Forming Buyer Coalition with Bundles of Items by Ant Colony Optimization

ANON SUKSTRIENWONG
School of science and Technology
Bangkok University
Bangkok
THAILAND
anon.su@bu.ac.th

Abstract: - The study of the buyer coalition has been reviewed by researchers for decades. However, there are few schemes applying ant colony optimization (ACO) for forming buyer coalition. In this paper, the approach called the Ant Colony for Buyer Coalition (ACBC) algorithm is proposed to form the buyer coalition with bundles of items. The main endeavor of the algorithm is to partition the whole group of buyers into smaller groups in order to achieve a common goal. The artificial ants search to find best disjoint subgroups of all buyers based on the total utility functions. The experimental results are compared with the genetic algorithm (GAs) in the terms of the global optimal solution. It indicates that the algorithm can improve the total discount of the buyer coalition.

Key-Words: - ant colony optimization; coalition structure; group formation.

1 Introduction
Several commercial websites such as http://www.amazon.com and http://www.staples.com prefer to sell their product on the electronic marketplaces because it helps selling tons of products in few days. Most sellers also prefer to offer a volume discount for customers if a number of selling is big. In common, buyers want to obtain a deduction from the price list offered by sellers in return for payment. One common shopping tactic which most buyers are likely to make is a group buying because a large group of buyers gains more negotiating power. A buyer coalition is set of buyers who agree to join together to bargain with sellers, so buyers can gain volume discount prices. Of course, bargaining is one of the traditional strategies for buyers and seller to reach beneficial agreements.

The other strategy is the buyer coalition scheme. Several buyer coalition schemes exist with the aim of having the best group utility [10, 12,13,14]. However, few schemes consider forming group of buyer with bundles of items which can be often occurring in the real world. There are several opportunities that it can be happened, such as a case that buyers cannot purchase the bundles of items by their own because the packages of products sold by sellers are compose of hundreds of items or multiple type of items.

The earlier algorithm in [11] called GroupPackageString scheme applies genetic algorithms (GAs) to form buyer coalitions with bundle of items. However, this algorithm does not consider the situation of partitioning the whole group of buyers into smaller groups. The partitioned group is called a coalition structure (CS). Some researchers have developed and evaluated the performance of anytime CSG algorithms to search for optimal coalition structures in characteristic function games (CFGs) [1, 2, 6]. It is also applied in many complex autonomous applications as electronic marketplace [3, 7, 8]. The CS aims to maximize the utility of the coalitions, but often the number of coalition structures is too large to allow for the exhaustive search for the optimal one [4]. The optimal solution of the problems can result at any of the n levels of the coalition structure. Furthermore, finding optimal coalition structure is NP-complete. The size of the search space is exponential in the number of agents. To guarantee an optimal solution, some papers such as in [4, 5] search the whole space, which is an exhaustive search.

Given a set of m members, \( A = \{a_1, a_2, \ldots, a_m\} \) and a subset or coalition \( C \subseteq A \), there are two challenging stages involving in this paper:

1. Search disjoint subsets of all of which the union of subsets equals \( A \).
2. Calculate the total utility of the whole group.

A CS consists of several agents in the environment grouped into one or more coalitions. In characteristic function game the value of coalition structure, CS, is given by

\[
V(CS) = \sum_{C \in CS} v(C, CS),
\]

(1)

where \( v(C, CS) \) is the value of coalition structure of \( C \in CS \). And, the optimal coalition structure is noted as
\[ CS^* = \arg \max_{CS \in M} V(CS). \]  

In CFG the value of each coalition is given by a characteristic function which is simply defined as the sum of the values of the coalitions that it contains.

Reference [15] shows the calculation of the total number of coalition structures as follow,

\[ \sum_{i=1}^{a} Z(a,i), \]  

where \( a \) is the number of agents and \( Z(a,i) \) is the number of coalition structures with \( i \) coalitions,

\[ Z(a,i) = i^*Z(a-1,i) + Z(a-1,i-1), \]  

\[ Z(a,a) = Z(a,1) = 1. \]

If \( a \) is small, \( a = 4 \), the number of coalition structures is 15, see Fig. 1 (a). However, when number of coalition is bigger, for example \( a = 24 \), the total coalition structures becomes 8388609. This is such a difficult task to search for optimal solution when the number of agent is bigger. So, the algorithm called ACBC is proposed to search for the optimal solution. The proposed algorithm is based on ant colony algorithms (ACO) which are inspired from the behavior of real ants. The major shortcoming of the ACBC is that it provides no guarantee to find the optimal solution because ACO is one of the heuristic functions. However, it seems to work well in our practice.

