Article

Finding Effective Item Assignment Plans with Weighted Item Associations Using a Hybrid Genetic Algorithm

Minho Ryu [1], Kwang-Il Ahn [2] and Kichun Lee [3,*]

[1] Vision AI Labs, SK Telecom, Seoul 04539, Korea; ryumin93@sktbrain.com
[2] Department of Interdisciplinary, Industrial Studies, Hanyang University, Seoul 04763, Korea; ahn21@hanyang.ac.kr
[3] Department of Industrial Engineering, College of Engineering, Hanyang University, Seoul 04763, Korea
* Correspondence: skylee@hanyang.ac.kr; Tel.: +82-22-220-0478

Abstract: By identifying useful relationships between massive datasets, association rule mining can provide new insights to decision-makers. Item assignment models based on association between items are used to place items in a retail or e-commerce environment to increase sales. However, existing models fail to combine these associations with item-specific information, such as profit and purchasing frequency. To find effective assignments with item-specific information, we propose a new hybrid genetic algorithm that incorporates a robust tabu search with a novel rectangular partially matched crossover, focusing on rectangular layouts. Interestingly, we show that our item assignment model is equivalent to popular quadratic assignment NP-hard problems. We show the effectiveness of the proposed algorithm, using benchmark instances from QAPLIB and synthetic databases that represent real-life retail situations, and compare our algorithm with other existing algorithms. We also show that the proposed crossover operator outperforms a few existing ones in both fitness values and search times. The experimental results show that not only does the proposed item assignment model generates a more profitable assignment plan than the other tested models based on association alone but it also obtains better solutions than the other tested algorithms.

Keywords: association rules; cross-selling; item assignment; hybrid genetic algorithm

1. Introduction

Association rule mining uncovers interesting patterns that may exist within large amounts of co-occurring elements in a dataset. These interesting patterns include frequent items, association rules, and association rule classifiers that establish relationships between the co-occurring elements or transaction data. A typical association rule consists of an antecedent (or condition) and a consequent (or result), which have a conditional connection such as "if a condition event takes place, how likely a result event is." Due to the existence of transaction data in many industries, association rule mining has been employed in various fields, including item placement, web usage mining, product portfolio identification, cross-selling, and fault diagnosis [1–5]. Extracting significant association between customer and product form [6], disease prediction [7], software defect prediction [8], and pollution monitoring in intelligent cities is also possible using weighted association rule mining [9]. When applying association rule mining to the referred fields, the principles are the same: we wish to discover items that frequently occur together in a database and, subsequently from these frequent items, generate association rules that satisfy a minimum constraint.

One representative application of association rule mining is market basket analysis, which identifies interesting purchasing patterns [10]. Association rule mining has been used in various settings that are related to market basket analysis. Brijs et al. developed

association rules considering business values in item assortment analysis [11]. In a manufacturing context, researchers developed cell formation approaches based on association rule mining in cellular manufacturing systems for parts and machines. Chen used association rule induction to find effective configurations for machines [12], and Liu et al. proposed uncertain association rule mining to incorporate essential production factors such as operation sequences [13]. Association rule mining was also adopted to develop an order clustering approach in distribution centers [14]. Beheshtian-Ardakani used customer loyalty and market segmentation variables for item bundling [15]. The item assignment problem in retail was suggested as a practical application of market basket analysis [1]. Item assignment in retail involves finding effective assignments in a given layout to maximize the chance of cross-selling. A basic principle of item assignment for cross-selling is to place associated items in close proximity, thereby encouraging the sale of such items.

In most applications of association rule mining, including those mentioned above, all items in a dataset are treated uniformly. However, in a retail business, e.g., item-specific information such as cost, price, and sales volume are easily obtainable and quite important. Commonly, managers place emphasis on specific items; e.g., if some items are more profitable, they become more interested. Indeed, the approach of weighting items has been explored by many researchers. Cai et al. suggested weighting frequent items in a database during mining [16], and Tao et al. proposed weighted association rule mining to efficiently generate frequent items [17]. Khan et al. also used weighted association rule mining for datasets with binary and quantitative attributes in weighted settings [18]. Pears et al. generated weight assignments automatically by formulating a model that considers the relationship between items [19]. Utility itemsets or utility rules based on profit were introduced in a market basket analysis [20]. Heydari et al. also formulated a non-linear binary programming problem with weighted associations incorporating interaction information between items with utility information such as unit profit and selling possibility for each item, solving it by a genetic algorithm [21]. While these weighting approaches offer insight into an item assignment for a given layout, they failed to include item–item associations with item-specific information. It would be more realistic and desirable if an item assignment method could incorporate the existence of practical item-specific information, such as profit and sales volume. In Heydari et al., they took into consideration difference in selling possibility depending on allocated location, but we, ignoring such difference, rather focus on interaction among items depending on distance. The rationale is that we focus on practical settings with numerous sites and items as in retail and e-commerce. This study utilized association rule mining in the item assignment problem in a setting with a given layout by considering the weights of items according to profit and frequency counts.

Specifically, this study aimed to improve the existing approaches of association rule mining for the item assignment problem using the following two criteria. First, an item assignment plan must consider not only the strength of correlation between items but also the profit incurred by cross-selling. The measure of association between items is called lift. Assuming that X and Y are the antecedent and consequent for item sets, respectively, lift (X → Y) measures the possibility of cross-selling Y given that X was previously purchased. The measure lift (X → Y) can be also written as $P(X \cap Y)/(P(X)P(Y))$, which relates to point-wise mutual information. When lift (item A → item B) and lift (item A → item C) have the same association strength, item A can be located next to either item B or item C. However, if the profit gained by cross-selling item B with item A is higher than by cross-selling item C with item A, it is reasonable that item A should be located next to item B to maximize sales.

Second, an assignment plan also needs to consider the purchasing frequency of items. Since the association among items is expressed as a ratio of probabilities, it is possible that lower frequency items have a stronger association with each other than higher frequency items. From an economic viewpoint, weakly correlated items with a high frequency can

be more valuable than strongly associated items with a low frequency. In other words, items with low profit can still be beneficial to retailers if the items are sold quite frequently.

Given layouts and item-purchase frequencies, an effective item arrangement is quite important as well. The task of finding an effective item assignment, which is computationally intensive, involves searching for the best assignment with the highest potential for cross-selling. Solvable by searching for all assignments in a factorial way, the item assignment problem can be viewed as a special case of the quadratic assignment problem (QAP) [22], which can be formulated as a mixed-integer linear programming (MILP), a well-known combinatorial optimization problem. QAP is NP-hard, so no exact algorithm can solve the problem in polynomial time. An exact algorithm for QAP such as branch-and-bound and cutting plane methods is possible for a small size n. Recently, numerous researches have been made to develop well-performing heuristic and metaheuristic methods, which can find high-quality solutions in a reasonable time but with no optimality guaranteed. Taillard proposed robust tabu search (RTS) producing high-quality solutions with fast comparison of neighborhood in $0(n^2)$ instead of $0(n^3)$, in which n is the problem size [23]. Benlic et al. proposed breakout local search (BLS), which exploits local search and adaptive perturbation strategies to explore the search space [24]. Their subsequent research also presented a population-based memetic algorithm for QAP, which integrates BLS within the population-based evolutionary algorithm. Dokeroglu proposed hybrid teaching-learning-based optimization using RTS, called as TLBO-RTS, in which two phases of exploration and exploitation exist: all individuals are trained by TLBO algorithm in the first phase and further exploited by RTS [25]. Dokeroglu et al. also proposed an RTS-integrated artificial bee colony (ABC) optimization for QAP (ABC-QAP), a population-based algorithm inspired by the behavior of honeybees [26]. Hybrid genetic algorithm with hierarchical iterated tabu search was proposed to solve for the solution of QAP [27]. In addition, cited heuristic algorithms directly are also applicable to urban transportation networks such as vehicle routing problem and ridesharing user equilibrium problem [28] since they also can be formulated as MILP. Whale optimization algorithm combined with tabu search was proposed for green vehicle routing problem [29].

