International Journal of Innovative Computing, Information and Control
Volume **14**, Number **5**, October **2018**
ICIC International ©2018 ISSN 1349-4198
pp. 1653–1679

A PRICE-BASED MECHANISM FOR ONLINE BUYER COALITION BY GENETIC ALGORITHMS

Anon Sukstrienwong
School of Information Technology and Innovation
Bangkok University
9/1 Moo. 5 Phaholyotin Road, Klong 1, Klong Luang Pathumthani 12120, Thailand
anon.su@bu.ac.th

Received January 2018; revised May 2018

Abstract. *E-commerce companies have been implementing several sales strategies to increase orders from online buyers. One of the most profitable strategies used to entice buyers to purchase more items is a volume discount on particular goods. Generally, an individual online buyer has limited bargaining power and always makes orders individually. From the perspective of buyers, they look for an efficient way to receive a lower price for their products without buying a large volume. Moreover, buyers are often heterogeneous in terms of preferences and willingness-to-pay. In such a situation, we propose a new approach for forming buying groups while taking consideration of buyers' heterogeneous preferences. The approach, which is based on genetic algorithms with roulette-wheel selection, searches for an optimized group of buyers by aggregating a number of buyer-selected items to obtain the highest utility received from the sellers. The paper compares the performance of the algorithm using roulette-wheel selection with generational replacement. Additionally, the web-based application of the proposed approach is developed in order to illustrate how the proposed algorithm works in the real world. The experimental results of our empirical case study show that the algorithm optimally searches for the best solution.*
Keywords: Bundles of items, Buyer coalition, Genetic algorithms, Group formation, Roulette-wheel selection, Online shopping

1. **Introduction.** The development of information technology has facilitated the adoption of online shopping over the past number of years. More and more e-commerce companies have been using online markets as a channel to sell their products because it allows them to sell products globally. Therefore, online markets are favored by sellers as they try to widen their customer base and increase their sales. When expecting to buy a product, buyers will naturally think about themselves and what they will gain. They look for the most affordable Website. In general cases, buyers can find products based on seller offers. To encourage customers to buy more items, some sellers often offer a certain percentage discount to customers who buy a large volume of products. Some sellers offer free shipping to buyers who purchase more than one item at a single time. Matsuo et al. [1] highlighted that a volume discount is an important aspect of e-commerce, which is called group buying. Occasionally, some sellers offer products in multiple categories and formats. Consequently, there exists a particular strategy for buyers in order to purchase products with satisfaction. The strategy referred to as group-buying is a way to forming groups of buyers to enlarge the total quantity of goods in a transaction. Theoretically, a buyer coalition is a group of two or more buyers who agree to aggregate their buying requests together in purchasing goods from sellers. Group-buying has increasingly become

DOI: 10.24507/ijicic.14.05.1653

the focus of several researchers, such as [2,3]. Most group-buying schemes have been proposed in the literature based on the price mechanism [4]. In the group-buying market, the price of goods that each buyer needs to pay depends on the aggregated participants in that group.

Hsieh and Lin [5] studied the benefits of buyer groups based on combinatorial reverse auctions in which the roles of buyer and seller are reversed. In a reverse auction, sellers bid for the prices at which they are willing to sell their goods and services as the sellers underbid each other. As documented by Dang and Jennings [6] and Chen et al. [7], a number of algorithms have been developed to achieve reverse auctions. Chen et al. also developed an approach to determine optimal biding of fixed-pricing strategies. Similarly, In the work of [8], authors stated that the buyer cooperative strategy originated from Internet bidding. The strategy referred to as 'group-buying price' is claimed to be an efficient approach to aggregate buying orders. Additionally, it allows all buyers to purchase at low cost if the order quantity reaches a specific point made by the seller.

Generally, a buyer may be in the buying-group if they pay less money for the items demanded. Otherwise, the buyer will refuse to be involved in the group. In addition, group formation is typically hard to solve since a group of buyers can be characterized by a diversity of preferences. It is likely to be difficult to search for a solution that is able to satisfy all buyers in the group. These issues illustrate how buyers should be coordinated to purchase goods in groups.

Another critical issue is how to form a buying-group of heterogeneous buyers as they may expect to buy different goods in a market. In practice, all buyers will be automatically placed into different groups. However, due to budget constraints, there will be some buyers that refuse to join the group if they pay for the products higher than they are expected. Additionally, some buyers have posted the reservation price, which is the maximum price that they are willing to pay for goods. In some cases, lowering the prices for some buyers would make it difficult to form the buying group. Consequently, it would be beneficial for all other participants to remove some of these buyers from the group. For instance, let us suppose there exist three potential buyers who want to buy an electric tower fan. The first buyer and the second buyer have made a reservation price for \$120, and the third buyer is willing to pay at most \$70. The reservation price is the maximum price a buyer can pay for a product [9]. Suppose the unit price listed by the seller is \$145 for one electric tower fan, \$115 for two, and \$105 for three. If each buyer purchases alone, no buyer can buy it because each one must pay at least \$145. If three buyers form the group by aggregating their demand together, they will pay at least \$315 to buy three electric tower fans. The total budget, however, of all buyers is \$310. The group of three buyers cannot buy it either. However, if the buyers identify that the third buyer made a reservation price as low as \$70, it is impossible to form the buying group. Therefore, the third buyer cannot join the group and indeed it is better to not allow them into the group. Then, a group composed of the first and second buyers, who are willing to pay at most \$120, can combine their demands and purchase two electric tower fans at a total of $115 * 2 = \$330$. As a result, they do not pay as high as \$145 for an electric tower fan, but each buyer will get the electric tower fan $120 - 115 = \$5$ lower than that they are expected. Hence, the group of two buyers will save $(145 - 115) * 2 = \$60$ for their purchase.

If there exist a large number of prospective buyers, there are various ways which form the group of buyers. The process of building an optimized group can be time-consuming when a large number of individual buyers place multiple orders. Despite the ubiquity of such group buying sites, buyers do not have more opportunities to post their evaluation in group buying sites since there is a lack of efficient tools to support the decisions of multiple buyers when multiple products are mostly combined in a bundle of items for a discounted price or the total sum. This is in practice true for prospective buyers when the purchasing time or deadline is short. Moreover, the discount policies of different sellers give different search results for the group of buyers. This makes an exhaustive search of a large space for an optimal solution. Accordingly, the search for an optimized group composition would be more difficult and time-consuming to satisfy all buyers in economic terms. Moreover, the solution space is exponentially increased if the number of buyers is big. Therefore, we use genetic algorithm (GA) to solve our problem because GA approach works well in large search space problems and offers significant benefits over the search of optimization techniques.

Therefore, the proposed algorithm called the forming buyer groups by genetic algorithms (FBGGA) aggregates buyers' requirements and then optimally forms the groups of buyers using a genetic algorithm to enable all buyers to receive the product at a lower price. The GA used in this paper is characterized by 5 parameters: population size, number of generations, crossover operator, mutation operator, and roulette-wheel selection. There are two criteria involved in the proposed algorithm: 1) group-buying with a fixed time period to completion as defined by the sellers and 2) group-buying with a discount list that is achieved when enough volumes of items are met. The main purpose of this paper is to optimally build a buyer coalition in order to maximize their benefits, where multiple items are sold together at a set price. The web-based application of our proposed algorithm is developed in order to support buyers participating in group buying on the online market.

This paper is divided into eight parts including this section. The rest of the paper is organized as follows. Section 2 presents a definition of the buyer coalition scheme that is found in the literature. The brief concepts of genetic algorithms and chromosome representations are presented in Section 3. In Section 4, we present a detailed mathematical formulation for a proposed algorithm used to form a buyer coalition under a discount policy where two or more products are sold together at a set price. Assumptions and definitions used in this work are illustrated in this section. It also demonstrates the steps of the formation of buyer coalitions by the FBGGA algorithm. In order to show the effectiveness of the proposed algorithm, we show an empirical example in Section 5. We then present the development of a web-based application and user interfaces of buyer coalitions in Section 6. Furthermore, the time complexity of the FBGGA is discussed in Section 7. Finally, the conclusions and future work are presented in the last section.

2. **Related Works.** Presently, in the online markets most authors focus on buyer coalitions formed to gain discounts. According to [10-13], coalition formation is defined as an alliance among individuals and widely studied as a characteristic function in game theory. The main idea of coalition formation is to deal with the analysis of several groups of agents, called coalitions, that join together to determine their actions. In the e-market, buyer coalition formation involves a group of buyers who unite in order to purchase products at a lower cost because together they have more bargaining power than an individual. There exist several types of algorithms for buyer coalitions to meet buyer requirements. Firstly, Yamamoto and Sycara [2] proposed an efficient algorithm of coalition formation called the GroupBuyAuction. The algorithm enables a large number of buyers to form coalitions based on item categories. Chen et al. [12] analyzed the seller's pricing strategy with the group-buying auction (GBA) when a number of people agree to buy a product or service. The researchers also studied the group-buying auction for the optimal fixed pricing mechanism. In GBA, buyers receive a coupon or voucher to claim their discount at the retailer. There are quite a few group-buying auction companies that have operated with

innovative and different business models. Some involve buying pools and buyer-supplier price negotiation.

Some researchers such as Sandholm et al. [15] studied the coalition structures that maximize the sum of the benefit of the coalitions. They presented a scheme that searches coalition structures with the worst case guarantees in the minimal search time. However, Van Horn et al. [16] reported that many group-buying auction websites have failed and subsequently reoriented their selling mechanism. Supported by Ni et al. [17], the authors documented that many group-buying websites have collapsed in China. Li and Sycara [3] then addressed a coalition of buyers where the buyers want to bundle items together to fulfill each other's needs, in which buyer coalition formation in an electronic marketplace is called "combinatorial coalition formation". They consider a market in which combinatorial auctions exist. Moreover, Boongasame et al. [18] addressed the mechanism of forming buyer coalitions with bundles of items called the GroupBuyPackage scheme in order to maximize the total discount of the coalition. The advantage to the buyers in the group is related to the difference between the sum of the reservation price of each buyer and the minimum cost needed to satisfy all members. However, the group-buying auction is an approach that is operated by the sellers.

An interesting issue investigated by Ito et al. [19] is to allow more than one seller to cooperate within the group formation when a buyer coalition is required. Another topic was investigated by Hyodo et al. [20]. They claimed that there exist several group-buying sites selling the same items at the same time. However, buyers have no means to optimally distribute them among these group-buying sites. If buyers can optimally allocate to several group-buying sites, then all buyers can purchase items at a lower price. A framework to price multi-product bundles is proposed by Yasar [21]. They employed multivariate normal distribution on several aspects involved bundling. They aimed to maximize the profit of a seller while matching supply and demand.

Finally, He and Ioerger [22] considered the problem of group buying in combination with a bundle search based on an empirical study. The authors simulated a heuristic algorithm to search for coalitions in order to minimize the cost to each buyer.

3. **Concept of Genetic Algorithms.** Genetic algorithms (GAs) are an adaptive heuristic that mimics the process of natural evolution. Typically, GAs can be considered as being part of the huge category of evolutionary algorithms (EA). Many researchers who observe the evolution of biological complexity are astonished that natural evolution has produced some remarkably complex organisms. Over millions of years the process of genetic selection has produced organisms that are perfectly adapted to their environment. This is the reason why GAs are being employed in real-world problems [23]. The original GA was introduced by John Holland at the University of Michigan in the early 1970s [24,25]. The key idea surrounding GAs is to adapt the power of evolution to solve optimization problems. Thereafter, a series of studies [26-29] were applied to a broad range of subjects, such as computer science and engineering, to develop solutions to complex combinatorial optimization problems.