The paper is divided into 4 sections including this introduction. The rest of the paper is organized as follows. Section 2 describes the motivated example and the detail of the proposed algorithm. The experimental results are in section 3. To guarantee the quality of the algorithm, the experimental results are compared with the GroupPackageString scheme. Finally, the conclusions and future works are in the last section.

## 2 Searching Optimal Buyer Coalition by Ants

This section gives the formal definition of the buyer coalition addressed in this paper. The motivated example of our problem is shown to demonstrate the difficulty of the problem.

### 2.1 The motivated example

Suppose sellers sell three kinds of product, \( x_1 \), \( x_2 \) and \( x_3 \). They prepare a large stock of goods with many attractive prices shown in Table 1. Each seller has made different package based on the number of items. The more items is with a single package, the more discount. The package number 1–3 are a single-item package. The package number 4 of seller 1 comprised of 1 item of \( X_1 \), \( X_2 \) and \( X_3 \) is set to be sold at the price of 54.0 dollars, which is 10% of the original price. The package number 4 of the same seller comprised of more items (10 items of \( X_2 \) and 1 item of \( X_3 \)) is set to be sold at the price of 180 which is 20% discount of the original price. The maximum discount of the seller number 1 is 30\%. The discount policy for the seller number 2 is different for the seller number 1. The seller number 2 has made 5 different packages. The package number 1–3 are also a single-item package. The package number 4 of the seller number 2 is comprised of 50 items of \( X_1 \). It is set to be sold at the price of 525.0 which is about 25\% discount of the original price.

Moreover, if the buyer purchases more items from one seller, this buyer may get free shipping. In electronic marketplaces, many buyers come from different places because the buyers order products from anywhere from the internet. Suppose there are three buyers called \( b_1 \), \( b_2 \) and \( b_3 \). After they have seen the price list of both sellers, they have made their decision to buy some products. Of course, buyers prefer to buy the product as lower as they can. However, they do not want to buy the whole package to get the special price. They have come to join their requests in the group buying. Their requests are shown in the Table 2.

As we can see, the \( b_1 \) and \( b_2 \) are resided in the same area, location \( L_1 \). If they join their requests, the seller sent the whole package to one of them without shipping cost. And, the best package for both \( b_1 \) and \( b_2 \) is package number 5 from the seller number 1. They would pay at most 170.0 dollars. Suppose \( b_3 \) and \( b_4 \) assemble their request to buy the package number 5 from the seller number 2. They need to pay at least 525 dollars including the shipping cost. It is because they live in the different areas. In general case, the seller would send the whole package to one person which has the largest demand without the shipping cost. So, the seller number 2 sends the package number 5 to \( b_1 \).

Then, \( b_3 \) sends 10 items of \( X_2 \) to \( b_4 \) with the cost of 15. The total spending for both \( b_3 \) and \( b_4 \) are 770+15 = 785 dollars. As the buyer coalition is formed, the total spend is 170+785 = 955 dollars. If all buyers unite into only one group, they need to buy 1 set of package number 5 from the seller number 1 and 1 set of the package number 4.
from the seller number 2. The total cost is $170+770 = 940$ dollars. However, buyers are located in different places. The seller sent all products to $b_3$ because $b_3$ has the biggest order. When $b_3$ gets the products, $b_3$ passes 5 items of $X_1$ to $b_1$ with the shipping cost of 10. And, $b_3$ sends 5 items of $X_1$ and 2 items of $X_3$ to $b_2$ with the shipping cost of 10. Finally, $b_3$ sends 10 items of $X_2$ to $b_4$ with the shipping cost of 30 dollars. The total shipping cost is $10+10+15 = 35$ dollars. So, the total spending from forming the group buying is $940+35 = 975$ dollars. As we can see that in this case forming the group buying by partitioned the whole group into smaller groups, $\{b_1,b_2\}$ and $\{b_3,b_4\}$, uses lower cost than forming the whole buyers into one big group $\{b_1,b_2,b_3,b_4\}$, which is about $975-955 = 20$ dollars. The problem is that how can we find the optimal solution. Specially, when the number of buyers is big, the possible number of partitioned groups is also big.

<table>
<thead>
<tr>
<th>Seller</th>
<th>Package</th>
<th>Product (k=3)</th>
<th>Discount of original price (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>$X_1$</td>
<td>$X_2$</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

0' means that the sellers do not put the item in the package.

<table>
<thead>
<tr>
<th>Buyer</th>
<th>Buyer requests (k=3)</th>
<th>Locations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$X_1$</td>
<td>$X_2$</td>
</tr>
<tr>
<td>$b_1$</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>$b_2$</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>$b_3$</td>
<td>0</td>
<td>40</td>
</tr>
<tr>
<td>$b_4$</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

0' means that the buyers do not put the item.