This study proposes not only an item assignment model incorporating lift, profit, and frequency for the effective assignment of items but also a hybrid genetic algorithm for finding better combinations of item assignments. The proposed model unifies profit and frequency as weighting coefficients of associations among items. Since the scales of profit and frequency may be different depending on the problems, in order to make them dimensionless, relative profit and frequency were used instead of absolute values. To find effective item assignments, we propose a hybrid GA among various heuristic algorithms that incorporate a robust tabu search. We adopt genetic algorithm due to its advantages of fast convergence, easy implementation, and competitive performance when combined with proper local search [30,31]. In the hybrid GA, we propose a special crossover, rectangular partially matched crossover, that is suitable for reflecting a physical layout form.

The rest of this paper is organized as follows. Section 2 describes the item assignment model. Section 3 explains the effect of profit and frequency in an environment with a given layout and defines the weighted item assignment model. Section 4 describes the proposed algorithm. Section 5 presents experimental results. Section 6 concludes the paper.

2. Item Assignment Model

The purpose of item assignment is to determine the most effective in-store locations of items to maximize the possibility of cross-selling. In practice, item assignment based on association rule mining starts by grouping sub-items; that is, sub-items are classified into different categories. Grouping sub-items allows the association information to remain useful, even if some sub-items are discontinued. Usually, the group information of sub-items is predetermined; e.g., diapers and pacifiers are to be grouped as "baby" items. The strengths of the associations are measured across these grouped items.

We apply *lift*, a measure of association, to represent the possibility of cross-selling. Let X and Y be sets of items. Assuming that Pr(X) is the probability of containing X in a transaction database and Pr(Y|X) is the conditional probability of including Y together when a transaction contained X, *lift* (X → Y) is defined as shown in Equation (1):

$$lift(X \to Y) = Pr(Y|X)/Pr(Y) = \frac{Pr(X \cap Y)}{Pr(X)Pr(Y)}. \tag{1}$$

Lift (X → Y) measures how much the possibility of purchasing Y increases after buying X rather than buying Y alone. If *lift* (X → Y) is greater than 1, then X and Y, positively correlated, can be placed together for cross-selling.

Let us assume that n items are to be located at n sites, with one item at each site. Let $I = \{I_1, I_2, \dots, I_n\}$ be the set of items and $S = \{S_1, S_2, \dots, S_n\}$ be the set of sites. Suppose that x_{ij} is 1 (one) if item I_i is located at site S_j and 0 (zero) otherwise, and y_{ij} is 1 if site S_i is adjacent to S_j and 0 otherwise. For a given layout, y_{ij} is given, and x_{ij} is a decision variable. The item assignment model can be stated as shown in Equation (2):

Maximize $\qquad \sum_{i \neq j, k \neq l} lift(I_i \to I_j) \times x_{ik} \times x_{jl} \times y_{kl}$ $\qquad\qquad$ (2)

Subject to $\qquad \sum_{i=1}^{n} x_{ik} = 1, k = 1, \dots, n,$

$\qquad\qquad\qquad \sum_{k=1}^{n} x_{ik} = 1, i = 1, \dots, n,$

$\qquad\qquad\qquad x_{ik} = \begin{cases} 1, & \text{if item } I_i \text{ is located at site } S_k \\ 0, & \text{otherwise.} \end{cases}$

The objective function in Equation (2) represents the sum of possibilities of cross-selling for a given assignment. Each item must be located at one site by the constraint $\sum_{k=1}^{n} x_{ik} = 1$. Likewise, each site must hold only one item by the constraint $\sum_{i=1}^{n} x_{ik} = 1$. According to y_{kl} in the objective function, the possibility of cross-selling is added to the objective value only if the two items are placed at sites near one another.

For example, suppose that six items are assigned to six sites and the lifts of two items are given in Table 1. Lift has the characteristic that it is symmetric: *lift* (X → Y) = *lift* (Y → X). Therefore, we give lifts of items in the form of a triangular matrix, as illustrated in Table 1.

Table 1. Exemplary *lift* values between 6 items.

	A	**B**	**C**	**D**	**E**	**F**
A	-	1.6	0	1.3	0	1.2
B	1.6	-	1.1	1.4	0	1.3
C	0	1.1	-	1.3	1.5	0
D	1.3	1.4	1.3	-	0	1.3
E	0	0	1.5	0	-	1.3
F	1.2	1.3	0	1.3	1.3	-

Let us assume that the layout of the six sites and an item assignment are as shown in Figure 1a,b, respectively. Let us denote Figure 1b as BASE_ASSIGN. Then, the total possibility of cross-selling can be calculated as follows. Item C is placed near item B and item E, and *lift* (C → B) and *lift* (C → E) are added to the total possibility of cross-selling, i.e., 1.1+1.5. Likewise, the seven lifts of items placed in close proximity are added to the total possibility of cross-selling. As a result, the total possibility of cross-selling is 1.1 + 1.3 + 0 + 1.3 + 1.5 + 1.6 + 1.3 = 8.1.
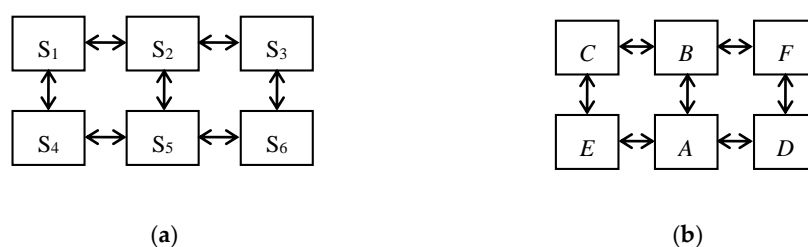
(**a**) (**b**)

Figure 1. (**a**) A rectangular layout of six sites. (**b**) Item assignment, denoted by BASE_ASSIGN.

The task of item assignment is to search for an assignment plan that maximizes the sum of the possibilities of cross-selling. The task is an NP problem, and various heuristic algorithms of assignment can be conducted.

3. Item Assignment Based on Weighted Associations

This section proposes an item assignment problem based on weighted associations. Since most of the decisions made in retail industries are related to sales and profits, the proposed item assignment model, which considers the profits and frequencies of purchasing items, is realistic and worthwhile.

3.1. Effects of Profit and Frequency

In Section 2, the final assignment of items for cross-selling was obtained based only on the correlations between the items. However, these associations between the items do not consider the profit increase due to cross-selling. Since all retailers are profit-seeking, increased profit is a more important factor than the strength of correlations among items for item assignment in a retail environment.

Given BASE_ASSIGN, assume that the expected profits increased due to cross-selling are as shown in Table 2a. Then, the numbers considering both lifts and the profit is shown in Table 2b. Allocating item C next to item D promotes more profit than placing item C next to other items according to Table 2b. In the same sense, item B should be located next to item F.

Table 2. (**a**) Expected profit increase due to cross-selling items. (**b**) Expected profit considering lifts.

	A	**B**	**C**	**D**	**E**	**F**
A	–	10	10	20	10	10
B	10	–	10	10	10	30
C	10	10	–	20	10	10
D	20	10	20	–	10	10
E	10	10	10	10	–	10
F	10	30	10	10	10	–
	A	B	C	D	E	F
A	–	**16**	**0**	**26**	**0**	**12**
B	16	–	11	14	0	39
C	0	11	–	26	15	0
D	26	14	26	–	0	13
E	0	0	15	0	–	13
F	12	39	0	13	13	–

As shown in Figure 2a, the item assignment, denoted as PROFIT_ASSIGN, is an updated version from BASE_ASSIGN, considering profit as well as lifts. Logically,

PROFIT_ASSIGN is a better assignment than BASE_ASSIGN from an economic view-point.
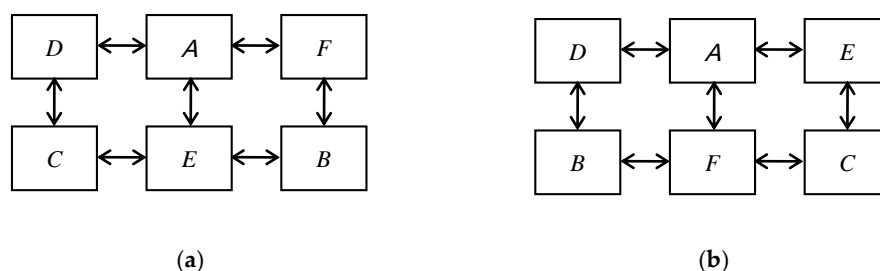

(**a**)　　　　　　　　　　　　　　　　　(**b**)

Figure 2. (**a**) Item assignment, denoted by PROFIT_ASSIGN. (**b**) Item assignment, denoted by PROF_FRE_ASSIGN.

