purchase materials or equipment, it usually considers a set of variables related to inventory theory: stock levels, delivery times, reorder points and in most cases the availability of money to make purchases. However, according to [1], inventory systems in purchasing departments have poor planning tools and strategies for decision making in inventory tasks, as a consequence of supply chain problem related to inventory is defined as a NP Hard problem, where optimization of tasks is difficult to perform in large-scale instances.

Purchasing is a complex process that involves searches in big catalogs (or huge in the worst-case scenario) or even physical inventories, commonly found in websites or company data warehouses, which are dependent to needs in several companies and the available physical accommodation where they are stored. Industrial applications of PSP, has also an economical constrain of available funds and scenarios where a purchasing department implies the work of many people in its different roles since an assistant until the managers. Where they must decide which of inventory variables are the most relevant for the company according to priorities established by customers and prices of products. In this manner, this work approaches a planning stage for purchasing based on an economical approach, which is mathematically modeled and solved using heuristic methods.

## **2 Purchasing Scheduling Problem: Public and Private Problematics**

Purchasing Scheduling Problem (PSP) is related to many companies in industrial, educational, and many other applications, where there are two main areas: private and public sectors. In the private sector, orders are programmed periodically in daily, weekly, monthly or yearly cycles. For each period, the supply manager authorizes company funds so that purchasing assistants can supply the goods according to reorder levels defined for each case. In the public educational sector, it is done in the same manner with the purchasing department authorizing funds to fill orders that must be prepared by personnel for each random period. Purchasing is quite different in the public sector, because it must be done through an open bidding process (defined in [2]) which must meet the rules and laws about purchases in public entities.

Additionally, in public education environment, this process requires order preparation with varying product quantities, which are commonly supplied at different times of the year because funds are not delivered periodically to managers. In addition, needs of customers in both sectors are sometimes unmet due to shipping delays, lack of suppliers' stock; whereas in the public educational sector these issues are present but mainly due to quality factors as schools' purchasing department personnel is not an expert in the technical requirements of the goods to be acquired. In addition, goods require a high description factor because suppliers are not to be taken into account when orders go out for bidding as required in [2].

The pressure for purchasing often leads to bad quality products, as expeditiousness is preferred over quality. In addition, order preparation usually must be done in haste. As a consequence, orders are usually not ready and often lack complete descriptions or specs but are still turned over to the supply department for purchasing. In addition,

complexity elements of described approaches are often a barrier to include mathematical formulations in real-time inventory systems.

### **3 Related Work**

Several applications of PSP present the problem according to the satisfaction of a function of time as a single objective, dependent of other variables; for example, holding times and penalties in project management environment [3], where the problem depends on the availability of products by the suppliers to supply independent items. In [4], is presented a purchasing scheduling in manufacturing systems, based on the computation of delivery times and quantities of product to supply items using special priorities, according to resource capacities, orders priority and leading times. In the related problem of economic lot size scheduling problem [5], time is also a decision factor according to inventory variables. However, a common feature of these approaches is influenced by no linear and unrestricted models to solve them, supposing that funds to supply the items are unlimited and planning is performed in stochastic way. Even though, multi-objective approaches to solve PSP using linear models are not reported in literature. Related works have solved similar multi-objective problems such as networking Flow Shop Scheduling [6] and the Knapsack Problem [7]. For these problems, algorithms have been developed using approaches like Ant Colony based on Pareto's optimization [8], strategies of multi-objective Particle Swarm Optimization [9], Fuzzy Ant Colony Optimization [10] or Genetic Algorithms [11]. In where, problems are formulated as a graph based-representation to generate feasible solutions through linear approaches.

According to [11], solution of multi-objective problems presents a behavior called Pareto optimality (named Pareto's front), in which a solution of a problem is represented by a set of objectives that are commonly in conflict, due to the non-dominance of objective values. In other words, if an objective is approximated to optimal value, remaining objectives can be away of their respective optimal values, or even they may be infeasible.

### **4 The Graph-based Representation of PSP**

The Purchasing Scheduling Problem is defined from a general catalog of products in terms of a graph  $G = (V, E)$ , where  $V = \{P \cup S\}$  consists of a set of  $n$  products ( $P$ ) that must be acquired from  $m$  suppliers ( $S$ ). The set  $E$  consists of pairs  $(p, s)$ , where  $p \in P$  and  $s \in S$ ; each pair has an associated economic cost  $c_{ps}$  of a product to be purchased from any supplier.