The basic idea of genetic algorithms starts off with a population of "chromosomes", which encodes a candidate solution to a problem. Traditionally, a standard representation of solutions is represented in binary form as strings of 0s and 1s. This is because bit strings can be viewed as a chromosome-type structure. The initial population of chromosomes is chosen randomly and evolves over a number of generations. Those that create a high quality solution will be selected in the next generation. The evaluation function, referred to as the "fitness function", must be designed to measure the quality of each chromosome with respect to the problem under consideration. In each generation, the fitness value of every chromosome in the population is evaluated; therefore, it should be sufficiently fast to compute.

The population size is the number of chromosomes involved in every generation. The diversity of the population should be maintained to yield a better solution; otherwise, this might lead directly to premature convergence [30], which is an undesirable condition in GAs. There are three common operators involved in the evolution of chromosomes.

Selecting operator: this involves selecting parents to mate and recombine to create off-spring for the next generation. This allows them to pass on good genes to the next generation. It is a vital force evolution in any algorithm. Moreover, it should be treated carefully to avoid leading to a loss of diversity among the population within a few generations. With the power of selecting the operator, the population tends to improve the quality of the fitness value on average with each generation. There are many techniques regarding how to select the chromosomes, such as generational replacement and roulette wheel selection. In generational replacement, the entire population of genomes is replaced at each generation. This technique picks parents to produce offspring from among the current population. After that, it inserts these individuals into a new generation. When the number of individuals that has been created is as big as a fixed number of the current population, the new population replaces the old one. This cycle is again repeated until it meets the maximum generation. Roulette-wheel selection is a frequently used technique in genetic algorithm and is also used in the generation of complex problems. The probability of roulette-wheel selection is proportional to the fitness of an individual. In roulette wheel selection, the circular wheel is divided into n pieces. Every chromosome has its place on the wheel according to its fitness value. The chromosomes with the best fitness have bigger shares of the wheel. This means that the better chromosomes have a greater chance of being selected, hence the comparison to a roulette wheel in a casino. When the wheel is rotated, a fixed point of the wheel chooses the chromosome.

Crossover operator: this is the process of how parents recombine to create off-spring. Using a selection operator alone will tend to fill the population with copies of the most promising chromosome. In general, there are three kinds of crossover operators. In a single-point crossover, two parent chromosomes are selected randomly, with one point chosen as a crossover point. The information beyond this point of the chromosomes is swapped. The result of this operator creates children that are different from their parents. Two-point crossover is similar to single-point crossover except that two crossover points are randomly selected. The contents between the two points are swapped between the parents, resulting in the generation of two offspring. Uniform crossover is similar to both single-point and two-point crossover, but the parents are not divided into segments. The bits of each parent chromosome are treated separately. Hence, bit strings are randomly selected, either from the first or second parent. The crossover occurs only with a crossover rate referred to as pc (crossover probability), which is the percentage of the time that the crossover process occurs when two parent chromosomes are chosen to recombine. When an individual parent is not subjected to crossover, two parents remain unmodified and are copied directly into the next generation. Typically, the value of pc is small in the range of 0.5$\sim$1.0 [31].

Mutation operator: this operator is used to modify some parts of chromosome in selected parents. Generally, the mutation operator helps to maintain diversity among the population. Therefore, it helps the GAs escape from local optimal traps. The mutation process occurs only with a mutation rate pm (mutation probability) which is the percentage of time that the mutation process occurs when some elements of a chosen parent are selected to mutate. The major parameter in the mutation operator is the mutation rate, which controls the speed of GAs in exploring a new area. Typically, small pm values in the range of 0.001∼0.05 are adopted in GAs [31].

In the most common type of genetic algorithms, the evolution of chromosomes works repeatedly until no better solution has been found or a common termination condition (maximum number of generations) has been reached, depending on the needs of the programmers. If the termination condition is satisfied, the best solution in the current population is the answer of GAs.

4. Mathematical Formulation for Buyer Coalition.

In this section, we illustrate a mathematical formulation for buyer coalition. We then detail our proposed approach for forming buyer groups by using genetic algorithms where several products are sold together at a set price.

4.1. Preliminaries and problem formulation.

In this section, we present our model for buyer formation with bundles of items using genetic algorithms with roulette-wheel selection and give some definitions for the presented model. Table 1 summarizes all the notations used in this paper.

Table 1. Summarized notations

Symbols	Meaning
B	A set of buyers, where $\|B\| = n$.
b_i	Buyer i, where $b_i \in B$ and $1 \le i \le n$.
S	A set of sellers where $\|S\| = t$.
s_i	Seller i where $s_i \in S$ and $1 \le i \le t$.
$Product_{ij}$	Product j sold by s_i and $1 \le j \le m$.
m	Number of products.
$price_{ij}$	Price of $Product_{ij}$.
d_{ij}	Demand of b_i for product j.
r_{ij}	Reservation price of b_i for product j.
$reservation(b_i)$	Total reservation of b_i.
$Reservation(C_i)$	Total reservation of C_i.
$Reservation(B)$	Total reservation of all buyers.
$paid(b_i)$	Total cost to buy all required items for b_i.
$Paid(C_i)$	Total cost to buy all required items for C_i.
k	Number of all subgroups.
C_j	Sub-group of buyers, where $1 \le j \le k$ and $C_1 \cap C_2 \cap \cdots \cap C_k = \emptyset$.
$Benefit(B)$	Benefit of all buyers.
$Benefit'(B)$	The best benefit of all buyers.
$Benefit(C_j)$	Benefit of C_j, where $1 \le j \le k$.
$Package_{ij}$	Package j sold by s_i.
$PackagePrice_{ij}$	Price of $Package_{ij}$.
G	Set of products, where $\|G\| = m$.
N_{ij}	Number of $Package_{ij}$ purchased by the group.

Let $B = \{b_1, b_2, \ldots, b_n\}$ be the set of buyers participating in the e-market, where n is the number of buyers. We can investigate the size of the search space in forming buyer groups. There are k different subgroups to be formed, and the sizes of all subgroups are $|C_1| = q_1$, $|C_2| = q_2$, $|C_3| = q_3$, $\ldots$, $|C_k| = q_k$. Then, we start with the following formula, $\begin{pmatrix} n! \\ q_1, q_2, q_3, \ldots, q_k \end{pmatrix} = \frac{n!}{q_1!q_2!q_3!\ldots q_k!}$. If the sizes of all subgroups are the same where $q = q_1 = q_2 = q_3 = \cdots = q_k$, then the equation can be represented as $\begin{pmatrix} n! \\ q_1, q_2, q_3, \ldots, q_k \end{pmatrix} = \frac{n!}{(q!)^k}$. Since all subgroups are the same, then $\frac{n}{k} = q$. The equation becomes $\begin{pmatrix} n! \\ q_1, q_2, q_3, \ldots, q_k \end{pmatrix} = \frac{n!}{\left(\left(\frac{n}{k}\right)!\right)^k}$. However, any subgroup i named q_i, where $1 \leq i \leq k$, can be labeled to be the other subgroup making the ways of forming groups become $\frac{n!}{k!\left(\left(\frac{n}{k}\right)!\right)^k}$. For instance, 15 buyers are divided into 5 smaller subgroups. Each subgroup contains 3 buyers. The different ways of forming buyer groups is $\frac{15!}{5!\left(\left(\frac{15}{5}\right)!\right)^5} = 140400$. Moreover, buyer i-th has its own preference to purchase some items. Thus, searching for the optimal solution where buyers pay for products at low prices becomes more complex.

Let $S = \{s_1, s_2, \ldots, s_t\}$ be the set of sellers and t be the number of sellers. Different sellers sell the same product at different prices, which can be expressed in the following matrix:

$$Price = \begin{pmatrix} price_{11} & price_{12} & \ldots & price_{1m} \\ price_{21} & price_{22} & \ldots & price_{2m} \\ \vdots & \vdots & & \vdots \\ price_{t1} & price_{t2} & \ldots & price_{tm} \end{pmatrix},$$

where p_{ij} is the price of product j made by a seller s_i and m is number of different products.

Then, multiple buyers' requirements with their corresponding reservation price are illustrated respectively as follows:

$$\mathrm{D} = \begin{pmatrix} d_{11} & d_{12} & \ldots & d_{1m} \\ d_{21} & d_{22} & \ldots & d_{2m} \\ \vdots & \vdots & & \vdots \\ d_{n1} & d_{n2} & \ldots & d_{nm} \end{pmatrix}, \quad \mathrm{R} = \begin{pmatrix} r_{11} & r_{12} & \ldots & r_{1m} \\ r_{21} & r_{22} & \ldots & r_{2m} \\ \vdots & \vdots & & \vdots \\ r_{n1} & r_{n2} & \ldots & r_{nm} \end{pmatrix}.$$

In order to find the benefit to the buying group, we simply calculate the total reservation price of each buyer. If buyer b_i needs to purchase some items, the total reservation of b_i is calculated as shown in (1).

$$reservation(b_i) = \sum_{j=1}^{m} (d_{ij} * r_{ij}). \tag{1}$$

The total amount of money needed to buy all required items for b_i is calculated by the following equation:

$$paid(b_i) = \sum_{j=1}^{m} \min_{1<r<t}(price_{rj}), \tag{2}$$

where t is the number of sellers.

Then, the benefit of all buyers purchasing goods can be presented as follows:

$$Benefit(B) = \sum_{i=1}^{n} (reservation(b_i) - paid(b_i)), \tag{3}$$

where n is the number of buyers.

We extend these equations for a sub-group of buyers. Let C_j be a non-empty set of buyers, where $C_j \subseteq B$, and $1 \leq j \leq k$. Then, $\bigcap_{j=1}^{k} C_j = \emptyset$, $\bigcup_{j=1}^{k} C_j = B$, and $1 \leq k \leq n$. Then, the total reservation price of C_j can be represented as below:

$$Benefit(C_j) = \sum_{i=1}^{n_j} (Reservation(C_i) - Paid(C_i)), \tag{4}$$

where $n_j = |C_j|$ and $1 \leq j \leq k$.

Finally, the benefit to all buyers can be shown as (5), in which its result is equal to (3).

$$Benefit(B) = \sum_{j=1}^{k} \sum_{i=1}^{n_j} (Reservation(C_i) - Paid(C_i)), \tag{5}$$

where k is the number of groups.

However, the benefit of C_j can be true only if the reservation of C_j, namely *Reservation*(C_j), is greater than the total amount of money paid to buy all requirements, referred to as $Paid(C_j)$. In this paper, we allow buyers to set a reservation price and sellers provide a list price of the items and bundles of items.

In order to investigate the group buying, we allow a group of buyers to combine their requirements and search for the cheapest price to achieve the greatest benefit. In this paper, a bundle of products is referred to as a "package". Hence, the possible packages sold by different sellers are represented as follows:

$$Package = \begin{pmatrix} Package_{11} & Package_{12} & \dots & Package_{1q} \\ Package_{21} & Package_{22} & \dots & Package_{2q} \\ \vdots & \vdots & & \vdots \\ Package_{t1} & Package_{t2} & \dots & Package_{tq} \end{pmatrix},$$

where q is the maximum number of packages provided.