The PROFIT_ASSIGN in Figure 2a considers not only the associations between the items but also the profit increase due to cross-selling. Note that the increased profit applied to the item assignment problem is a unit profit for one occurrence of the purchased items. Considering the frequency of purchasing items, in addition to profit, is a realistic approach for the item assignment problem.

Suppose that the number of transactions containing items cross-sold in a database is provided as shown in Table 3a. The numbers in Table 3 indicate how frequently the paired items are sold. Then, the values in Table 3b correspond to expected profits by taking into account the frequencies of occurrence as well as the lifts. The PROFIT_ASSIGN in Figure 3 needs to be changed when considering the frequencies of occurrence.

Table 3. (**a**) Frequencies of occurrence (database size = 22,000). (**b**) Expected profit considering lifts and frequencies.

	A	B	C	D	E	F
A	–	1000	1000	1000	1000	4000
B	1000	–	1000	4000	1000	1000
C	1000	1000	–	1000	3000	1000
D	1000	4000	1000	–	1000	1000
E	-	1000	1000	1000	1000	4000
F	1000	-	1000	4000	1000	1000
	A	**B**	**C**	**D**	**E**	**F**
A	–	16,000	0	26,000	0	48,000
B	16,000	–	11,000	56,000	0	39,000
C	0	11,000	–	26,000	45,000	0
D	26,000	56,000	26,000	–	0	13,000
E	–	16,000	0	26,000	0	48,000
F	16,000	–	11,000	56,000	0	39,000

For example, according to Table 3b, assigning item B next to item D can promote higher sales than other assignments. In the same way, item C should be located next to item E. The assignment denoted as PROF_FRE_ASSIGN in Figure 2b is an updated version from PROFIT_ASSIGN, exchanging B and E with C and F, respectively. Then, PROF_FRE_ASSIGN is a better assignment than PROFIT_ASSIGN when considering increased profit as well as frequency.
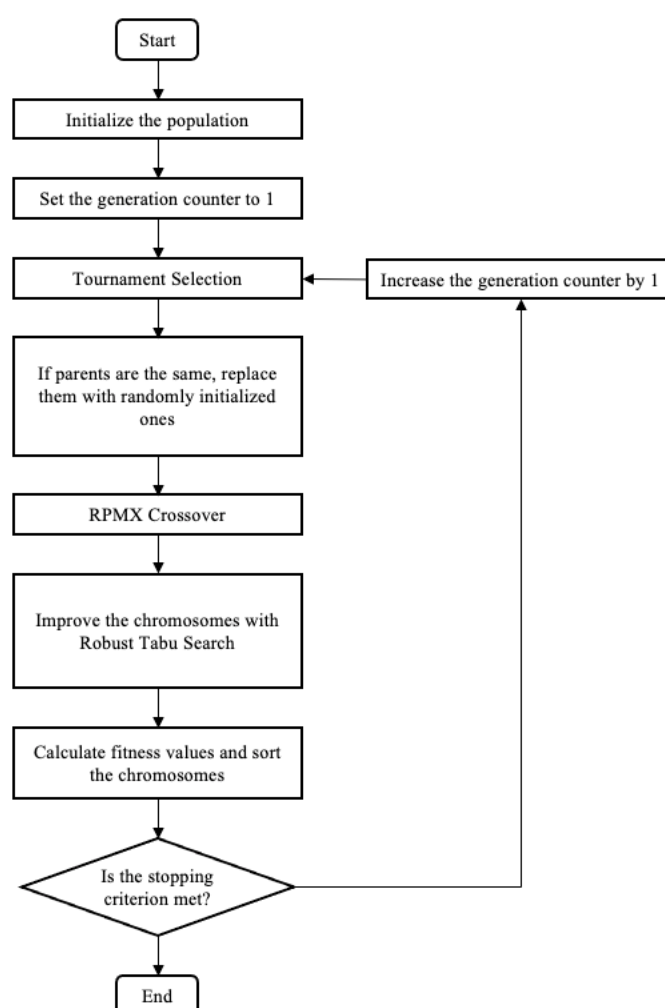
Figure 3. Flowchart of the proposed hybrid genetic algorithm (HGA).

3.2. Weighted Item Assignment Model

Section 3.1 discussed the influences of profit and frequency on item assignment plans. In this section, we propose a weighted item assignment model that considers both profit and frequency as in Equation (3), which is a modified model of Equation (2). Interestingly, this optimization problem becomes a special case of the quadratic assignment problem (QAP). In Equation (3), the weights of profit and frequency are multiplied:

Maximize

$$\sum_{i \neq j, k \neq l} w_{profit(I_i, I_j)} \times w_{frequcncy(I_i, I_j)} \times lift(I_i \rightarrow I_j) \times x_{ik} \times x_{jl} \times y_{kl} \qquad (3)$$

Subject to

$$\sum_{i=1}^{n} x_{ik} = 1, k = 1, \dots, n,$$

$$\sum_{k=1}^{n} x_{ik} = 1, i = 1, \dots, n,$$

$$x_{ik} = \begin{cases} 1, & \text{if item } I_i \text{ is located at site } S_k \\ 0, & \text{otherwise.} \end{cases}$$

Since the scales of profit and frequency are different depending on the problems, this study used relative profit and relative frequency as weights of the model. Relative profits were calculated by dividing the profit by the lowest increased profit among items. For instance, since the lowest profit in Table 2a is $10, the relative profits are calculated by dividing all profits by $10. Table 4 shows the relative profits obtained from Table 2a.

Table 4. Relative profit increased by cross-selling items.

	A	B	C	D	E	F
A	–	1	1	2	1	1
B	1	–	1	1	1	3
C	1	1	–	1	1	1
D	21	1	1	–	1	1
E	1	1	1	1	–	1
F	1	3	1	1	1	–

Likewise, relative frequencies are calculated by dividing the frequencies by the lowest frequency among all of them. By dividing all the frequencies in Table 3a by the lowest frequency, 1000, the relative frequencies can be obtained as in Table 5a. Then, we can rewrite the weights in Table 3b with those in Table 5b.

Table 5. (**a**) Relative frequencies of occurrence. (**b**) Expected relative profit considering lifts and frequencies.

	A	B	C	D	E	F
A	–	1	1	1	1	4
B	1	–	1	4	1	1
C	1	1	–	1	3	1
D	1	4	1	–	1	1
E	1	1	3	1	–	1
F	4	1	1	1	1	–
	A	**B**	**C**	**D**	**E**	**F**
A	–	1.6	0	2.6	0	4.8
B	1.6	–	1.1	5.6	0	3.9
C	0	1.1	–	2.6	4.5	0
D	2.6	5.6	2.6	–	0	1.3
E	0	0	4.5	0	–	1.3
F	4.8	3.9	0	1.3	1.3	–

According to the weighted item assignment, the objective values of Equation (3) for BASE_ASSIGN, PROFIT_ASSIGN, and PRO_FRE_ASSIGN are 15, 18.4, and 21.4, respectively, as shown in Table 6. This result verifies the superiority of PRO_FRE_ASSIGN among the three assignments when considering both profits and frequencies. Since the weighted item assignment model considers the three factors of association, profit, and frequency among the items, the model can offer insight to decision-makers for finding the most effective assignment of items for cross-selling.

Table 6. *Relative profit* increased by cross-selling items.

	BASE_ASSIGN	PROFIT_ASSIGN	PROF_FRE_ASSIGN
Objective value according to Equation (3)	1.6 + 2.6 + 1.1 + 3.9 + 4.5 + 1.3 = 15	2.6 + 4.8 + 3.9 + 2.6 + 4.5 = 18.4	2.6 + 4.8 + 5.6 + 3.9 + 4.5 = 21.4

4. Hybrid Genetic Algorithm to Search for Effective Item Assignments

To identify effective assignments in the weighted item assignment problem, we propose a hybrid genetic algorithm (HGA) that focuses on a rectangular layout of sites. We describe the whole process of HGA in Figure 3. We first define a chromosome appropriately to represent the proposed item assignment model. Then, we calculate the fitness of each chromosome using the objective function in Equation (3). For the first generation, we

generate each chromosome up to a given population size by randomly assigning items to a rectangular layout. With chromosomes in the population, we create new chromosomes through the process of selection, crossover, applying a robust tabu search, until the generation counter reaches the given maximum number of generations. In the selection procedure, if the genes of selected parents are the same, this will lead to producing the same children. To avoid this problem, we replace both parents with randomly initialized chromosomes when parents are the same. The abovementioned setting obviates the need of mutation operation. Our observation in the proposed algorithm is that the exclusion of mutation induces a simple parameter setting without compromising the performance.

4.1. Equivalence to the TSP Problem

First, we show that a special case of the proposed item assignment problem is equivalent to a symmetric traveling salesman problem when the layout is circularly linked. We prove this by transforming the symmetric traveling salesman problem into an item assignment problem. Let $I = \{I_1, I_2, \ldots, I_n\}$ be the set of cities and $S = \{S_1, S_2, \ldots, S_n\}$ be the set of sites structured in a circularly linked form. In other words, the following site configuration in Figure 4a is considered.
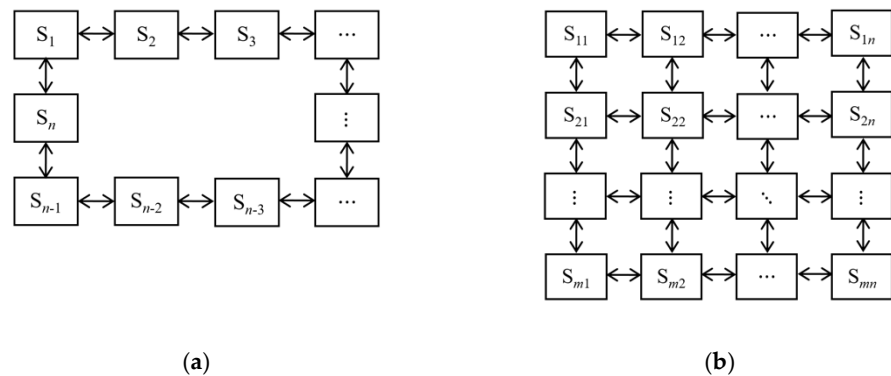

(**a**)   (**b**)

Figure 4. (**a**) Circularly linked layout. (**b**) Rectangular layout form.

Now, we allocate n cities to n sites, with one city at each site, which represents a traveling sequence of the n cities. For example, for n = 4, the allocation of I_2 at S_1, I_3 at S_2, I_1 at S_3, and I_4 at S_4 represents a traveling sequence, $I_2 \rightarrow I_3 \rightarrow I_1 \rightarrow I_4 \rightarrow I_2$. It is straightforward that a traveling sequence of the n cities is mapped to an allocation of the n cities to form the circularly linked layout, and the mapping is one-to-one. Now, we formulate a symmetric traveling salesman problem given a circularly linked layout **S**. Suppose that c_{ij} is the distance between city i and city j; $x_{i,j}$ is 1 if city I_i is located at site S_j and 0 otherwise, and y_{ij} is 1 if site S_i is adjacent to S_j and 0 otherwise. For a given layout **S**, y_{ij} is given, and x_{ij} is a decision variable. Since $c_{ij} \times x_{i,k} \times x_{j,k+1}$ is a traveling cost from the k-th visiting city (i) to the (k + 1)-th visiting city (j), the total cost to minimize is given by

$$\sum_{i=1} \sum_{j \neq i} c_{ij} \times x_{i,1} \times x_{j,2} + c_{ij} \times x_{i,2} \times x_{j,3} + \cdots + c_{ij} \times x_{i,n-1} \times x_{j,n} + c_{ji} \times x_{j,n} \times x_{i,1} = \sum_{i \neq j, k \neq l} c_{ij} \times x_{i,k} \times x_{j,l} \times y_{k,l} \quad (4)$$

with the constraints that each city i should be visited once, $\sum_{k=1}^{n} x_{i,k} = 1$, and each site k should be taken once, $\sum_{i=1}^{n} x_{i,k} = 1$. For example, a traveling sequence, $I_2 \rightarrow I_3 \rightarrow I_1 \rightarrow I_4 \rightarrow I_2$, will have its objective value calculated by $c_{23} \times x_{2,1} \times x_{3,2} + c_{31} \times x_{3,2} \times x_{1,3} + c_{14} \times x_{1,3} \times x_{4,4} + c_{42} \times x_{4,4} \times x_{2,1} = c_{23} + c_{31} + c_{14} + c_{41}$ with only $x_{2,1} = x_{3,2} = x_{1,3} = x_{4,4} = 1$ among the decision variables. Thus, we rewrite the symmetric TSP as the following minimization programming:

$$\text{Minimize} \qquad \sum_{i \neq j, k \neq l} c_{ij} \times x_{ik} \times x_{jl} \times y_{kl} \tag{5}$$

$$\text{Subject to} \qquad \sum_{i=1}^{n} x_{ik} = 1, k = 1, \dots, n,$$

$$\sum_{k=1}^{n} x_{ik} = 1, i = 1, \dots, n,$$

$$x_{ik} = \begin{cases} 1, & \text{if item } I_i \text{ is located at site } S_k \\ 0, & \text{otherwise.} \end{cases}$$

Equation (5) has the same structure as Equation (3) except that TSP is a minimization problem. Thus, we conclude that the proposed model in Equation (5) is a general case of TSP.

4.2. Definition of a Rectangular Chromosome

We design a rectangular-shape chromosome to apply a genetic algorithm on the item assignment problem. In the rectangular chromosome, an assignment of items is represented in a matrix format. Suppose, an item set I is given by $I = \{I_{11}, I_{12}, \dots, I_{mn}\}$ and a site set $S = \{S_{11}, S_{12}, \dots, S_{mn}\}$ is given with a rectangular layout with m rows and n columns, as shown in Figure 4b.

When items $I_{11}, I_{12}, \dots, I_{mn}$ are assigned to sites $S_{11}, S_{12}, \dots, S_{mn}$, we represent them with the following rectangular chromosome $r = [r_{ij}]$:

$$r = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m1} & \cdots & r_{mn} \end{bmatrix},$$

where r_{ij} indicates an item assigned at the site corresponding to site s_ij. For example, we represent the item assignment in Figure 2(b) as a rectangular chromosome with numeric encoding $[^{4}_{2}{^{1}_{6}}{^{5}_{3}}]$.

In this representation, the fitness value of the chromosome, which is the total possibility of cross-selling, can be calculated using Equation (3) as follows:

$$\sum_{i=1}^{m-1} \sum_{j=1}^{n} w_{profit(r_{ij}, r_{(i+1)j})} \times w_{frequcncy(r_{ij}, r_{(i+1)j})} \times lift(r_{ij} \rightarrow r_{(i+1)j}) + $$
$$\sum_{i=1}^{m} \sum_{j=1}^{n-1} w_{profit(r_{ij}, r_{i(j+1)})} \times w_{frequcncy(r_{ij}, r_{i(j+1)})} \times lift(r_{ij} \rightarrow r_{i(j+1)}). \tag{6}$$

4.3. Robust Tabu Search

To cope with the complexity of the proposed model with a rectangular layout, we adopt a robust tabu search (RTS) [23,25,26] as a local optimization step. Tabu search starts with approaching a local minimum and records recent moves in a tabu list. The tabu list forms adaptive memory to explore better solutions, and the size of tabu list decides a degree of diversification and intensification. While a large size of tabu list encourages the algorithm to avoid stagnation, a smaller size of tabu list encourages to produce local best solutions to be located around. Tabu search also uses aspiration criteria to allow for exceptions from the tabu list whenever such moves lead to a better result. Taillard's RTS makes use of a short-term memory and multiple levels of aspiration criteria. Short-term memory prevents the algorithm from revisiting recently visited solutions, which can produce high-quality results. However, long-term memory for diversification drives the search to promising solutions. RTS utilizes tabu list sizes, aspiration, and the number of iterations as parameters. In HGA, we set tabu list sizes to between $[0.9 \times n, 1.1 \times n]$ (where n is the problem size) and aspiration to $10n^2$, which are large enough to explore wide search space by allowing to escape from local optima. Often a larger number of iterations produce better solutions, and we set the number of iterations to 2000. Since the proposed method, being evolutionary, reuses previous found solutions, it can gradually approach better local optima as a generation increases. Algorithm 1 provides the details of the adopted RTS algorithm.

Algorithm 1: Robust Tabu Search Algorithm

1: **authorized:** if a move is not tabu, it is authorized.

2: **aspired:** Allow tabu moves if they are decided to be interesting.

3: **tabu_list:** A list to forbid reverse move.

4: **Function** RobustTabuSearch(bestPerm, flow, dist, minSize, maxSize, aspiration, numIters)

5:     n = size of bestPerm;

6:     bestCost = fitness of bestPerm;

7:     curCost = bestCost;

8:     curPerm = bestPerm;

9:     **for** (I = 0; I < n; I ++) **do**

10:         **for** (j = 0; j < n; j ++) **do**

11:             tabu_list[i][j] = -(n*I + j);

12:             **if** (I < j) **then**

13:                 delta[i][j] = compute_delta(curPerm, i, j);

14:             **end if**

15:         **end for**

16:     **end for**

17:     **for** (iter = 0; iter < numIters; iter ++) **do**

18:         i_retained = infinite;

19:         min_delta = infinite;

20:         already_inspired = false;

21:         **for** (I = 0; I < n−1; I ++) **do**

22:             **for** (j = I + 1; j < n; j ++) **do**

23:                 authorized = (tabu_list[i][j] < iter) || (tabu_list[j][curPerm[i]] < iter);

24:                 aspired = (tabu_list[i][p[j]] < iter-aspiration) ||

25:                  (tabu_list[j][p[i]] < iter-aspiration) ||

26:                  (curCost + delta[i][j] < bestCost);

27:                 **if** ((aspired &&!already_aspired) ||

28:                  (aspired && already_aspired && (delta[i][j] < min_delta)) ||

29:                  (!aspired &&!already_aspired && (delta[i][j] < min_delta) && authorized)) **then**

30:                     i_retained = i;

31:                     j_retained = j;

32:                     min_delta = delta[i][j];

33:                     **if** (aspired) **then**

34:                         already_aspired = true;

35:                     **end if**

36:                 **end if**

37:             **end for**

38:         **end for**

39:         **if** (i_retained! = infinite) **then**

40:             swap(curPerm[i_retained], curPerm[j_retained]);

41:             curCost = curCost + delta[i_retained][j_retained];

42:             tabu_list[i_retained][curPerm[j_retained]] = iter + unif(minSize, maxSize);

43:             tabu_list[j_retained][curPerm[i_retained]] = iter + unif(minSize, maxSize);

44:             **if** (curCost < bestCost) **then**

45:                 bestCost = curCost;

46:                 bestPerm = curPerm;

47:             **end if**

48:         **end if**

49:         **for** (i = 0; i < n−1; i++) **do**

50:             **for** (j = i + 1; j < n; j++) **do**

51:                 **if** (i! = i_retained && i! = j_retained && j! = i_retained && j! = j_retained) **then**

52:                     delta[i][j] = compute_delta_part(curPerm, delta, i, j, i_retained, j_retained);

```
53:                          else
54:                              delta[i][j] = compute_delta(curPerm, i, j);
55:                          end if
56:                      end for
57:              end for
58:      end for
```

4.4. Rectangular Partially Matched Crossover

In the proposed genetic algorithm, we use tournament selection [32] to determine parents. After selecting chromosomes up to a tournament size, which is set to three in this study, from a population at random, we choose the best and the second-best chromosomes as the parents based on their objective values. In addition, we propose rectangular partially matched crossover (RPMX), which effectively adapts a partially matched crossover (PMX) operator to the item assignment model. PMX uses a vector-format chromosome and randomly finds two cut points from parent chromosomes, performing mapping and exchanging to produce a feasible solution while RPMX uses a matrix-format chromosome. RPMX selects two cut points each from both row and column sides to choose a preserved area in a rectangular chromosome. As shown in Figure 5, e.g., the first cut point from the rows is between the second and third rows, and the second one is the end of the rows. Likewise, the first cut point from the columns is between the first and second columns, and the second one is between the fourth and fifth columns. Let us denote the set of selected genes as u_1 and u_2, respectively. For example, u_1 is {1,4,5,7,8,9,10,12,18}, colored orange in Figure 5. On each parent p_i, we map u_1 to keep the offspring o_i intact, and the rest of the genes for each parent are copied onto the other offspring after exchanging them to produce feasible solutions. In the process, we obtain conflicted genes in parent p_1 as $u_2 - u_1 \cap u_2$ and those in parent p_2 as $u_1 - u_1 \cap u_2$. For example, in Figure 5, $u_1 \cap u_2$ is {1,5,8,10,18}, and the conflicted genes in parent p_1 is $u_2 - \{1,5,8,10,18\}$, colored in white in the first row of the figure. Then, we obtain unconflicted genes as $p_1 - u_1 \cup u_2$ and $p_2 - u_1 \cup u_2$, respectively, in each parent, exchanging them to construct offsprings. For example, the unconflicted genes are highlighted in blue in Figure 5, i.e., the blue-colored genes on each parent are copied onto the other offspring intact since they have no conflict to produce feasible solution, while the white-colored genes are copied after exchanging procedure, and the orange-colored genes are kept intact as they were in each parent.
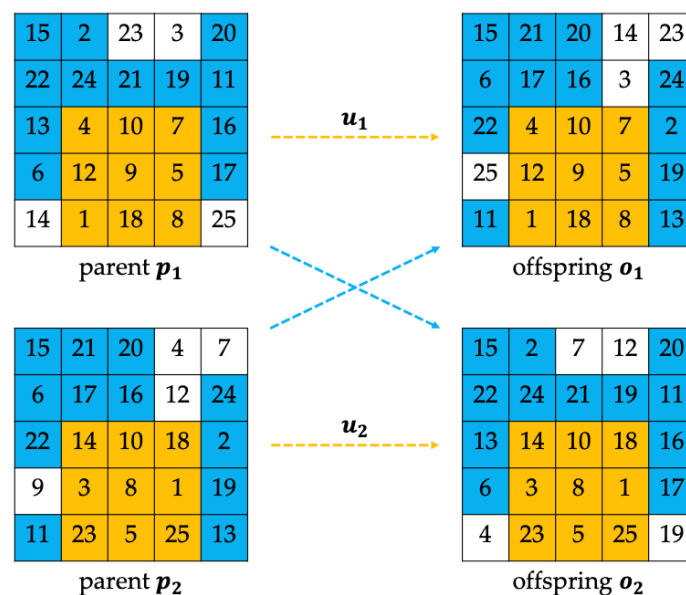

Figure 5. Rectangular partially matched crossover (RPMX) for weighted item assignment in rectangular layouts.

5. Experiments

In this section, we demonstrate the application of the item assignment models based on association and the effectiveness of the proposed HGA with QAPLIB instances and synthetic problems designed for item assignment situations. We compare the method with a few state-of-the-art algorithms. We carry out experiments in the Java language on an Intel Core 3.6 GHz PC, and the implemented codes and datasets are available via the following GitHub url: https://github.com/bzantium/HGA-RPMX.