2.2 Problem formulation
In the common ant colony optimization, the problem must be represented as a graph where the optimal solution of buyers can be defined in a certain way through the graph. The proposed algorithm involves partitioning a set of elements into subsets based on the utility function that are associated to each subset. The formulization of the ACBC algorithm can be described as below.

Given a set of buyers $A = \{a_1, a_2, \ldots, a_n\}$, there are two kinds of relationships between two buyers which are represented by edges. If buyer $a_i$ and $a_j$, $i \neq j$, are in the same subgroup, there is a path using solid lines to walk from $a_i$ and $a_j$. But, it is not necessary to have a solid line directly connecting between $a_i$ and $a_j$. Let $A$ is divided into n
different groups \((C_1, C_2, ..., C_n)\) and \(\bigcap_{k=1}^{m} C_k = \emptyset, \bigcup_{k=1}^{m} C_k = A\), where \(1 \leq k \leq m\). There exits a dotted line connecting between \(C_k\) and \(C_l\) where \(k \neq l\). For example, Let a set of \(A = \{1, 2, 3, 4, 5, 6\}\), the coalition structure of 6 buyers are shown in Fig. 1 (b). In our graph seen in Fig. 2 (a), there are exactly \(2^6 - 1 = 10\) lines for each buyer to connect to others. If \(A\) is divided into two subgroups, \(C_1 = \{1, 2, 4\}\) and \(C_2 = \{3, 5, 6\}\), the graph represented the relation among buyers can be shown in Fig. 2(b). Also, it can be represented as Fig 2(c). These graphs can be created during the search by artificial ants so they are called the ACBC graph. If \(A\) is divided into three subgroups, \(C_1 = \{1, 2, 4\}\), \(C_2 = \{3, 6\}\), and \(C_3 = \{5\}\), then the graph represented the relation among buyers can be shown in Fig. 2(d). The example of the ACBC graph of \(\{1, 2, 3\} \{4\} \{5\} \{6\}\) is shown in Fig. 2(e).

(a) Coalition structure of 4 buyers shown in [4]

(b) Coalition structure of 6 buyers

Figure 1. Coalition structures
Figure 2. Representing the possibility of the ACBC graphs for 6 buyers as graphs

The ants construct the path from each buyer to unvisited buyers until all the buyers have been visited. This means that each buyer can be visited only one time during the constructing of the path except for the first buyer. Then, the ACBC graph becomes the closed graph. In addition, the total number of edges for constructing the ACBC graph is $m-1$, where $m$ is the number of buyers. In this paper, the proposed algorithm relies on the assumption that the value of a coalition is independent of other coalitions in the coalition structure. All buyers in the group participate in the process of the algorithm, and each buyer is represented exactly once in the ACBC graph. At the beginning all of the pheromone values of each package are initialized to the very small value $c$, $0 < c \leq 1$. The artificial ant, called ant $m$, chooses all members for finding the best group’s utility on return. After initializing the problem graph with a small amount of pheromones and defining each ant’s starting point, several ants run for a certain number of iterations. The probability of the ant $m$ to choose one member called $i$ to join with the other called $j$ with the relation $k$ (where $k \in T = \{\text{dotted line}, \text{solid line}\}$) is $p_{lh}^{ij}$ defined formally as below:

$$p_{lh}^{ij} = \begin{cases} \sum_{l \in A} \sum_{a \in A \setminus l} \left( \frac{\tau_{il}^{(m)}}{\tau_{il}^{(m)}} \right) \left( \eta_{il}^{(m)} \right)^{\beta} & \text{if } l \in A \text{ and } a_i \text{ has not been selected} \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$
where \( \alpha \) and \( \beta \) are two parameters which determine the relative influence of pheromone trail and heuristic information and \( \Delta r^m_{ij} \) is the amount of pheromone laid on line between \( a_i \) and \( a_j \) on either solid line or dotted line by ant \( m \) defined as follow:

\[
\Delta r^m_{ij} = \begin{cases} 
1(U_{ij}) & \text{if } a_i \text{ and } a_j \text{ were selected by ant } m \text{ with the relation } k, \\
0 & \text{otherwise.}
\end{cases}
\]

If \( k = \) dotted line, then \( C_n = \{ a_i \{ a_j \} \} \). If \( k = \) solid line, \( C_n = \{ a_i a_j \} \).