Purchasing must be performed through orders, which are defined as subsets  $G_k$ , where  $k$  corresponds to the number of purchasers, who are the decision makers to get the products. Each order  $G_k$  must be formed according to an available fund  $a_k$  for each purchaser. The problem is formulated through two objectives: maximization the amount of products that must be satisfied and minimization of purchasing costs in an inventory cycle. So, these elements are the basis to perform a mathematical approach of PSP.

#### 4.1 The Purchasing Scheduling Formulation

Solution of PSP is performed to represent purchasing plans in an inventory cycle. In this way, PSP is formulated through the next data sets:

- $P$ : is the set of products to be supplied, in other words an inventory catalog with  $n$  products.
- $S$ : is the set of suppliers of the product catalogue, requested by a purchaser with  $m$  suppliers.
- $A$ : is a set of available funds for  $s$  purchasers, with a number  $a_k$  for each order  $k$ , where each non negative fund  $a_k$  is assigned to a purchaser.

PSP formulation uses the next variables:

- $c_{ij}$  is the economic cost for each product  $i$  of the supplier  $j$ .
- $x_{ijk}$  is a decision integer variable with values  $\{0,1\}$ , one if the product  $i$  is assigned to the supplier  $j$  in the order  $k$ , zero in otherwise.

In the context of the problem,  $f$  and  $g$  values are introduced to normalize objectives values of PSP in the domain  $[0,1]$ , as the multi-objective knapsack problem [7]. Where  $f$  represents a profit in terms of assigned products, while  $g$  is a vector that indicates a reference of an assigned product with regard to the  $c_{ij}$  values of products to be supplied in an inventory cycle. These values are based on the utility principle proposed in [7], [14], [15], which is defined in terms of PSP through expressions (1) and (2).

$$\max f = 1 - \frac{1}{k} \sum_{k=1}^s \left[ \frac{\sum_{j=1}^m \sum_{i=1}^n x_{ijk}}{|G_k|} \right] \quad (1)$$

$$\min g = \frac{1}{k} \sum_{k=1}^s \left[ \frac{\sum_{j=1}^m \sum_{i=1}^n c_{ij} x_{ijk}}{\sum_{j=1}^m \sum_{i=1}^n c_{ij}} \right] \quad (2)$$

These values define a profit/weight relationship that heuristics are able to exploit for objectives of PSP. Therefore, the normalized  $g$  value is transformed to solve the multi-objective PSP as a maximization problem in the next formulation:

$$\max z = f - g \quad (3)$$

Subject to:

$$\sum_{j=1}^m \sum_{i=1}^n c_{ij} x_{ijk} \leq a_k \quad k = 1, 2, \dots, s \quad (4)$$

$$x_{ijk} \in \{0,1\} \quad i = 1, 2, \dots, n, j = 1, 2, \dots, m, k = 1, 2, \dots, s \quad (5)$$

Objectives of the problem are related with equation (3). Equations (4) and (5) verify constrains of assigned funds and with regard to the available resources according to integer values  $x_{ijk}$ . This model is solved with an ant colony system algorithm, which obtains feasible solutions according to the related ant approach.

## 5 The Ant Colony System Algorithm

Ants in their evolutionary process, search feasible solutions of minimal cost for a problem. During the execution of an ant algorithm, an ant located in a state (a selected product  $i$ , has to choose the next supplier  $j$  to be assigned. Suppliers are chosen from among the set of unassigned products in the rest of suppliers. Selection of the next supplier  $j$  uses a probability value  $q_0$ , which is used in deterministic or random approaches. In a deterministic form, selecting the supplier with the greater amount of pheromone  $\tau$ , following an exploitation of the obtained knowledge of the problem  $\eta$  using the neighborhood  $N_k(i)$  for a selected product  $i$ ; while in random selection, a rule of exploration is used. Both focuses are represented in equation (6).

$$j = \begin{cases} \operatorname{argmax}_{u \in N_k(i)} \{ [\tau(i, j)] \times [n(i, j)^\beta] \} & q < q_0 \\ f(p_{ij}^k) & \text{otherwise} \end{cases} \quad (6)$$

Where  $q$  is a uniformly distributed random number between  $[0,1]$ ,  $q_0$  is a constructive parameter  $0 < q_0 < 1$ , which represents a probability value and  $f$  is a function based on the well-known roulette technique, which selects the supplier for a product with a probability proportional to  $f(p_{ij}^k)$ , obtained as a result of the computation of expression (7). The basic idea of this algorithm is to choose different suppliers at each iteration. Heuristic information  $n_{ij}$  is the previous knowledge, given through the inverse of accumulated cost over the pairs  $(i, j)$ . Heuristic information  $n_{ij}$  of equation and pheromone trails  $\tau_{ij}$  are used by ACS to guide all ant movements.

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}][n_{ij}]^\beta}{\sum_{l \in N_k^i} [\tau_{il}][n_{il}]^\beta} & j \in N_k^i \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

Where  $p_{ij}^k$  is the probability value, associated for an ant  $k$  that selects a product  $i$  from a supplier  $j$ .  $N_k^i$  is the possible neighborhood for an ant  $k$  when it choose a product  $i$ ; in other words, all reachable suppliers. Pheromone trails, deposited by ants evaporate in local and global forms. Local evaporation is given through equation (8), where  $\tau_0$  is the initial value of the pheromone trail and  $\rho$  ( $0 < \rho < 1$ ) is the rate of local dissipation.

$$\tau_0 \leftarrow (1 - \rho)\tau_{ij} + \rho\tau_0 \quad (8)$$

Evaporation of global pheromone follows the pattern of behavior given by equation (9).

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \rho\Delta\tau_0, \quad \forall (i, j) \in T_k \quad (9)$$

Where  $\Delta\tau_0 = 1/C_k$  represents the amount of deposited pheromone and  $\rho$  is a rate of global evaporation, and  $C_k$  is the accumulated resources of the packed products performed by an ant for all arcs in an order  $T_k$ . Artificial ants always try to build feasible solutions using pheromone trails; however, infeasible solutions also can be obtained [12]. General algorithm ACS is shown in Figure 1.

AntColonySystem procedure consists of the construction of ants and activities for these and global evaporation of pheromone trails. Construction of a colony is performed through the *Generate\_ants\_and\_activities* procedure, shown in Figure 2.

1	Procedure AntColonySystem ()
2	Initialize_parameters
4	While(stop_criteria_is_reached) do
5	Generate_ants_and_activities()
6	Evaporation_pheromone_trails()
7	End_of_while
8	End Procedure

**Fig. 1.** The AntColonySystem Procedure.

1	Procedure Generate_ants_and_activities ()
2	For each ant $k$ until $m$
3	$i = \text{current\_product}$
4	$S_k = i$
5	$L_k = i$
6	While (current_state is not equal to objective_state) do
7	For each $j \in N_k(i)$
8	$b_{ij} = \tau_{ij} \times n_{ij}^\beta$
9	$q = \text{generate\_random\_number}[0,1]$
10	If ( $q < q_0$ ) then
11	$j = \max(b_{ij}, N_i^k)$
12	Else
13	For each $j \in N_i^k$
14	$p_{ij}^k = \frac{b_{ij}}{\sum_{l \in N_i^k} b_{il}}$
15	End For
16	$j = f(P, N_i^k)$
17	End If
18	$i = \text{next\_product}$
19	$S_k = \langle S_{kj} \rangle$
20	$\tau_0 \leftarrow (1 - \rho)\tau_{ij} + \rho\tau_0$
21	$L_k = L_k \cup \{j\}$
22	End_while
23	Next ant $k$
24	End Procedure

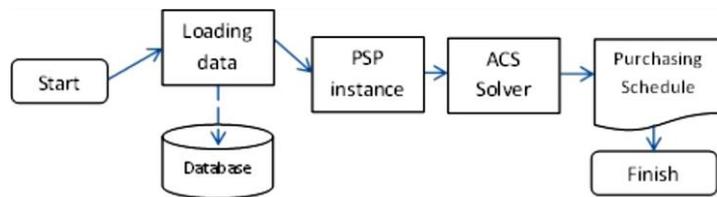
**Fig. 2.** The Generate\_ants\_and\_activities procedure.