5.1. Tuning the Parameters of HGA

The time performance of the proposed algorithm varies depending on two main parameters: population size and maximum generation number. Besides, execution time linearly increases in proportion to the maximum generation number since it is a stopping criterion in the algorithm. According to the practice in the QAP literature, however, it is common to use maximum time limit for fair evaluation of the performance. Thus, to properly tune parameters, we set maximum generation number to infinite, fixing maximum time limit to 30 min. In Figure 6, for the problem of tai60a, we provide boxplots of 30 repeated percentage deviations from the published best-known solution with increasing population size ranging from 10 to 100. The reason we choose tai60a for tuning parameters is that it is a fairly hard problem of a moderate size.
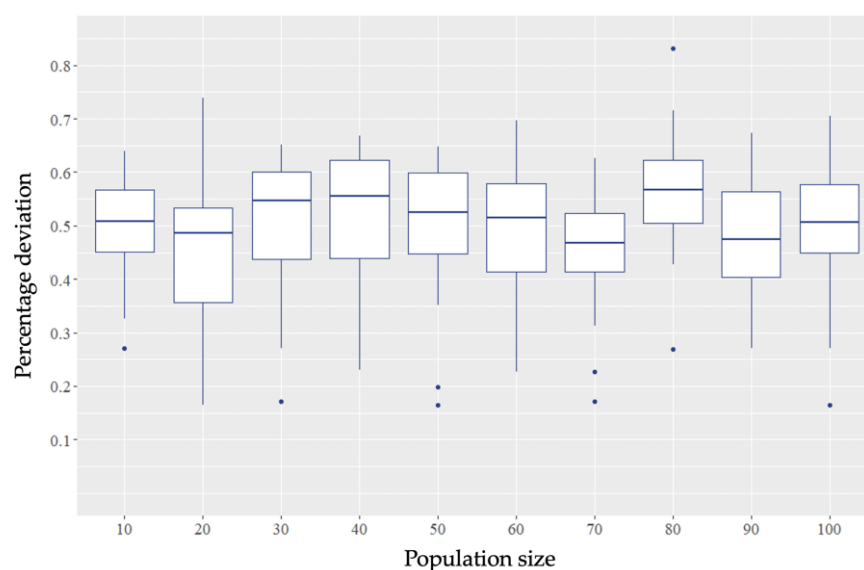

Figure 6. Boxplot of percentage deviations with increasing population size.

We observe performance is not substantially different along with population size, meaning that population size has little effect on the performance of the proposed algorithm. Therefore, we simply decide to use population size 20, which has the minimum sample average under the infinite maximum-generation number. Instead, we use a 2-h maximum time limit for a practical purpose to test the proposed HGA on all the following comparative experiments. Settings of parameters are summarized in Table 7.

Table 7. Parameter setting for the proposed hybrid genetic algorithm (HGA).

Parameters	Value
Population size	20
Maximum generation number	Infinite
Maximum time limit	2 h
Tabu list sizes	$[0.9n, 1.1n]$
Aspiration	$10 \times n^2$

Number of iterations	2000

5.2. Experimental Results and Comparisons on the QAPLIB

QAPLIB is the most popular collection of instances for QAP to evaluate performances of a new algorithm by comparing with other methods. In this part of experiments, we compare our proposed HGA with TLBO-RTS and ABC-QAP in terms of averaged percentage deviation (APD) from the best-known solution (BKS) value of the problem instances given by QAPLIB and averaged execution time (denoted by "min." in Table 8) to reach the reported result. We also take into account the number of reaching an optimal or BKS over 10 repeated executions. Since small and easy instances from the collection are easily and fast solved with no substantial differences between state-of-the-art algorithms, we present the results of challenging instances. Table 8 shows comparative results on the selected 19 instances. In this comparison, 255 number of processors were used in parallel for ABC-QAP algorithm. The proposed HGA outperformed the tested methods in the 15 problems, and it was both the most accurate and the fastest in terms of the average time in the 7 problems.

Table 8. Comparison of the teaching-learning-based optimization using robust tabu search (TLBO-RTS), RTS-integrated artificial bee colony (ABC) optimization for QAP (ABC-QAP), and HGA on synthetic data.

Problem	BKS	TLBO-RTS		ABC-QAP		Proposed HGA	
		APD	Min.	APD	Min.	APD	Min.
tai40a	3,139,370	0.440(0)	2.3	0.000(10)	4.0	0.000(10)	41.4
tai50a	4,938,796	0.961(0)	4.4	0.312(0)	19.8	0.241(1)	58.5
tai60a	7,205,962	1.098(0)	7.6	0.449(0)	34.0	* 0.335(0)	71.8
tai80a	13,499,184	1.124(0)	18.8	0.817(0)	83.5	* 0.603(0)	52.9
tai100a	21,052,466	1.091(0)	36.1	0.614(0)	173.0	0.573(0)	47.6
tai50b	458,821,517	0.000(10)	4.2	0.000(10)	1.6	0.000(10)	0.8
tai60b	608,215,054	0.000(10)	7.2	0.000(10)	0.7	0.000(10)	0.8
tai80b	818,415,043	0.005(0)	17.5	0.000(10)	8.0	0.000(10)	10.4
tai100b	1,185,996,137	0.027(2)	35.7	0.000(10)	164.7	0.024(8)	18.4
sko72	48,498	0.024(0)	13.2	0.000(10)	7.0	0.000(10)	3.3
sko81	66,256	–	–	0.000(10)	15.2	0.000(10)	15.5
sko90	90,998	0.049(0)	25.2	0.000(10)	14.7	0.000(10)	12.9
sko100a	115,534	0.064(0)	35.7	0.008(0)	170.6	* 0.003(8)	28.1
sko100b	152,002	0.005(0)	35.7	0.000(10)	127.8	0.000(10)	38.6
sko100c	147,862	0.019(0)	35.7	0.000(10)	48.3	0.000(7)	41.7
sko100d	149,576	0.031(0)	35.7	0.000(10)	45.9	0.000(10)	28.4
sko100e	149,150	0.023(0)	35.7	0.000(10)	42.0	0.000(10)	23.1
sko100f	149,036	0.047(0)	35.7	0.000(10)	28.3	0.002(7)	22.0
wil100	273,038	0.010(0)	36.1	0.000(10)	10.4	0.001(6)	31.5
Average		0.279	23.5	0.116	52.6	0.094	28.8

Column "APD" stands for the average percentage deviation from the best-found solution from all trials, and column "Min." average time to reach the found fitness values in minutes. "*" means the HGA result is statistically significant with significance level of 0.01 compared to ABC-QAP.

From Table 8, we observe that HGA shows competitive and better performance on all instance except only a few, finding the best-known solution for 16 out of 19 instances. Especially, for harder instances, tai40a, tai50a, tai60a, tai80a, and tai100a, it consistently outperforms the other algorithms.

5.3. Synthetic Experimental Design

We also tested the hybrid genetic algorithm using synthetic data including lifts, relative profits, and relative frequencies of items. Table 9 shows the range of the parameters set for the synthetic data. Lifts between the two items are randomly generated as a value within the range of 1 to 1.4 with probability of 0.9 otherwise 0. Relative profits and relative frequencies between the two items are randomly generated within the range of 1 to 4.

Table 9. Parameters used generate synthetic data.

Parameters	Range
Lift	1 to 1.4 or 0
Relative profit	1 to 4
Relative frequency	1 to 4

The synthetic databases shown in Table 10 are used in these experiments. The layouts of the sites in Table 10 are rectangular layouts of sites in which all the items are assigned. The synthetic databases consist of the table of lifts, table of relative profits, and table of relative frequencies. For example, the table of lifts for DB4 in Table 10 is a 225×225 matrix with values from 1 to 1.4 or zero, and tables of relative profits and relative frequencies are 225×225 matrices with values from 1 to 4.

Table 10. Synthetic databases (boldfaced) and the layout of sites.

	DB1	**DB2**	**DB3**	**DB4**
Number of items	50	100	150	225
Layout of sites	10×5	10×10	10×15	15×15

5.4. Comparison of Assignment Models

In order to verify the superiority of the weighted item assignment model, three types of assignment models were compared, as shown in Table 11. Each assignment was optimized by the proposed algorithm with only one maximum generation number. The first assignment, denoted as (A), was updated from the initial assignment when considering only lifts. The fitness values in column (A) were calculated according to Equation (2). The second assignment was denoted as (B) and was updated from the initial assignment when considering both lifts and relative profits. Denoted as (C), the third assignment was updated from the initial assignment when considering lifts, relative profits, and relative frequencies. The numbers in column (C) are the fitness values of Equation (3).

Table 11 shows that the assignments denoted as (C) are better solutions when compared to the other assignments. For example, according to the experimental results of DB4, if retailers assign items to sites based on (A), which only considers lifts, they will lose around 94% of the chance of gaining profits compared to assignment (C). In addition, placing items at sites based on (C) increases the chance of sales by 27% more than assignment (B), which only accounts for lifts and relative profits. The experimental results in Table 11 demonstrated that the weighted item assignment model can mimic real situations in retail more accurately than the previous model that considered only lifts and even shows better performances as the problem size increases.

Table 11. Comparison of three assignments considering different factors: (A) lifts; (B) lifts and relative profits; and (C) lifts, relative profits, and relative frequencies.

DB	(A)	(B)	(C)	(C)/(A)	(C)/(B)
DB1	1328.2	1856.8	2256.2	1.70	1.22
DB2	2767.8	4035.6	4967.8	1.79	1.23
DB3	4146.8	6204.4	7812.4	1.88	1.26

DB4	6253.8	9571.4	12,121.0	1.94	1.27

5.5. Comparison of HGA Using Different Crossover

We performed experiments to compare the proposed crossover, RPMX, in HGA with several existing crossover operators, position-based crossover (POS) [33] and uniform crossover (UX) [34]. Note that position-based crossover, possessing the same structure as RPMX, is different on how to choose exchanging genes. We compared the performance of crossover methods, RPMX, POS, and UX, using DB1, DB2, DB3, and DB4 in the item allocation situation. We present averaged fitness value and average time for each synthetic data in Table 12. We note that RPMX consistently outperforms the other crossover methods in terms of averaged fitness within comparative computational time.

Table 12. Computational results of the proposed HGA integrating different crossover operators.

Instance	POS		UX		RPMX	
	Fitness	**Min.**	**Fitness**	**Min.**	**Fitness**	**Min.**
DB1	2305.6	68.3	2303.2	53.6	* 2306.0	16.2
DB2	5033.0	29.6	5029.4	57.0	* 5105.8	49.7
DB3	7891.0	38.8	7894.6	61.8	* 8040.4	69.8
DB4	12,228.8	52.6	12,226.8	54.5	* 12,464.8	78.7

"Fitness" and "Min." represent averaged fitness and average time to reach the found fitness values in minutes, respectively. "*" means that rectangular partially matched crossover (RPMX) is statistically significant with significance level of 0.01 in comparison with other crossover methods by two-way ANOVA for each instance.

5.6. Experimental Results and Comparison on the Synthetic Data

We evaluate the performance of the HGA on the synthetic data. Compared with TLBO-RTS and ABC-QAP, the proposed method HGA consistently outperforms the state-of-the-art algorithms in all databases. Table 13 shows the best-found solution (BFS) from trials, the averaged fitness value, averaged percentage deviation from BFS, and the averaged time to reach the reported fitness over 10 repeated executions for each database. All best-found solutions are derived by the proposed HGA.

Table 13. Comparison of the TLBO-RTS, ABC-QAP, and HGA on synthetic data.

Instance	BFS	TLBO-RTS			ABC-QAP			Proposed HGA		
		Fitness	**APD**	**Min.**	**Fitness**	**APD**	**Min.**	**Fitness**	**APD**	**Min.**
DB1	2306	2281.6	1.058	4.5	2295.8	0.442	45.3	* 2306.0	0.000	16.2
DB2	5162	5027.0	2.615	29.6	5037.8	2.406	55.9	* 5105.8	1.089	49.7
DB3	8086	7898.8	2.315	29.0	7925.2	1.989	76.5	* 8040.4	0.564	69.8
DB4	12,554	12,301.6	2.011	39.3	12,310.0	1.944	57.5	* 12,464.8	0.711	78.7

Column "APD" stands for the average percentage deviation from the best-found solution from all trials. Columns "fitness" and "min." represent averaged fitness and average time to reach the found fitness values in minutes, respectively. "*" means that rectangular partially matched crossover (RPMX) is statistically significant with significance level of 0.01 in comparison with ABC-QAP by paired *t* test for each instance.

We observe that the proposed HGA method consistently outperformed the other methods in all the databases in terms of fitness and APD. As can be seen above, HGA produced better solutions (higher fitness values) with substantially low deviation while keeping up competitive computational time for the tested problems in all instances. These results demonstrate that our proposed algorithm is effective, in particular, for weighted item assignment situations. Finally, we show averaged fitness values with 50 repetitions according to the maximum time limit from 30 to 150 min with step size 30 for DB2 in

Figure 7 to demonstrate the convergence of the proposed method. The values increase as the maximum time limit increases with a trend of saturation.
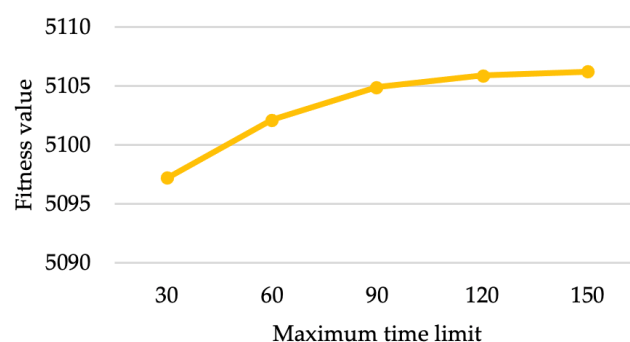

Figure 7. Averaged fitness values with 50 repetitions along with maximum time limit for DB2.

6. Conclusions

This work proposes an item assignment model based on association rule mining to achieve a more realistic assignment model for the retail industry. Strong associations between the items can be used by retailers to position-correlated items close together for cross-selling. However, the strength of the association and the information regarding the profit and frequency of purchasing items are crucial factors for determining effective item assignments. This study suggests a profitable item assignment model combining association with profit and frequency and also proposes a hybrid genetic algorithm to find the most effective assignment. Lift was used to measure the strength of the associations. The absolute profits that increased due to cross-selling items were transformed into relative profits to mitigate gaps in the scale. The frequencies of cross-selling items were also transformed to their relative values. To test the effectiveness of the proposed weighted item assignment model, we tested three assignment types. The first assignment included only information for lifts. The second assignment was based on both lifts and profits. The third assignment considered lifts, profits, and frequencies. The experimental results showed that item assignment based on the third assignment increased the chance of gaining profits more than the other assignments. The results verified the superiority of the profitable item assignment model combining association, profit, and frequency. In addition, we performed two more experiments to compare the proposed hybrid genetic algorithm with state-of-the-art algorithms and the proposed crossover operator with other crossover operators. The tests showed that the proposed hybrid genetic algorithm, HGA, consistently provides a better solution in a reasonable computation time for the tested databases compared to the other methods. We showed that the proposed model, a quadratic assignment problem, is a general case of traveling sales problems. The proposed genetic algorithm is equipped with a specially designed gene, suitable for a layout representation, and its associated rectangular partially matched crossover. The weakness of the proposed method is a lack of theoretical analysis, which is inherent in metaheuristic algorithms. As a future research direction, research considering qualitative factors such as shopping path and layout type should be conducted in practical retail environments to provide a more realistic model for retail purposes. We also envision extending it to other assignment problems such as multi-mode multi-user traffic assignment problem in urban transportation networks.

References

1. Ahn, K.I. Effective product assignment based on association rule mining in retail. *Expert Syst. Appl.* **2012**, *39*, 12551–12556, doi:10.1016/j.eswa.2012.04.086.
2. Yang, Z.; Tang, W.H.; Shintemirov, A.; Wu, Q.H. Association Rule Mining-Based Dissolved Gas Analysis for Fault Diagnosis of Power Transformers. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2009**, *39*, 597–610, doi:10.1109/tsmcc.2009.2021989.
3. Huang, J.; Ji, S.; Xie, H.; Wang, X. Research on cross-selling in the express delivery enterprise based on data mining. In Proceedings of the 2008 International Conference on Service Systems and Service Management, Melbourne, Australia, 30 June–3 July 2008; pp. 1–6.
4. Jiao, J.; Zhang, Y. Product portfolio identification based on association rule mining. *Comput. Des.* **2005**, *37*, 149–172, doi:10.1016/j.cad.2004.05.006.
5. Cho, Y.H.; Kim, J.K.; Kim, S.H. A personalized recommender system based on web usage mining and decision tree induction. *Expert Syst. Appl.* **2002**, *23*, 329–342, doi:10.1016/s0957-4174(02)00052-0.
6. Kang, X.; Porter, C.S.; Bohemia, E. Using the fuzzy weighted association rule mining approach to develop a customer satisfaction product form. *J. Intell. Fuzzy Syst.* **2020**, *38*, 4343–4357.
7. Ramasamy, S.; Nirmala, K. Disease prediction in data mining using association rule mining and keyword based clustering algorithms. *Int. J. Comput. Appl.* **2017**, *42*, 1–8, doi:10.1080/1206212x.2017.1396415.
8. Shao, Y.; Liu, B.; Wang, S.; Li, G. Software defect prediction based on correlation weighted class association rule mining. *Knowl. Based Syst.* **2020**, *196*, 5742, doi:10.1016/j.knosys.2020.105742.
9. Tuysuzoglu, G.; Birant, D. Air Pollution Monitoring in Intelligent Cities Using Weighted Association Rule Mining. In *Advances in Civil and Industrial Engineering*; IGI Global: Hershey, PA, USA, 2021; pp. 171–197.
10. Han, J.; Pei, J.; Kamber, M. *Data Mininig: Concepts and Techniques*; Elsevier: Amsterdam, The Netherlands, 2011.
11. Brijs, T.; Swinnen, G.; Vanhoof, K.; Wets, G. Building an Association Rules Framework to Improve Product Assortment Decisions. *Data Min. Knowl. Discov.* **2004**, *8*, 7–23, doi:10.1023/b:dami.0000005256.79013.69.
12. Chen, M.C. Configuration of cellular manufacturing systems using association rule induction. *Int. J. Prod. Res.* **2003**, *41*, 381–395, doi:10.1080/0020754021000024184.
13. Liu, C.; Yasuda, K.; Yin, Y.; Tanaka, K. Uncertain association rule mining algorithm for the cell formation problem in cellular manufacturing systems. *Int. J. Prod. Res.* **2008**, *47*, 667–685, doi:10.1080/00207540701501849.
14. Chen, M.-C.; Huang, C.L.; Chen, K.Y.; Wu, H.P. Aggregation of orders in distribution centers using data mining. *Expert Syst. Appl.* **2005**, *28*, 453–460, doi:10.1016/j.eswa.2004.12.006.
15. Ardakani, B.A.; Fathian, M.; Gholamian, M. A novel model for product bundling and direct marketing in e-commerce based on market segmentation. *Decis. Sci. Lett.* **2018**, *7*, 39–54, doi:10.5267/j.dsl.2017.4.005.
16. Cai, C.; Fu, A.; Cheng, C.; Kwong, W. Mining association rules with weighted items. In Proceedings of the IDEAS'98. International Database Engineering and Applications Symposium (Cat. No.98EX156), Cardiff, UK, 10 July 2002; pp. 68–77, doi:10.1109/ideas.1998.694360.
17. Tao, F.; Murtagh, F.; Farid, M.; Coenen, F. Weighted association rule mining using weighted support and significance framework. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, Washington, DC, USA, 24–27 August 2003; pp. 661–666.
18. Khan, M.S.; Muyeba, M.; Coenen, F. Weighted Association Rule Mining from Binary and Fuzzy Data. In *Computer Vision*; Springer International Publishing: New York, NY, USA, 2008; Volume 5077, pp. 200–212.
19. Pears, R.; Koh, Y.S.; Dobbie, G.; Yeap, W. Weighted association rule mining via a graph based connectivity model. *Inf. Sci.* **2013**, *218*, 61–84, doi:10.1016/j.ins.2012.07.001.
20. Lee, D.; Park, S.H.; Moon, S. Utility-based association rule mining: A marketing solution for cross-selling. *Expert Syst. Appl.* **2013**, *40*, 2715–2725, doi:10.1016/j.eswa.2012.11.021.
21. Heydari, M.; Yousefli, A. A new optimization model for market basket analysis with allocation considerations: A genetic algorithm solution approach. *Manag. Mark. Chall. Knowl. Soc.* **2017**, *12*, 1–11.
22. Awler, E.L. The quadratic assignment problem. *Manag. Sci.* **1963**, *9*, 586–599.
23. Taillard, E. Robust taboo search for the quadratic assignment problem. *Parallel Comput.* **1991**, *17*, 443–455, doi:10.1016/s0167-8191(05)80147-4.
24. Benlic, U.; Hao, J.K. Memetic search for the quadratic assignment problem. *Expert Syst. Appl.* **2015**, *42*, 584–595, doi:10.1016/j.eswa.2014.08.011.

25. Dokeroglu, T. Hybrid teaching–learning-based optimization algorithms for the Quadratic Assignment Problem. *Comput. Ind. Eng.* **2015**, *85*, 86–101, doi:10.1016/j.cie.2015.03.001.

26. Dokeroglu, T.; Sevinc, E.; Cosar, A. Artificial bee colony optimization for the quadratic assignment problem. *Appl. Soft Comput.* **2019**, *76*, 595–606, doi:10.1016/j.asoc.2019.01.001.

27. 27. Misevičius, A.; Verenė, D. A Hybrid Genetic-Hierarchical Algorithm for the Quadratic Assignment Problem. *Entropy* **2021**, *23*, 108.

28. Ma, J.; Xu, M.; Meng, Q.; Cheng, L. Ridesharing user equilibrium problem under OD-based surge pricing strategy. *Transp. Res. Part B Methodol.* **2020**, *134*, 1–24, doi:10.1016/j.trb.2020.02.001.

29. Dewi, S.K.; Utama, D.M. A New Hybrid Whale Optimization Algorithm for Green Vehicle Routing Problem. *Syst. Sci. Control. Eng.* **2021**, *9*, 61–72, doi:10.1080/21642583.2020.1863276.

30. Hanh, N.T.; Binh, H.T.T.; Hoai, N.X.; Palaniswami, M.S. An efficient genetic algorithm for maximizing area coverage in wireless sensor networks. *Inf. Sci.* **2019**, *488*, 58–75, doi:10.1016/j.ins.2019.02.059.

31. Yu, Y.; Li, J.; Li, Y.; Li, S.; Li, H.; Wang, W. Comparative Investigation of Phenomenological Modeling for Hysteresis Responses of Magnetorheological Elastomer Devices. *Int. J. Mol. Sci.* **2019**, *20*, 3216, doi:10.3390/ijms20133216.

32. Miller, B.L.; Goldberg, D.E. Genetic algorithms, tournament selection, and the effects of noise. *Complex Syst.* **1995**, *9*, 193–212.

33. Oliver, I.; Smith, D.; Holland, J.R. Study of permutation crossover operators on the traveling salesman problem. Genetic algorithms and their applications In Proceedings of the second International Conference on Genetic Algorithms, Massachusetts Institute of Technology, Cambridge, MA, USA; Hillsdale, NJ, USA, 28–31 July 1987.

34. Syswerda, G. A Study of Reproduction in Generational and Steady-State Genetic Algorithms. In *Foundations of Genetic Algorithms*; Elsevier BV: Amsterdam, The Netherlands, 1991; Volume 1, pp. 94–101.