There are m different products to sell online, $G = \{g_1, g_2, \dots, g_m\}$. Each package is comprised of multiple items; therefore, the representation of a pack i of seller j becomes $Package_{ij} = \left(g_{ij}^1, g_{ij}^2, \dots, g_{ij}^m\right)$ and the price of this package is $PackagePrice_{ij}$. Let N_{ij} be the number of $Package_{ij}$ purchased by the group. Then, the money paid for $Package_{ij}$ is equal to $N_{ij} * PackagePrice_{ij}$. However, there are several possible ways to serve the buyers' needs. Let d be the number of possible ways to buy. From (2), the greatest benefit to the group can be calculated as follows:

$$Benefit'(B) = \min_{1 \leq l \leq d} \left(\sum_{i=1}^{t} \sum_{j=1}^{q} \left(N_{ij}^l * PackagePrice_{ij}\right) \right) - \sum_{i=1}^{n} (reservation_i), \tag{6}$$

subject to the following equalities.

$$\sum_{i=1}^{t} \sum_{j=1}^{q} \left(g_{ij}^1 * N_{ij}\right) - \sum_{i=1}^{n} d_{i1} = 0,$$

$$\sum_{i=1}^{t} \sum_{j=1}^{q} \left(g_{ij}^2 * N_{ij}\right) - \sum_{i=1}^{n} d_{i2} = 0,$$

$$\dots$$

$$\sum_{i=1}^{t}\sum_{j=1}^{q}\left(g_{ij}^{m} * N_{ij}\right) - \sum_{i=1}^{n} d_{im} = 0.$$

To maximize the total utility for the group of buyers, the value of $Benefit'(B)$ in (6) must be greater than the value of $Benefit(B)$ in (5).

4.2. **Formation of buyer groups using genetic algorithms.** In this paper, the forming buyer groups by genetic algorithms (FBGGA) approach works on the assumption that buyers do not share information while they are in the group except for the group leader who initiates the formation of the group itself. Buyer reservation prices will be kept secret among buyers after they express their preferences. More importantly, the objective of our algorithm is to obtain a lower price for their products received from the sellers. Since our approach allows several sellers to offer multiple products online, there are various ways for buyers to enjoy lower costs when buying products. In addition, there are many possible ways for sellers to combine two or more products for sale. In such a situation, the optimal group formation of buyers could be very complex.

4.2.1. *Problem encapsulation.* Using genetic algorithms involves defining a chromosome that describes the problem itself. Then, we encode the problem as a chromosome. We assume that there are p packages mixed by at most r different products and associated with $Package$ and $Price$, as illustrated above. The group of buyers aims to buy packages at low cost in order to gain the optimized benefit of the group; therefore, the first section of a chromosome comprises an array of p integers. Each v_i, where $0 \leq i \leq p$, is the number of package i^{th} that the group needs to meet the buyer's request. The second section of the chromosome comprises an array of r integers for all single items. The value of u_j represents the number of product j^{th}, where $0 \leq j \leq r$. In order to calculate the total money spent on all packages and single items, let us suppose the price for v_i is $PackagePrice_i$ and the price of u_j is $ItemPrice_j$. The chromosomes for our algorithm will be sequences of integers with length $p + r$, as demonstrated in Figure 1.

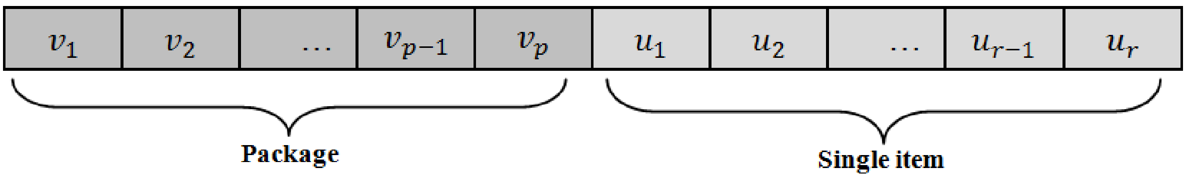


FIGURE 1. Chromosome structure of buyer coalition, where p is the number of packages and r is the number of products

Suppose there are six products ($r = 6$) requested by the group of buyers and the sellers have made a total of five bundles of items ($p = 5$). As the result, the chromosome length of our design will be $p + r = 5 + 6 = 11$. In addition, each integer in the array represents the number of packages or single item required by all buyers of the group. If buying nine packages of v_1 and two packages of u_4 can fully support all of the buyers' needs, the chromosome can be encoded as represented in Figure 2(a) and referred to as $chromosome_x$. We can see that unwanted packages and items will be set as zero. There are various ways to obtain all of the products for the entire group of buyers. If buying three packages of v_2 and one item of v_1 to v_6, $chromosome_y$ can be encoded as shown in Figure 2(b).

In this paper, we consider a situation where all prospective buyers post their preferences. Then, the information about buyers' preferences and decisions are kept for the group formation which will be used for evaluating the quality of chromosomes.

9	0	0	0	0	0	0	0	2	0	0

(a) $chromosome_x$

0	3	0	0	0	1	1	1	1	1	1

(b) $chromosome_y$

Figure 2. Examples of chromosomes for a buyer coalition with five packages and six products

4.2.2. *Fitness function.* Generally, GAs use the fitness function to map the aim of the problem and search the space of possible chromosomes in an attempt to find suitable solutions. As assumed earlier, all buyers must post the reservation price, which is the highest price that they are willing to pay for the product. The reservation price of the group i, called g_i, is $Reservation(chromosome_i)$; therefore, the total amount paid for all constructed groups is illustrated below.

$$Reservation = \sum_{i=1}^{k}(Reservation_i), \tag{7}$$

where k is the number of subgroups of buyers.

Moreover, the reservation of all subgroups formed can be calculated by (1). That is, the total reservation of all buyers is presented as follows.

$$Reservation = \sum_{i=1}^{k}\left(\sum_{j=1}^{m} d_{ij} * r_{ij}\right), \tag{8}$$

where m is the number of products.

Consequently, the fitness evaluation of $chromosome_k$ with length $p+r$ can be calculated as follows.

$$\begin{aligned} fitness(chromosome_k) = {} & Reservation - (PackageCost(chromosome_k) \\ & + ItemCost(chromosome_k)), \end{aligned} \tag{9}$$

where

$$PackageCost(chromosome_k) = \sum_{i=1}^{p}(v_i * PackagePrice_i) \tag{10}$$

and

$$ItemCost(chromosome_k) = \sum_{j=1}^{r}(u_j * ItemPrice_j). \tag{11}$$

As stated earlier, the main objective of the proposed algorithm is to obtain the highest utility received from the sellers. Therefore, the better chromosome that yields the highest fitness value is considered as a promising solution.

4.2.3. *FBGGA operators.* After we have designed the chromosome structure and fitness function of the buyer coalition, the operators must be designed to guide the FBGGA towards the optimized solution. Figure 3 illustrates a flowchart of the FBGGA. There are a few methods in designing the stopping criteria for GAs. Additionally, proof of convergence of an algorithm to an optimal solution is the most popular method, as it assures the optimal solution in infinite iterations [32]. Additionally, convergence can be measured by

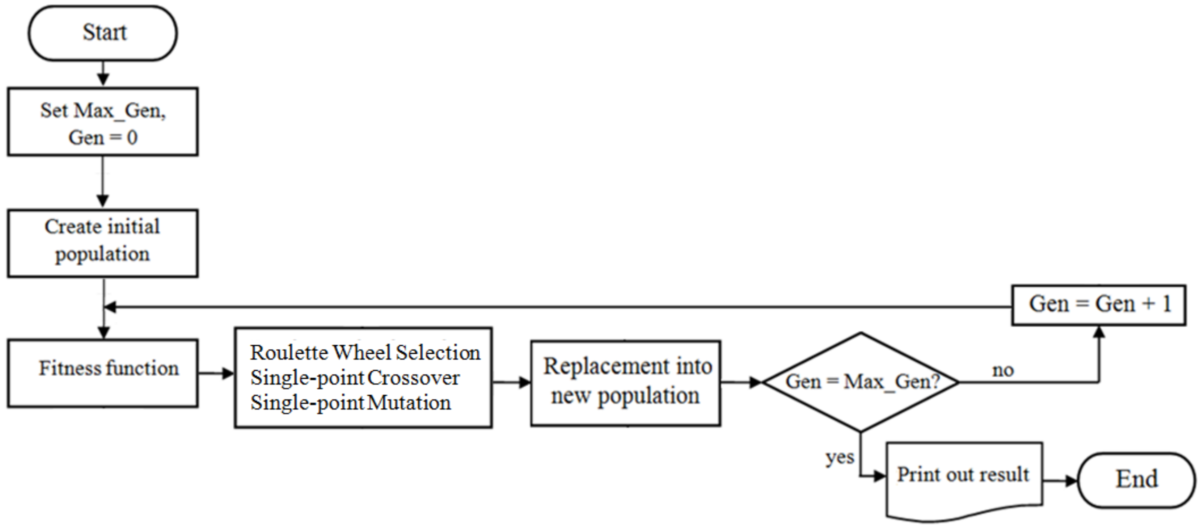


FIGURE 3. Flowchart of the FBGGA

observing the change of the optimal value of each generation. In general GAs, nevertheless, in this paper the criterion is simple. The proposed algorithm proceeds until the best solution is found within a predefined value of maximum number of generations (Max_Gen). This predefined number can be found during the experiment, which is explained in Section 7.

The three main operators are 1) the roulette wheel selection operator, 2) the single-point crossover operator and 3) the single-point mutation operator. The details of each operator are described in detail below.

(1) Roulette wheel selection operator: Parent chromosomes are selected according to their fitness value. Chromosomes with higher fitness values have a greater chance of being selected than weaker ones. The probability that $chromosome_k$ is selected, $P(chromosome_k)$, is computed as follows:

$$P(chromosome_k) = \frac{fitness(chromosome_k)}{\sum_{j=1}^{n} fitness(chromosome_j)}, \tag{12}$$

where n is the total number of chromosomes and $1 \leq k \leq n$.

Additionally, the principle of roulette selection is a linear search through a wheel. The pseudocode for the roulette wheel selection operator can be presented as below.

Algorithm: RouletteWheelSelection()
$SumFitness := \sum_{j=1}^{n} fitness(chromosome_j)$
r: random number; where $0 \leq r \leq 1$;
sum := 0;
for each individual i
{
    $P(chromosome_i) = fitness(chromosome_i)/SumFitness$;
    $sum := sum + P(chromosome_i)$;
    If $r < sum$
        return i;
}

The variable r from the pseudocode presented above is a selection point on a roulette wheel. Additionally, all of the population's chromosomes will be placed on a roulette wheel.

For instance, let us assume that five chromosomes of the population have the fitness value represented below.

$$fitness(chromosome_1) = 5.0$$
$$fitness(chromosome_2) = 3.0$$
$$fitness(chromosome_3) = 6.0$$
$$fitness(chromosome_4) = 14.0$$
$$fitness(chromosome_5) = 4.0$$

Then, $SumFitness := 5 + 3 + 6 + 12 + 6 = 32$. $chromosome_3$ has the best fitness value; therefore, it is given a bigger portion of the wheel. The weakest chromosome is $chromosome_2$; thus, it is given the smallest portion of the wheel. The probability of each chromosome can be calculated as follows:

$$P(chromosome_1) = 5/32 = 15.63\%$$
$$P(chromosome_2) = 3/32 = 9.38\%$$
$$P(chromosome_3) = 5/32 = 18.75\%$$
$$P(chromosome_4) = 5/32 = 43.75\%$$
$$P(chromosome_5) = 4/32 = 12.50\%$$

Therefore, the roulette wheel can be illustrated as in Figure 4.

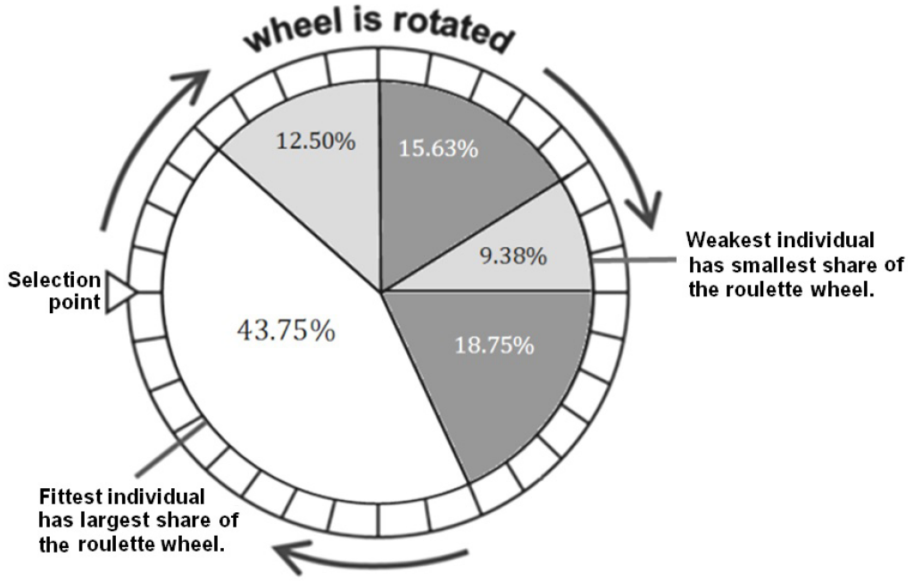


Figure 4. Roulette wheel selection for FBGGA shows that the survival probability of each individual is proportional to its relative fitness.

(2) Single-point crossover operator: This operator selects two parents in a completely random fashion, referred to as $chromosome_x$ and $chromosome_y$. It then randomly chooses a crossover point and exchanges everything before the crossover point between $chromosome_x$ and $chromosome_y$ producing children. Figure 5 illustrates a single-point crossover for our algorithm. The dotted line indicates the crossover points. Thus, this results in an exchange between two parents producing two new offspring.

(3) Single-point mutation operator: The purpose of this operator is to maintain population diversity. It is carried out by a single parent and searches through the search space to prevent premature convergence. Like the crossover operators, the parent is randomly chosen. Then, a mutation point is selected randomly. Figure 6 presents an example of a single-point mutation for FBGGA.

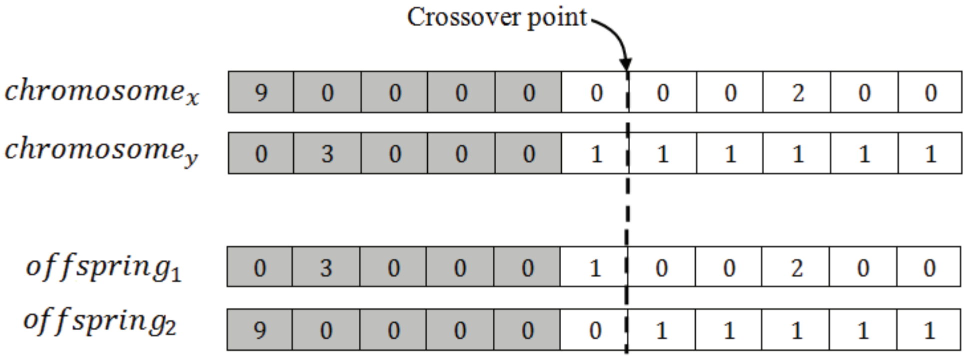


Figure 5. An example of the single-point crossover operator for the FBGBA

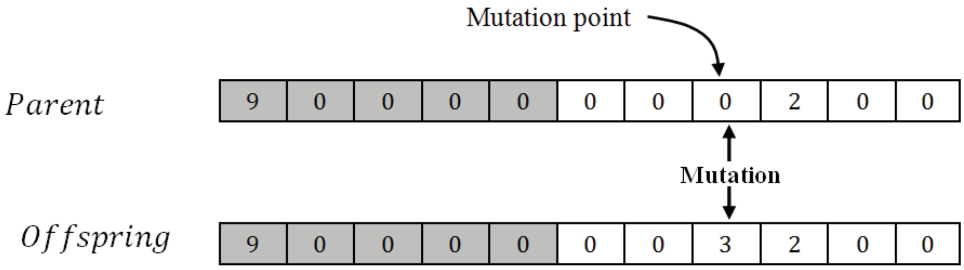


Figure 6. An example of the single-point mutation operator for the FBGGA

(4) Steps of forming buyer coalition with the FBGGA algorithm. The application allows anyone to start a new group in purchasing goods from online sellers. It is important to emphasize that the objective of the application is to combine all requests of online buyers who want to get the products at low cost and do not purchase a big volume of discount products. Moreover, the sellers are using selling strategies to get more orders from the buyers, such as bungling pricing or bungling items of products. The application has been operating a member-only. Hence, each buyer who wants to take part in the group formation must complete an application for membership. All account details, such as username, password, shipping address and email address, will be stored in the database of the system.

To this paper, the steps for forming buyer coalition are presented in Figure 7. More importantly, there are three criteria involving the proposed algorithm:

1) group-buying with a fixed time period for completion as defined by the leaders,

2) group-buying with a discount list that is achieved when enough volume of items is met, and

3) the benefit of buyers gained from the group formation based on the buyers' requests. In some cases, requests are poor, causing a failure of a group formation. As a result, the group of buyers cannot buy any products.

The steps of forming a buyer coalition, where the FBGGA algorithm is employed, are demonstrated below.

1) Group leader: A group leader is a buyer who initiates the new group. The group leader can be anyone, but he must be responsible for setting up the group and entering the information of the products and the sellers. Prospective buyers who have interested and entered into a certain group must enter their requests for buying products, which are listed by the group leader. Additionally, all group members will be asked to put the reservation prices associated with the requested goods.

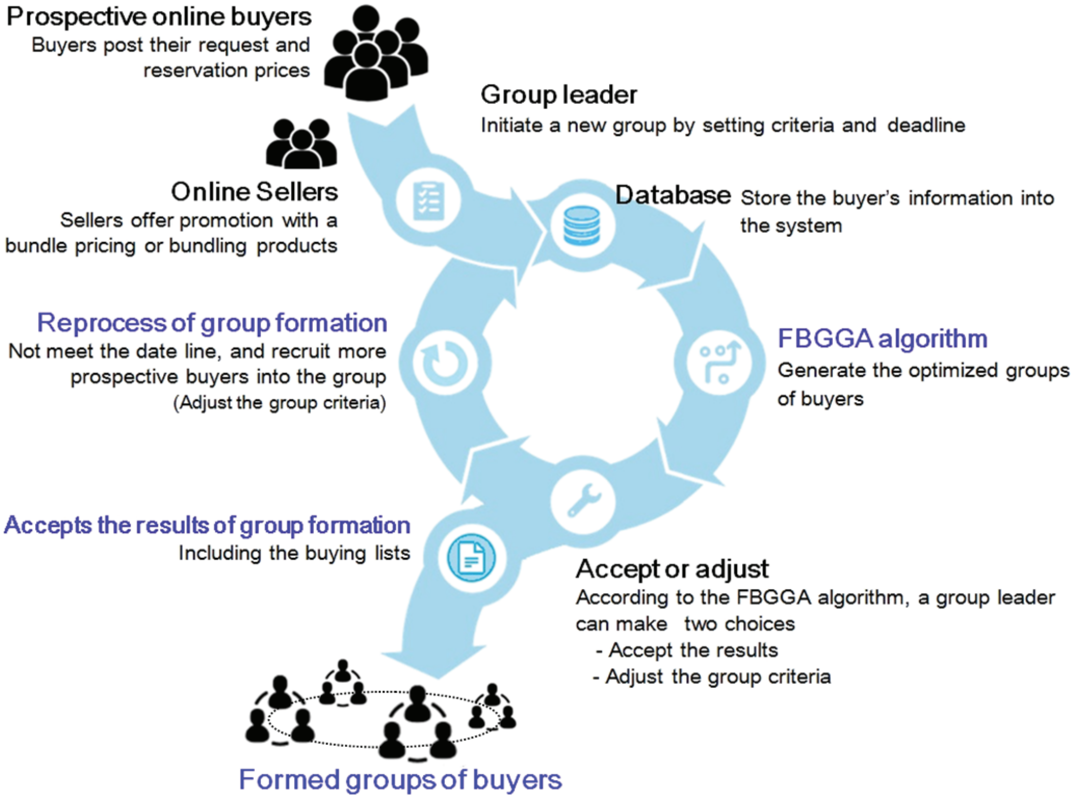


Figure 7. Steps of forming buyer coalition where the FBGGA algorithm is employed

2) Database: All buyer requests and reservation prices will be stored in the database and kept secret from others after they have expressed their preferences.

3) FBGGA algorithm: When all buyers' information is completely entered, the FBGGA algorithm can be performed. This can be done by either an automatic running of the application or the group leader.

4) Accept or adjust: This step is also called "Decision step", since the group leader can make a decision to accept the result of the FBGGA algorithm. If the group of buyers is well-established meaning that most buyers can get the products at the reservation price or lower, at this point the leader may accept the result. Otherwise, the leader can repeat the process. During this time, the buyers can edit the requests or put more requests in buying goods. Also, new online buyers can enter to participate with. The active group of buyers, which is in a process, is called "Ongoing group". The group is either closed by the group leader or met the group criteria which are called "Closed group". If the group formation is successfully formed, all participants will get the results.

5) Reprocess the group formation: If the group is still active, the group reprocesses the group formation. More prospective buyers can join in and some buyers can leave the group.

5. **FBGGA Algorithm Revisited with an Empirical Case Study.** In this section, we present a case study to illustrate how the proposed algorithm works. Assume that we have two online sellers available, s_1 and s_2, that are selling the same products. The list price of each seller's products is presented in Table 2. Suppose that each seller has combined several items to sell in the discount price as shown in Table 3.

5.1. **Chromosome structure.** We can see from Table 2 that only the first two products of s_1 (item1 and item2) are cheaper than the same products sold by that of s_2. However,

Table 2. Lists of products

Seller	Products	Price ($)
s_1	Item1	76.00*
	Item2	38.00*
	Item3	170.00
	Item4	16.00
s_2	Item1	83.00
	Item2	39.00
	Item3	143.00*
	Item4	15.00*

Table 3. Lists of packages (bundle of items)

Seller	Packages	Products				Price ($)
		Item1	Item2	Item3	Item4	
s_1	p1	1	1	–	1	100.00
	p2	1	–	–	1	85.00
s_2	p3	–	–	1	1	145.00
	p4	2	1	–	–	150.00
	p5	–	1	1	1	165.00

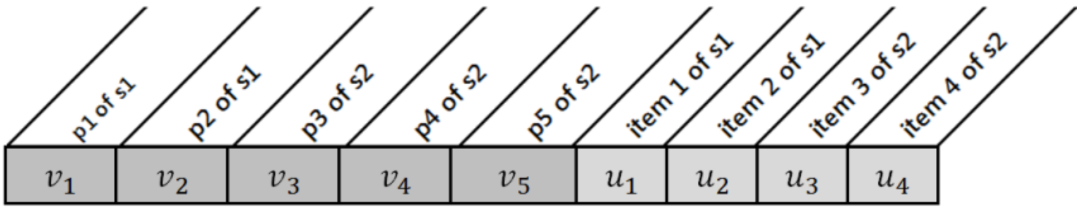


Figure 8. Chromosome structure of buyer coalition

the last two products of s_2 (item3 and item4) are sold at a lower price than s_1. Therefore, these cheaper items will be encoded into the chromosome, as indicated in Figure 8 in detail. Additionally, five packages listed in Table 3 are encoded into the first section of the chromosome. Hence, the chromosome will be a sequence of integers with length $5 + 4 = 9$, as demonstrated in Figure 8.

Suppose there are four buyers, namely b_1, b_2, b_3 and b_4, participating in the group. Assume that all buyers have heterogeneous preferences, as represented in Table 4, in which the total reservation price of all buyers is \$1,292.

Suppose there are four buyers, namely b_1, b_2, b_3 and b_4, participating in the group. Assume that all buyers have heterogeneous preferences, as represented in Table 4, in which the total reservation price of all buyers is \$1,292.

Suppose that in the FBGGA algorithm there are two individual chromosomes that are chosen randomly, as presented in Figure 9.

Tables 5-7 are for parent1. These tables show how much the group spends on buying packages and single products. They also present that the packages and products encoded in parent1 have served all of the buyers' requests. In Table 7, the package cost is \$740 and the single items cost is \$686. And, the total cost of buying products for all buyers is \$1,326, which is higher than the reservation price of all buyers. As shown in Table 4, the total reservation of all buyers is \$1,292. Based on (9), the fitness value of parent1 becomes 1,292 − 1,326 = −\$34. Tables 8-10 are for parent2. They demonstrate that parent2 is better than parent1. For parent2, the total cost of buying products for all buyers is \$1,275. As a result, the fitness value of parent2 is 1,292 − 1,275 = \$17. This means that parent2 is better than parent1 because it yields a better profit to the group.

TABLE 4. Buyers' demands and reservation prices

Buyer	Item1		Item2		Item3		Item4		Total Reservation (\$)
	Number of items	Reservation price (\$)	Number of items	Reservation price (\$)	Number of items	Reservation price (\$)	Number of items	Reservation price (\$)	
b_1	0	–	2	33	0	–	1	12	78
b_2	0	–	0	–	2	131	0	–	262
b_3	0	–	2	36	2	138	1	11	359
b_4	2	70	1	33	3	140	0	–	593
Total	2	–	5	–	7	–	2	–	1292

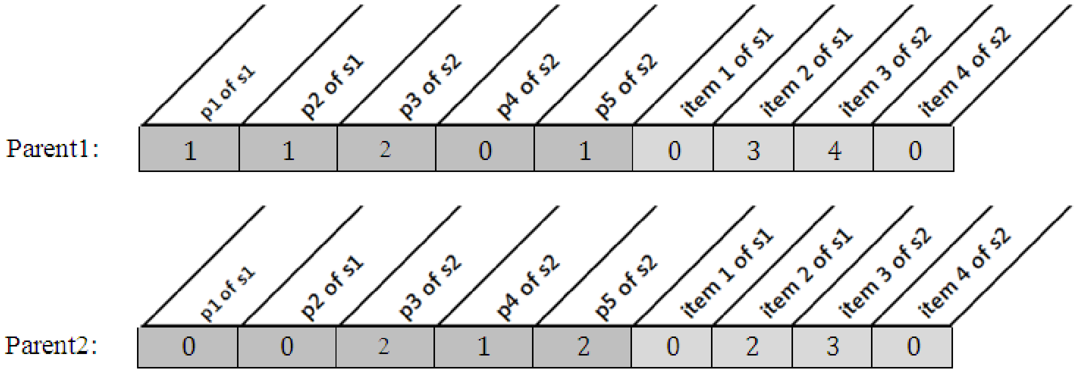


FIGURE 9. Example of chromosomes

TABLE 5. Packages (parent1)

Packages		Item1	Item2	Item3	Item4	Cost (\$)
p1	1	1	1	–	1	100.00
p2	1	1	–	–	1	85.00
p3	2	–	–	2	2	290.00
p4	0	–	–	–	–	0.00
p5	1	–	1	1	1	165.00
Total packages		2	2	3	5	= 100 + 85 + 290 + 165
					Total	**640.00**

TABLE 6. Single items (parent1)

	Item1	Item2	Item3	Item4	Cost (\$)
Single items	0	3	4	0	$= 0 * 76 + 3 * 38 + 4 * 143 + 0 * 165$
				Total	**686**

TABLE 7. Total cost of buying packages and single items (parent1)

	Item1	Item2	Item3	Item4	Cost ($)
Package	2	2	3	5	740
Single item	0	3	4	0	686
Total items	2	5	7	5	$= 640 + 686$
				Total cost	**1,326.00**

TABLE 8. Packages (parent2)

		Item1	Item2	Item3	Item4	Cost ($)
p1	0	–	–	–	–	0.00
p2	0	–	–	–	–	0.00
p3	2	–	–	2	2	290.00
p4	1	2	1	–	–	150.00
p5	2	–	2	2	2	330.00
Total packages		2	3	4	4	$= 290 + 150 + 330$
					Total	**770.00**

TABLE 9. Single items (parent2)

	Item1	Item2	Item3	Item4	Cost ($)
Single items	0	2	3	0	$= 0 * 76 + 2 * 38 + 3 * 143 + 0 * 165$
				Total	**505**

TABLE 10. Total cost of buying packages and single items (parent2)

	Item1	Item2	Item3	Item4	Cost ($)
Package	2	3	4	4	770
Single item	0	2	3	0	505
Total items	2	5	7	4	$= 770 + 505$
				Total cost	**1,275**

5.2. **Example of mating of two chromosomes in the FBGGA algorithm.** Let us now do a single-point crossover of parent1 and parent2. Let us assume that the crossover point is 4. An example of a single-point crossover can then be presented in Figure 10. From this breeding, two offspring are created. It must be evaluated if they are able to support all of the buyers' requests as shown in Table 4. Tables 11-13 are the evaluation for offspring1. Table 13 points out that the offspring can support all buyers' requests. Consequently, the fitness value of offspring1 is 1,292 − 1,291 = \$1. We also examined offspring2 and the results are presented in Tables 14-16. Table 16 shows that offspring2 is good because its fitness value is 1,292 − 1,410 = −\$118. Therefore, only offspring1 may survive into the next generation since its fitness value is better than that of parent1. In the general process of the genetic algorithm, the mating of two random chromosomes can be repeated several times in one generation and done repeatedly to complete the maximum number of generations.

6. **Experimental Results.** The development of the web-based application for establishing price-based buyer groups with bundles of items will be illustrated in this section. The application "Buyer Coalition" was developed in order to illustrate how the FBGGA

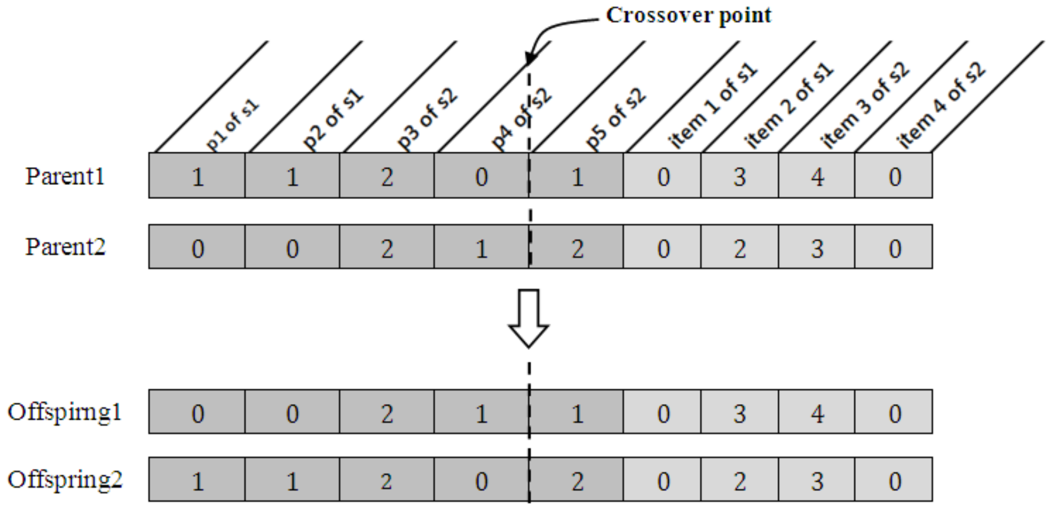


FIGURE 10. Example of a single-point crossover

TABLE 11. Packages (offspring1)

		Item1	Item2	Item3	Item4	Cost ($)
p1	0	–	–	–	–	0.00
p2	0	–	–	–	–	0.00
p3	2	–	–	2	2	290.00
p4	1	2	1	–	–	150.00
p5	1	–	1	1	1	165.00
Total packages		2	2	3	3	$= 290 + 150 + 165$
					Total	**605.00**

TABLE 12. Single items (offspring1)

	Item1	Item2	Item3	Item4	Cost ($)
Single items	0	3	4	0	$= 0 * 76 + 3 * 38 + 4 * 143 + 0 * 165$
				Total	**686**

TABLE 13. Total cost of buying packages and single items (offspring1)

	Item1	Item2	Item3	Item4	Cost ($)
Package	2	2	3	3	605
Single item	0	3	4	0	686
Total items	2	5	7	3	$= 605 + 686$
				Total cost	**1,291**

outlined in Section 5 works in the real world. In this paper, we focus on facilitating the formation of buyer groups and ensuring that all buyers receive a lower price for their products without buying in large volumes.

6.1. **Initial parameters for the FBGGA algorithm.** The web-based application is designed to be accessed by buyers anywhere via the Internet using any Internet web browser. The application was implemented in the PHP language and was done on a notebook with an Intel Core i7-3612QM Ram 8GB VGA HD Graphics 4000. The initial

Table 14. Packages (offspring2)

		Item1	Item2	Item3	Item4	Cost ($)
p1	1	1	1	–	1	100.00
p2	1	1	–	–	1	85.00
p3	2	–	–	2	2	290.00
p4	0	–	–	–	–	0.00
p5	2	–	2	2	2	330.00
Total packages		2	3	4	6	$= 100 + 85 + 290 + 330$
					Total	**805.00**

Table 15. Single items (offspring2)

	Item1	Item2	Item3	Item4	Cost ($)
Single items	0	2	3	0	$= 0 * 76 + 2 * 38 + 3 * 143 + 0 * 165$
				Total	**505**

Table 16. Total cost of buying packages and single items (parent2)

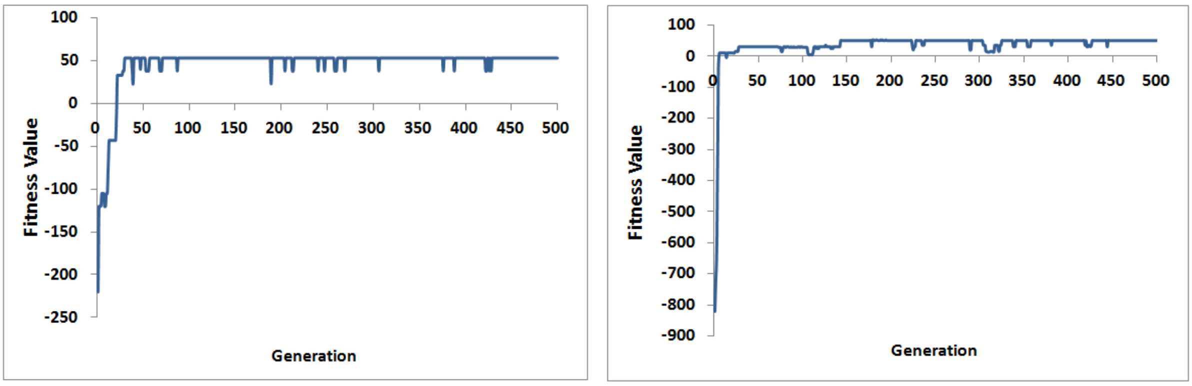
	Item1	Item2	Item3	Item4	Cost ($)
Package	2	3	4	6	805
Single item	0	2	3	0	505
Total items	2	5	7	6	$= 805 + 505$
				Total cost	**1,410**

values are found during the experiment. Based on our experiment, we allow the program proceeds until the best solution does not change to a better value. By doing this, we found that the FBGGA algorithm yields the best solution at 200 generations, which can be seen in Figure 11(a). Therefore, the initial value for the maximum number of generations is 200. In addition, we also found that having different population sizes also affects the quality of the algorithm. If the population of chromosome is low, the best solution will be bad. The example result of our algorithm, where Population_size = 100, is presented in Figure 12. The best result is found around Gen = 300, but the best result becomes lower when it works longer. Consequently, we have run the program multiple times to see which initial parameter values will steer the algorithm toward the best result. Finally, the parameter values that the FBGGA algorithm uses are illustrated in Table 17.

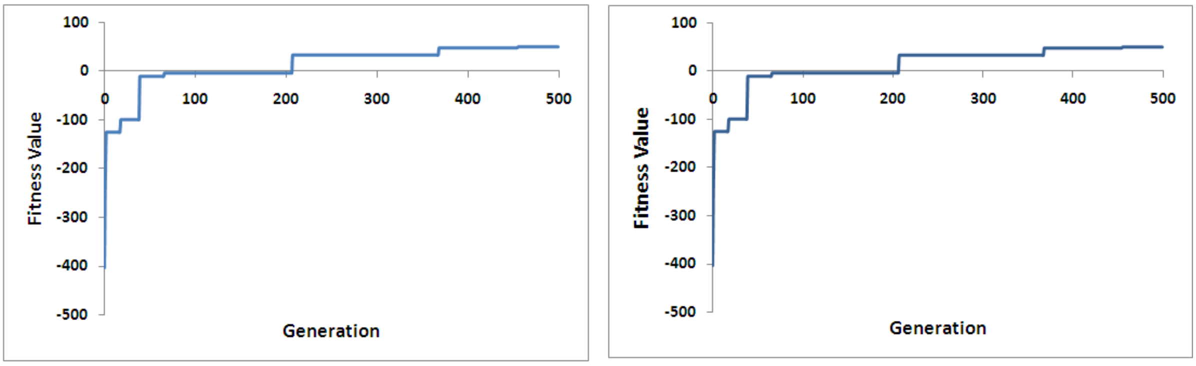
6.2. **Limitations of the system.** The limitations of the system are listed below.

- Buyers have limited power in purchasing and they cannot deal directly with the seller.
- Collaborating with each participant will not be taken into consideration.
- A buyer joins the purchasing group after the price lists are ready.
- The payment and product transfers are out of scope.
- We suppose that the seller has an unlimited number of products.
- All buyers in the group are trusted. If the requirement has been made and the group formation is successful, each individual buyer must pay for their requirements.
- The profit of all buyers earned by the group formation will be returned to all participants. However, this feature is out of our scope.

6.3. **Comparing roulette-wheel selection and generational replacement.** In this section, we provide an example to illustrate how our algorithm works by using the tables



(a) Roulette-wheel selection



(b) Generational replacement

FIGURE 11. Comparison results, where initial parameters are $p_c = 0.5$ and $p_m = 0.05$

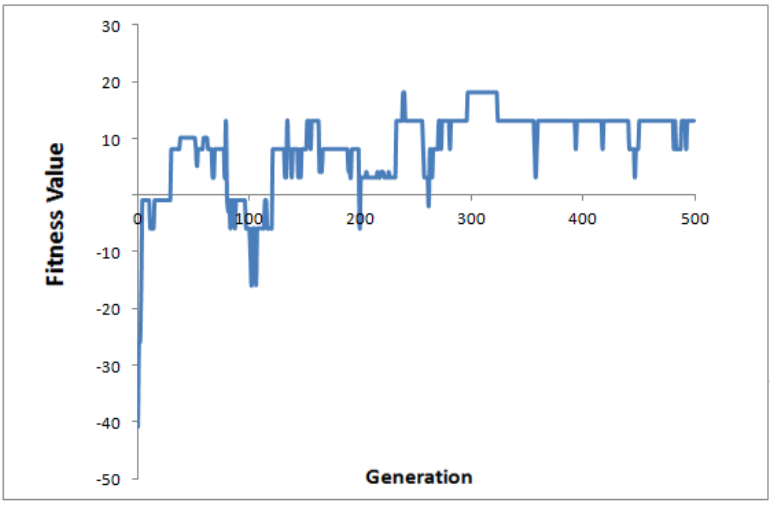


FIGURE 12. Example result of the FBGGA algorithm, where Population_size = 100, $p_c = 0.50$ and $p_m = 0.05$

in the previous sections. Since the selection operator affects significantly the results on the convergence of GAs, two different techniques for selecting two parents, a roulette-wheel selection and a generational replacement, were tested to create offspring. Let us suppose Table 2 is a list of products, and Table 3 is a list of mixed products which are sold at a discount price offered by sellers. In addition, we assume that four buyers in Table 4

TABLE 17. Initial parameters for the FBGGA algorithm

Parameter	Meaning	Value
Max_Gen	Maximum number of generations	200
Population_size	Population size of chromosomes	200
p_c	Probability of crossover	0.50
p_m	Probability of mutation	0.05

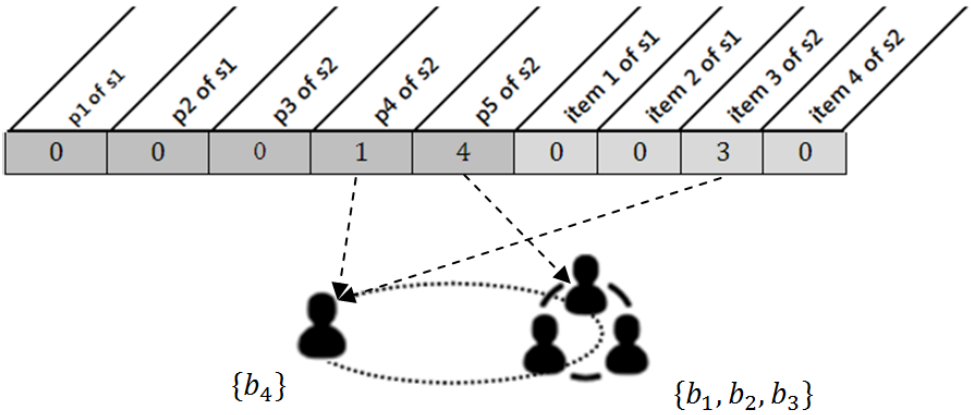


FIGURE 13. The best chromosome received from the FBGGA algorithm and divided subgroups of buyers, where the fitness value is \$53

are participating in the group. Figure 11(a) shows the results of the FBGGA algorithm with roulette-wheel selection, and Figure 11(b) presents the results of the algorithm with generational replacement. The Y axis represents the fitness value. And the X axis is the number of generations. We can see that the algorithm with roulette-wheel selection provides optimal solutions in early generations (see Figure 11(a)). Figure 11(b) is the result of the algorithm with generational replacement, where the best 200 chromosomes of the current population will be selected in the next generation. The figure shows that the algorithm with generational replacement gives the better results after 200 generations. Hence, we can see that the FBGGA algorithm with roulette-wheel selection outperforms that using generational replacement. Nevertheless, the best fitness value of all figures is \$53 as presented in Figure 13. The calculation can be illustrated as below. Let us divide all of the buyers into two subgroups, $C_1 = \{b_4\}$ and $C_2 = \{b_1, b_2, b_3\}$. The first group needs to buy one package of p4 and three items of item3 from s_2, while C_2 needs to buy four packages of p5. Therefore, the total cost of buying products for all buyers is \$1,239, see Tables 18-20 for the detail. As previously calculated, the total reservation of all buyers in Table 4 is \$1,292. As a result, the total benefit of the groups is $1{,}292 - 1{,}239 = \$53$.

6.4. **User interface.** In this section, we describe the features of our web-based application presented through the type of users. Since authentication is required to maintain the privacy of the system, a prospective buyer must sign up for membership. Each buyer has to fill in the given fields, namely username, email, password and confirm password. Then, the user clicks on the "Signup" button to register. The buyer's information is saved in the database located on the server.

Group leader: In this application, the first person in the group will be designated as the leader because this person starts recruiting the buyers to join the group. Therefore, the group leader can do more things than the other buyers in order to manage the group. In addition, an important task that the group leader must do is to enter the sellers'

Table 18. Packages

		Item1	Item2	Item3	Item4	Cost ($)
p1	0	–	–	–	–	0.00
p2	0	–	–	–	–	0.00
p3	0	–	–	–	–	0.00
p4	1	2	1	–	–	150.00
p5	4	–	4	4	4	660.00
Total packages		2	5	4	4	$= 150 + 660$
					Total	**810.00**

Table 19. Single items

	Item1	Item2	Item3	Item4	Cost ($)
Single items	0	0	3	0	$= 0 * 76 + 0 * 38 + 3 * 143 + 0 * 165$
				Total	**429**

Table 20. Total cost of buying packages and single items

	Item1	Item2	Item3	Item4	Cost ($)
Package	2	5	4	4	810
Single item	0	0	3	0	429
Total items	2	5	7	4	$= 810 + 429$
				Total cost	**1,239**

information including the products and packages (a bundle of several products) sold by the sellers at a discount price. The leader must add the single products sold at the original price by the sellers. After that, the packages of products can be entered by clicking the “make package” button. An example screenshot for entering the single products is illustrated in Figure 14. It must be emphasized that “package” in this paper means that the bundle of items and the price of a package must be cheaper than the total price of all items in the package supplied by the seller. A screenshot of this feature is presented in Figure 15. Moreover, the group criteria can be set by the leader, such as group name, and deadline for joining. An example screenshot for this feature is presented in Figure 16. The group leader must constantly monitor the group’s progress. If the group formation made by the FBGGA algorithm yields the perfect outcome, where all buyers in the group receive their requested products at low cost, the leader can stop recruiting more members to join before the deadline. In such a case, the group formation can be considered a “success”. Otherwise, if the deadline is met and the FBGGA algorithm cannot carry out group formation, the group is then considered a failure. If the group formation is considered a “success”, the buying details of each buyer will be displayed for the group leader. Figure 17 shows an example screenshot for the group leader; it presents the detail of each buyer resulting from the success of the group formation.

Buyers: When all products and packages have been successfully entered into the system, buyers who are interested in these products in a certain list can post the requirement as well as the reservation price. The buyers can see the ongoing groups which are still active. Originally, ongoing groups last until the deadline, but they can be closed before this time if the group leader satisfies the results derived from the FBGGA algorithm. The requirement of each buyer will be stored in the database. The requirement can be either canceled or edited before the deadline. More importantly, the requirement must

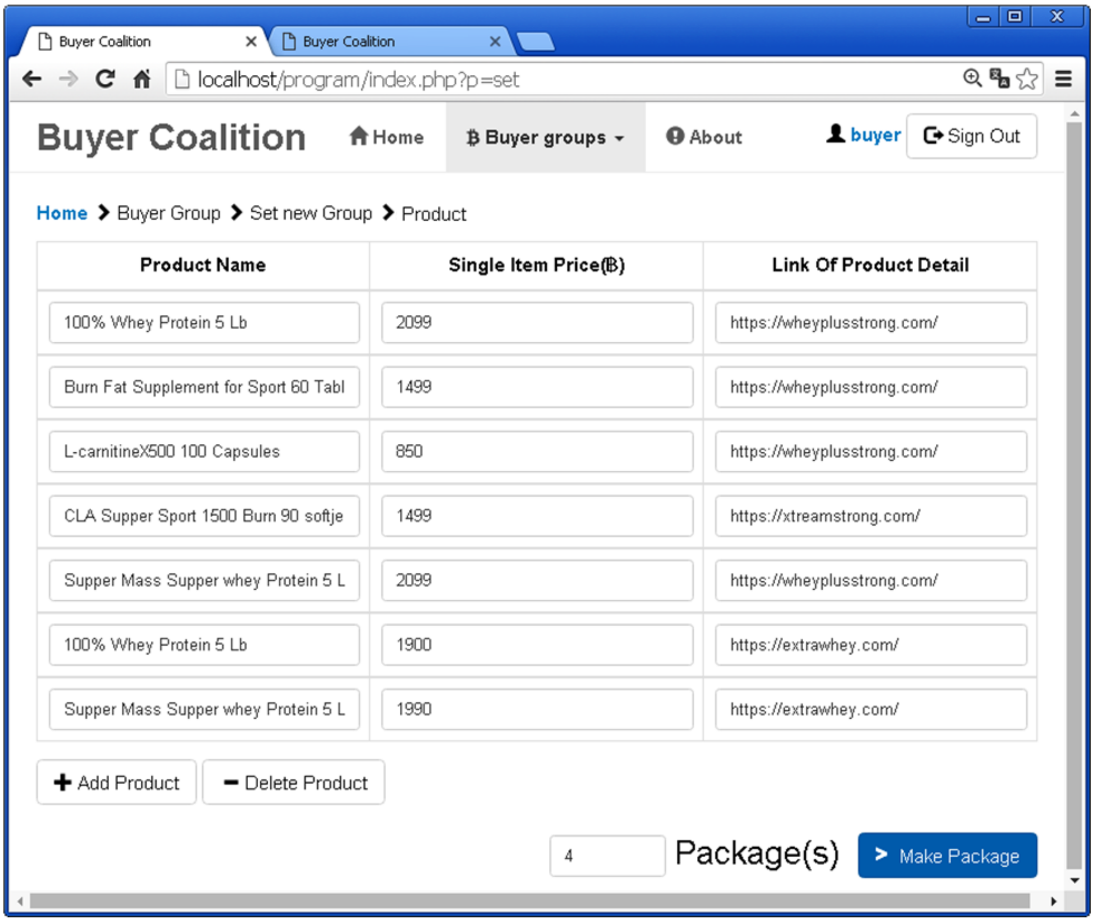


FIGURE 14. Example screenshots for adding single products

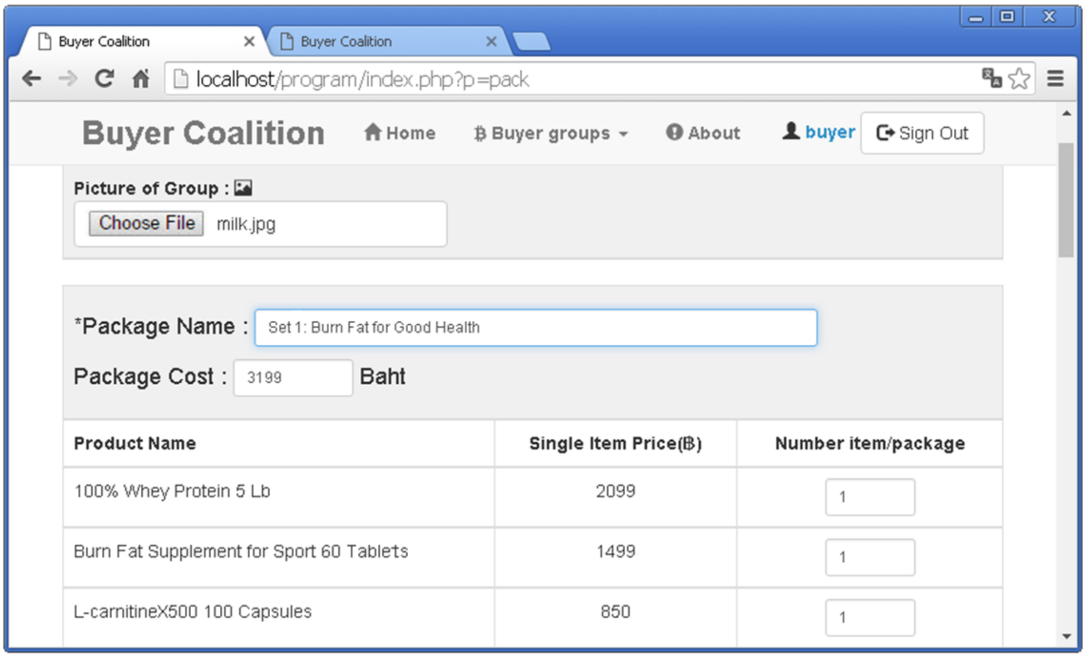


FIGURE 15. An example screenshot for making the package (bundle of items)

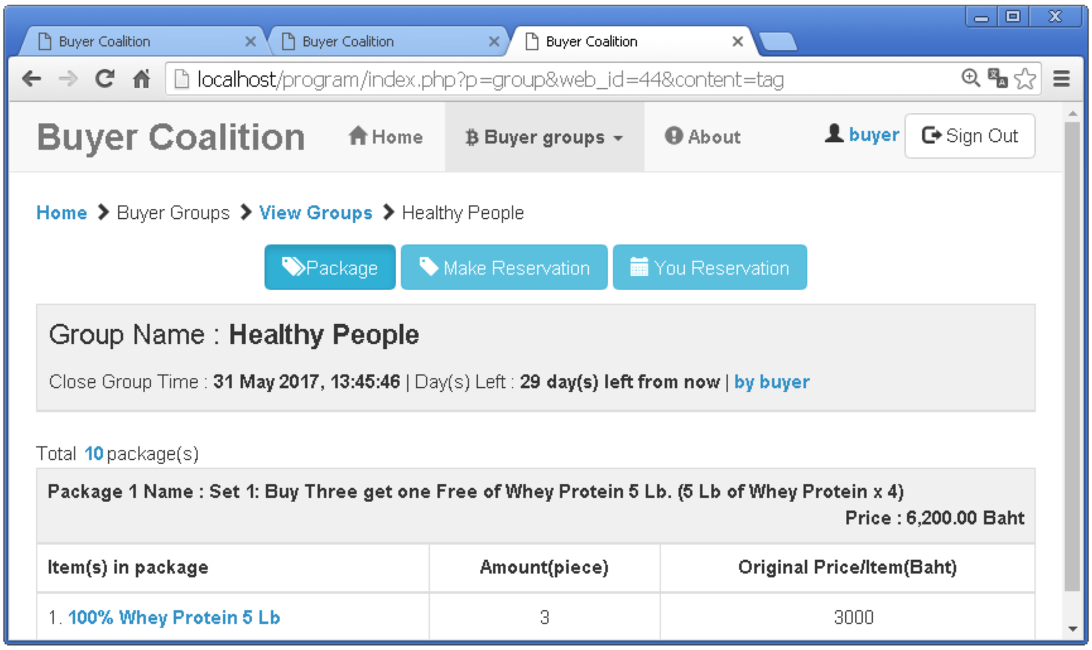


FIGURE 16. An example screenshot for setting the group criteria

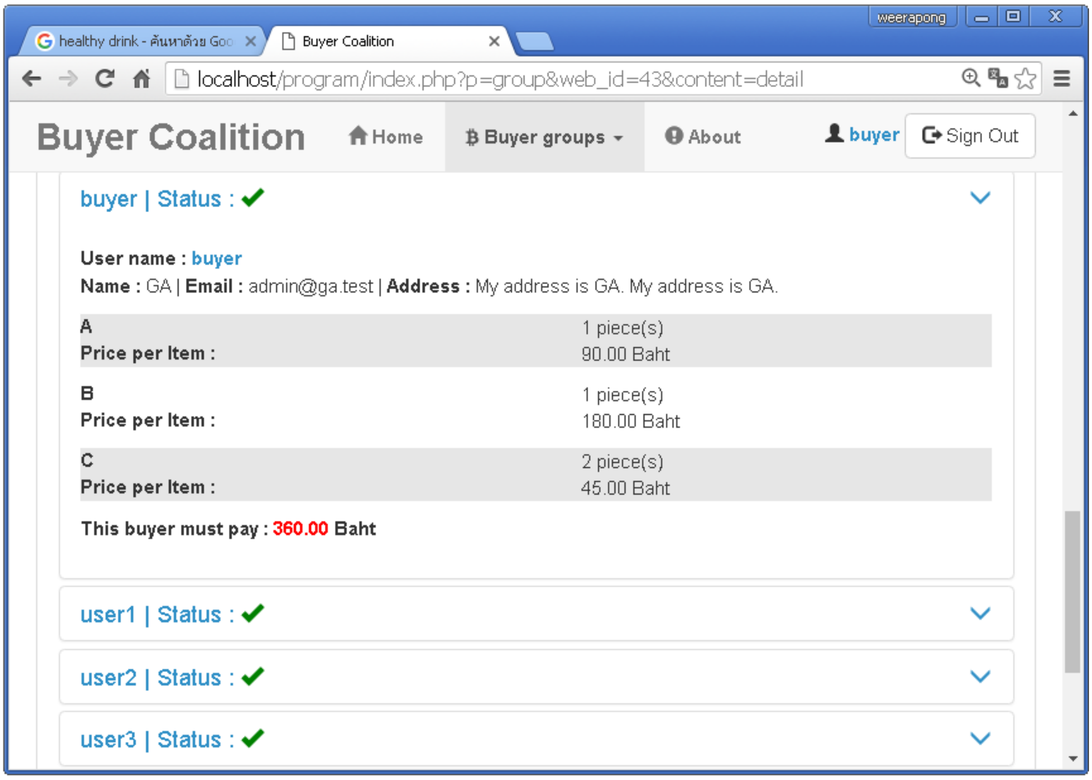


FIGURE 17. An example screenshot for screenshot for a group leader

be confirmed by the buyer. The buyers are likely to post the reservation price as low as possible. Hence, the buyers will see the price list for the decision. In some cases, some buyers may be excluded out of the group after the work of the FBGGA algorithm. This situation can happen if some buyers make an unacceptable reservation price for the product. If this buyer is part of the group, it can possibly damage its success. While the

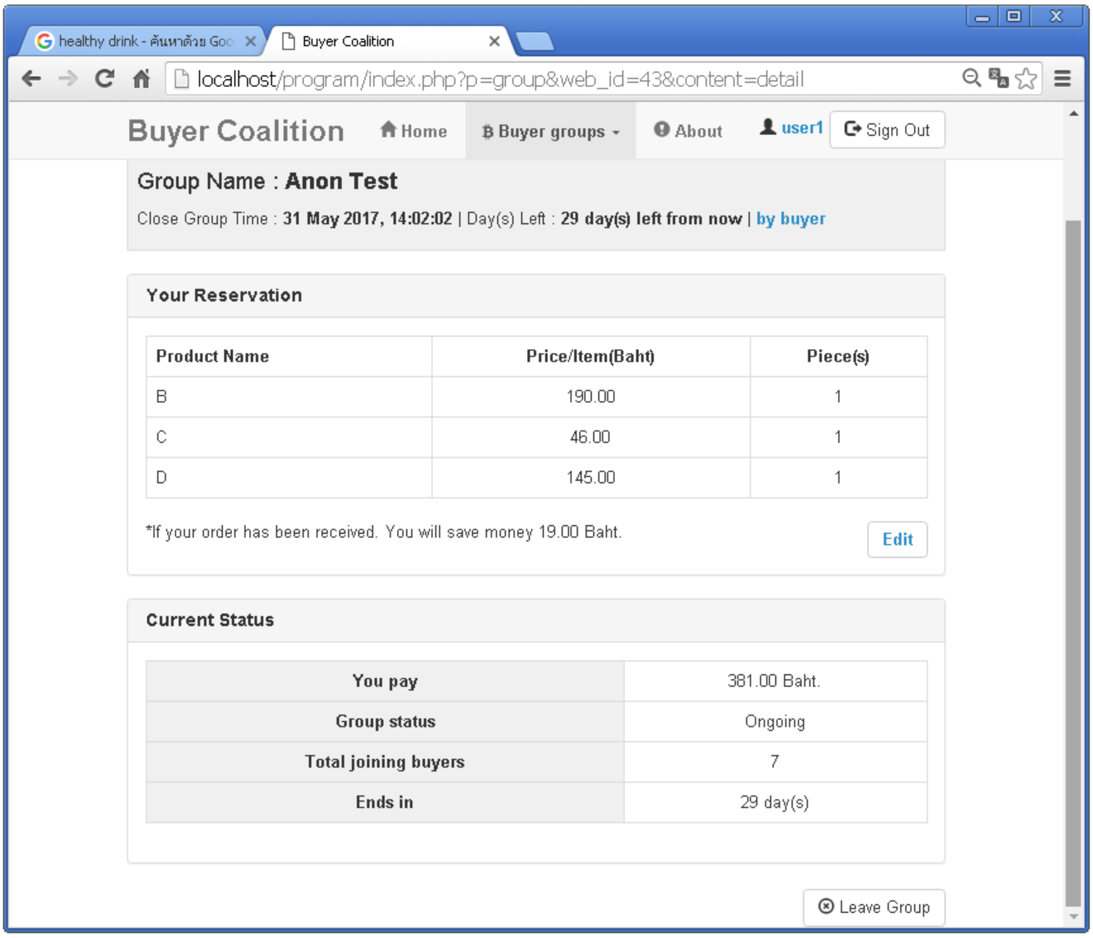


FIGURE 18. An example screenshot for buyers in making requirements

group is still active, each individual buyer can see its current status, such as the number of buyers joining, the deadline, and the details of the buyer's reservation. An example screenshot for this feature is presented in Figure 18. To establish a fair outcome, buyers' requirements must be kept secret, especially reservation prices. If the group is successful formed by the FBGGA algorithm, all members of the group will see the result including the benefit of this buying.

7. **Discussion.** Time complexity analysis is a part of computational complexity theory that is commonly used to estimate the time taken for running an algorithm. In our proposed algorithm, the FBGGA has a large number of generations, so we can go with the number of generations for time complexity. Furthermore, the time complexity of a single generation can be divided into a fitness function and three main operators, which are the roulette wheel selection operator, the single-point crossover operator and the single-point mutation operator. These operators including the fitness function have their own complexity that is how the calculation does. The fitness function is important because it is employed to determine the quality of candidate solutions. Given the chromosome structure of buyer coalition presented in Figure 1, where p is the number of packages and r is the number of products offered by online sellers, both p and r will be fixed and small. Therefore, the fitness function of each chromosome calculates the total benefit of the groups (n buyers) as presented previously. Running time is expressed as a function of its input. Then, we can simply say that the running time is $O(n)$. As can be seen in the algorithm of the roulette wheel selection for FBGGA, the running time of this operator is

based on the number of a population of chromosomes which is a set of possible solutions to the problem. Since the number of a population of chromosomes is fixed as shown in Table 17, the running time of the roulette wheel selection is trivial. This is the same to both the single-point crossover operator and the single-point mutation operator. Their running time is based on the value of p and r, which are small as well. During each generation, it has $O(n)$ time complexity. The FBGGA proceeds until the predefined value of maximum number of generations (Max_Gen) is met, and then the time complexity for the FBGGA is Max_Gen*n. However, we can ignore the value of Max_Gen because this value can be considered as a coefficient and becomes insignificant when compared to the large number of buyers (n). Finally, time complexity of the FBGGA is $O(n)$.

8. **Conclusions and Future Work.** In this paper, the FBGGA algorithm was used to form groups of heterogeneous buyers. The algorithm is based on genetic algorithms in which a roulette-wheel selection aggregates a number of buyer-selected items that are purchased from sellers. We assume that there are multiple buyers in the e-marketplace and that there are different ways to meeting buyers' requirements. Furthermore, the buyer coalition is constructed to obtain the highest utility received from the sellers. Based on the experimental results, the algorithm with roulette wheel selection is more efficient in convergence than the algorithm based on generational replacement. The algorithm was implemented as a web-based application in order to illustrate how it could work in the real world. The empirical example demonstrated that the proposed algorithm is able to search for optimized solutions in early generations. For future work, more factors affecting the coalition formation of buyers such as the location of buyers and buyer's coalition will be added to the model. Another issue for future work is to optimize the application for larger and more sophisticated problems by eliminating the limitations of the system discussed in the paper.

REFERENCES

[1] T. Matsuo, T. Ito and T. Shintani, A volume discount-based allocation mechanism in group buying, *Proc. of International Workshop on Data Engineering Issues in E-Commerce*, pp.59-67, 2005.

[2] J. Yamamoto and K. Sycara, A stable and efficient buyer coalition formation scheme for E-marketplaces, *Proc. of the 5th Int'l Conf. Autonomous Agents*, New York, pp.576-583, 2001.

[3] C. Li and K. P. Sycara, Algorithm for combinatorial coalition formation and payoff division in an electronic marketplace, *Proc. of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-02)*, pp.120-127, 2002.

[4] S. Narabin and V. Boonjing, A location-price-based buyer coalition, *IEEE the 12th International Conference on Computer and Information Technology (CIT)*, pp.416-420, 2012.

[5] F. S. Hsieh and J. B. Lin, Assessing the benefits of group-buying-based combinatorial reverse auctions, *Electronic Commerce Research and Applications*, vol.11, no.4, pp.407-419, 2012.

[6] V. D. Dang and N. R. Jennings, Polynomial algorithms for clearing multi-unit single-item and multi-unit combinatorial reverse auctions, *Proc. of the 15th European Conference on Artificial Intelligence*, pp.23-27, 2002.

[7] Y. C. Chen, J. H. Wu, L. Peng and R. C. Yeh, Consumer benefit creation in online group buying: The social capital and platform synergy effect and the mediating role of participation, *Electronic Commerce Research and Applications*, vol.14, no.6, pp.499-513, 2015.

[8] J. Chen, L. Guan and X. Cai, Analysis on buyers' cooperative strategy under group-buying price mechanism, *Journal of Industrial Management and Optimization*, vol.9, pp.291-304, 2013.

[9] H. R. Varian and J. Repcheck, *Intermediate Microeconomics: A Modern Approach*, WW Norton & Company, New York, 2010.

[10] T. W. Sandholm and V. R. Lesser, Coalition formation among bounded rational agents, *Proc. of the 14th International Joint Conference on Artificial Intelligence*, no.1, pp.662-669, 1995.

[11] D. Sarne and S. Kraus, Buyer's coalition for optimal search, *Proc. of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, pp.1225-1226, 2005.

[12] J. Chen, X. Chen and X. Song, Comparison of the group-buying auction and the fixed pricing mechanism, *Decision Support Systems*, vol.43, no.2, pp.445-459, 2007.

[13] M. Klusch and A. Gerber, Dynamic coalition formation among rational agents, *IEEE Intelligent Systems*, vol.17, no.3, pp.42-47, 2002.

[14] C. Li, S. Chawla, U. Rajan and K. Sycara, Mechanism design for coalition formation and cost sharing in group-buying markets, *Electronic Commerce Research and Applications*, vol.3, no.4, pp.341-354, 2002.

[15] T. Sandholm, K. Larson, M. Andersson, O. Shehory and F. Tohmé, Anytime coalition structure generation with worst case guarantees, *arXiv Preprint cs/9810005*, 1998.

[16] T. Van Horn, N. Gustafsson and D. Woodford, *U.S. Patent No. 6,047,266*, US Patent and Trademark Office, Washington, D.C., 2000.

[17] G. Ni, L. Luo, Y. Xu, J. Xu and Y. Dong, Optimal decisions on group buying option with a posted retail price and heterogeneous demand, *Electronic Commerce Research and Applications*, vol.14, no.1, pp.23-33, 2015.

[18] L. Boongasame, H. F. Leung, V. Boonjing and D. K. Chiu, Forming buyer coalitions with bundles of items, *KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*, pp.714-723, 2009.

[19] T. Ito, H. Ochi and T. Shintani, A group buy protocol based on coalition formation for agent-mediated e-commerce, *Proc. of the 2nd International Conference on Software Engineering, Artificial Intelligence, Networking & Parallel/Distributed Computing (SNPD-01)*, Nagoya, Japan, 2001.

[20] M. Hyodo, T. Matsuo and T. Ito, An optimal coalition formation among buyer agents based on a genetic algorithm, *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pp.759-767, 2003.

[21] A. Yasar, Pricing of multi-product bundles to match supply and demand, *Proc. of 2004 IEEE Systems and Information Engineering Design Symposium*, 2004.

[22] L. He and T. R. Ioerger, Combining bundle search with buyer coalition formation in electronic markets: A distributed approach through negotiation, *Proc. of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)*, New York, pp.1438-1439, 2004.

[23] L. B. Booker, Improving the performance of genetic algorithms in classifier systems, *Proc. of the 1st International Conference on Genetic Algorithms and Their Applications*, pp.80-92, 1985.

[24] C. Karr and L. M. Freeman (eds.), *Industrial Applications of Genetic Algorithms*, CRC Press, 1998.

[25] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, MIT Press, 1992.

[26] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Optimization*, John Wiley & Sons, 2000.

[27] B. J. Babb, F. Moore and M. Peterson, Optimized satellite image compression and reconstruction via evolution strategies, *SPIE Defense, Security, and Sensing*, International Society for Optics and Photonics, 2009.

[28] K. Sastry, D. E. Goldberg and G. Kendall, Genetic algorithms, in *Search Methodologies*, E. K. Burke and G. Kendall (eds.), Springer, US, 2014.

[29] S. Baluja and R. Caruana, Removing the genetics from the standard genetic algorithm, *Proc. of the 12th International Conference on Machine Learning*, pp.38-46, 1995.

[30] C. K. Ting, S. T. Li and C. Lee, On the harmonious mating strategy through tabu search, *Information Sciences*, vol.156, no.3, pp.189-214, 2003.

[31] D. K. Dey, Mathematical study of adaptive genetic algorithm (AGA) with mutation and crossover probabilities, *Compusoft International Journal of Advanced Computer Technology*, vol.3, no.5, 2014.

[32] D. Bhandari, C. A. Murthy and S. K. Pal, Variance as a stopping criterion for genetic algorithms with elitist model, *Fundamenta Informaticae*, vol.120, no.2, pp.145-164, 2012.