Algorithm in Figure 3, is used with the described components of ACS (lines 2-23), for each ant  $k$ . Lines 4-6 choose the current product and initializes the solution. Selection of the correspondent supplier  $j$  for a product  $i$  is performed in lines 7-18 according to the defined transition rules. Line 16 obtains the supplier  $j$  for the current product  $i$  using a policy of decision, in which the values of pheromone trails for the current product and restrictions of the problem are considered according to the  $P$  values of equation (6). Lines 18-20 perform the state transition, the upgrading of a

solution  $S_k$  and the evaporation of pheromone trails  $\tau_{ij}$ . Line 21 includes to the algorithm a memory  $L_k$ , which stores the current assignment. It is used to build valid solutions, to evaluate a generated solution and to make a reallocation for an ant if a supplier  $j$  chosen for a product, makes infeasible the current solution.

## 6 Architecture of the Solution

The described ACS is proposed according to the architecture of Figure 3, which describes the construction of a purchasing schedule.



**Fig. 3.** Proposed solution methodology.

This architecture is designed validate the performance of heuristic approach solving PSP. For this reason, it is supposed that there is an inventory database, which contains information about the sets defined in the PSP formulation, this information is read to a model database described in [13]. Once that sets are read to the database, input sets are placed in a PSP instance. These inputs consist of a plain-text file that makes the solver independent to the database. This permits the use of the methodology in several purchasing scenes. PSP inputs are used by the solver to build purchasing plans according to the objectives through the described ACS algorithm, defined in terms of an inventory cycle.

## 7 Experiments and Results

Due to real instances for the problem were not available, a dataset of ten orders was built using a pseudo-random number generator. It uses the queries of web catalogs, stored in a database. The generator creates the orders with different prices and suppliers for products, maximum and minimum prices for products and available funds, parameters of generator are shown in Table 1. The ten generated orders (it is supposed an inventory cycle with ten purchasers) are the input sets for PSP, whole products have sometimes costs more expensive than available funds.

Instances were solved with ACS algorithm in a server Dell T710, Xeon four-core processor (2.2 Ghz per core), 48 GB of RAM memory, 500 GB hard disk under Debian Linux 6.0. Algorithm was developed in Java Standard Edition 7 with Eclipse Helios. Pheromone trails of ACS have initial values of 0.005, a parameter that according with [12], which permits enough exploration. Evaporation rate  $\rho$  is 0.5, and  $q$  value of ACS is 0.5, which permits to have 50% of exploration and 50% of exploitation. Algorithm design consists of 20 generations and ten ants per colony with

the  $\beta$  value of 1. In each execution, an accumulated sum ( $\Sigma$ ) is stored with the average of the values of best solutions.

**Table 1.** Input parameters for instances generated for PSP.

Order	Quantity of Products	Min price	Max price	Available Funds
1	126	14.00	10999.00	25000.00
2	123	73.00	60000.00	50000.00
3	63	29.00	120000.00	40000.00
4	146	99.90	15980.00	65000.00
5	70	3.00	60000.00	30000.00
6	194	56.90	20000.00	80000.00
7	128	75.00	18799.00	75000.00
8	119	95.00	18000.00	55000.00
9	108	3.00	88996.00	48000.00
10	126	14.00	11499.00	40500.00

**Table 2.** Results of ACS algorithm solving PSP.

Iterations	$\Sigma$	$f$	$g$	$z$	$t$
1000	22.26	0.8436	0.1027	0.7408	585.15
5000	24.17	0.8412	0.1030	0.7382	4753.96
10000	23.32	0.8395	0.1030	0.7406	5830.87
15000	26.47	0.8425	0.1029	0.7407	6840.30
20000	25.22	0.8431	0.1030	0.7408	8963.43
30000	25.53	0.8441	0.1030	0.7409	10732.78

It indicates a measurement of heuristic information with regard of the solution space (the exploration degree of the algorithm). Results of ACS of table 2 show also the average time ( $t$ ) in which ACS obtains the best solutions and the  $z$  value for each instance. Table 2 shows the performance for ACS for six tests with 30 executions with: 1000, 5000, 10000, 15000, 20000 and 30000 iterations. ACS has a convergence of 6.2 seconds in average, with a profit coefficient of 0.8423 (84% in average of normalized  $f$  values) and 0.1030 in cost coefficient (10% in average of  $g$  values). Values PSP objectives reaches 0.7403 in average (74%) of efficiency in the multi-objective problem.

It occurs because of ACS has finished its executions before assignments of products have been completed; as a result of many remaining products have expensive costs that make infeasible equation (2) and funds can not completely be assigned. In the same way, summation column has a coefficient of variation of (0.6%). It reveals that adjusting of ACS parameters can be recommended to explore

new solutions. Best values can be obtained because of results of PSP objectives ( $f$  and  $g$  values) present a Pearson coefficient of correlation of -0.3826, which demonstrates the existence of Pareto's front in solutions of PSP.

## **8 Conclusions and Future Works**

The purchasing scheduling problem was presented in this work with an economic approach in a combinatory scene. It was feasibly solved through an ant colony system algorithm and a heuristic based on the multi-objective knapsack problem. In this way, results of the developed ACS can be tested using other neighborhood techniques, such as 2-Opt and its variants or Cross Exchange. These strategies can be used to generate a bigger subset of feasible solutions to compute Pareto's optimal values for PSP. In same way, problem can be solved using other heuristics as taboo search, simulated annealing or genetic algorithms or they can be improved through hybridization of the developed ant colony approach. Optimality degree of PSP and speed of computation shows the advantages of heuristic approaches to integrate them in making decision systems can be used as planning tools to develop ERP systems (Enterprise Resource Planning), which can be applied in industrial applications to optimize productive processes in many companies.

## **References**

1. Herroelen, W.: Project scheduling—Theory and practice. *Production and Operations Management*, 14(4), pp. 413–432 (2005)
2. Cámara de Diputados del H. Congreso de la Unión.: *Ley de Adquisiciones, Arrendamientos y Servicios del Sector Público. Reforma* (2011)
3. Ronen, B., & Trietsch, D.: Optimal scheduling of purchasing orders for large projects. *European journal of operational research*, 68(2), pp. 185–195 (1993)
4. Huang, P., Li, H., & Han, L.: Order scheduling problems in make-to-order manufacturing systems. In *Mechatronics and Automation, 2005 IEEE International Conference*. Vol. 4, pp. 2179–2184, IEEE (2005)
5. Sarker, R., & Newton, C.: A genetic algorithm for solving economic lot size scheduling problem. *Computers & Industrial Engineering*, 42(2), pp. 189–198 (2002)
6. Graham, R. L.: Bounds for certain multiprocessing anomalies. *Bell System Technical Journal*, 45(9), pp. 1563–1581 (1966)
7. Kumar, R., Joshi, A. H., Banka, K. K., & Rockett, P. I.: Evolution of hyperheuristics for the biobjective 0/1 knapsack problem by multiobjective genetic programming. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*. pp. 1227–1234. ACM (2008)
8. Doerner, K., Gutjahr, W. J., Hartl, R. F., Strauss, C., & Stummer, C.: Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. *Annals of Operations Research*, 131(1–4), pp. 79–99 (2004)

9. Reyes-Sierra, M., & Coello, C. C.: Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International journal of computational intelligence research*, 2(3), pp. 287–308 (2006)
10. George, A., & Rajakumar, B. R.: Fuzzy aided ant colony optimization algorithm to solve optimization problem. In *Intelligent Informatics*, pp. 207–215, Springer Berlin Heidelberg (2013)
11. Coello, C. C., Lamont, G. B., & Van Veldhuizen, D. A.: *Evolutionary algorithms for solving multi-objective problems*. Springer (2007)
12. Dorigo, M., & Gambardella, L. M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. *Evolutionary Computation, IEEE Transactions on*, 1(1), pp. 53–66 (1997)
13. Coronel, J.A.: *Solution of Assignment Problem in Purchasing of Goods Using Metaheuristic Algorithms*. Bachelor Thesis. Universidad del Mar campus Puerto Escondido (2014)
14. Falkenauer, E.: A hybrid grouping genetic algorithm for bin packing. *Journal of heuristics*, 2(1), pp. 5–30 (1996)
15. Martello, S., & Toth, P.: Lower bounds and reduction procedures for the bin packing problem. *Discrete Applied Mathematics*, 28(1), pp. 59–70 (1990)