And, \( \eta^m_k \) is given by:

\[
\eta^m_k = \begin{cases} 
(U'_D - U_{c(\{ a_i \{ a_j \})}) & \text{if } k = 0 \\
(U'_D - U_{c(\{ a_i a_j \)})} & \text{otherwise,}
\end{cases}
\]

where \( D \) is the constant value, and both \( U_{c(\{ a_i \{ a_j \})} \) and \( U_{c(\{ a_i a_j \)} \) are derived by (8) and (9). The pheromone \( \tau_{ij} \), associated with the line joining \( a_i \) and \( a_j \), is updated as follow:

\[
\tau_{ij} \leftarrow (1-\rho) \cdot \tau_{ij} + \sum_{k=1}^{m} \Delta r^m_{ik},
\]

### 2.3 ACBS algorithm

The ACBS algorithm for forming buyer group with bundles of items can be described by the following procedure:

**Procedure ACBS()**

Initialization all pheromone values to a small numerical constant \( c > 0 \)
- Initialization of the ACBF
- \( T = \{ 0 = \) dotted line, 1 = solid line\};
  
  while not (isFinish(Iteration )){
    for Ant = 1 to MaxAnt {
      ManageAntsActivity();
      EvaporatePheromone();
      Calculate the Utility based on (8) and (9) and save the best solution found so far.
      UpdatePheromone();
    }
  }

**ManageAntsActivity()**

While not (isAntFinish(tour)){
  Choose a buyer \( a_i \) to be visited with probability \( p^m_{ai} \) in (4), (5), and (6).
  If selected path is a dotted line (\( T=0 \)), then \( \{ a_i \} \) separate with \( \{ a_j \} \).
  If selected path is a solid line (\( T=1 \)), then \( \{ a_i \} \) union with \( \{ a_j \} \).
EvaporatePheromone()

Old pheromone should not have too strong an influence on the future. The evaporation rate value is $\rho$ which is initialized to be small, $\rho \in (0,1]$.

UpdatePheromone()

Update the all the path according to (7).

3 Experimental results

In the experiment, we empirical use the utility function as shown in (8) and (9).

$$U_{c_r} = \sum_{a_{m} \in c_r} (K_1 \times \sum_{i=1}^{k} X_{r_i}^{a_m}) + K_2 \times \max \left\{ \sum_{a_{m} \in c_r} (L_1^{a_m} \times \sum_{r=1}^{k} X_{r_i}^{a_m}), \ldots, \sum_{a_{m} \in c_r} (L_{n}^{a_m} \times \sum_{r=1}^{k} X_{r_i}^{a_m}) \right\},$$

(8)

where $K_1$ and $K_2$ are the constant, and $X_{r_i}^{a_m}$ is the request $X_i$ of an buyer $a_m$. So,

$$n(C) = \sum_{c_r} U_{c_r}.$$  

(9)

In our algorithm, we assume that all buyers are requested to participate in constructing the coalition structure. To test the ACBC algorithm, a simulation was developed using Java programming language. The simulation runs on a Pentium(R) D CPU 2.80 GHz, 2 GB of RAM, IBM PC. ACBC parameters include $\alpha=0.5$, $\beta=1$, and the number of artificial ants = 1000. Several experiments have been conducted using a different set of buyers by random with 5, 10, 20, and 30 buyers respectively while the price list for input is in Table 2. The summary results of our experiments are presented in Table 3. In most cases, the average result (ten runs) of all tests derived by the ACBC algorithm is better than the result of GroupPackageString scheme. When the number of buyers is small, both ACBC algorithm and GroupPackageString scheme give similar results. However, when the number of buyers is bigger, ACBC algorithm yields about 14.2% better than the GroupPackageString. This is because the ACBC algorithm partitions the whole buyers into the best smaller subgroups in order to find the better utility.

<table>
<thead>
<tr>
<th>Number of buyers</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupPackageString</td>
<td>552.0</td>
<td>576.0</td>
<td>818.0</td>
<td>1456.0</td>
</tr>
<tr>
<td>ACBC</td>
<td>546.0</td>
<td>566.0</td>
<td>702.0</td>
<td>1316.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Average of Utility (10 runs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of buyers</td>
<td>5</td>
</tr>
<tr>
<td>GroupPackageString</td>
<td>552.0</td>
</tr>
<tr>
<td>ACBC</td>
<td>546.0</td>
</tr>
</tbody>
</table>

117

4 Conclusions and Future work

In this paper, the proposed method called ACBC is presented to form a buyer coalition with bundle of items by using the ant colony optimization technique. The algorithm is based on an imitation of the foraging behavior of real ants. The ants construct the trail by depositing pheromone after moving through a path and updating pheromone value associated with good or promising solutions thought the lines of the path. The experimental results show that in most cases the average of total utility of any coalitions formed by ACBC is better than the average of total utility derived by GroupPackageString scheme. In future, we plan to adapt the proposed algorithm to other real-complex world problems to see how well to apply.
References: