Engineering Optimization
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/geno20

A bi-objective model for the retail spatial design problem

Haluk Yapicioglu [a] & Alice E. Smith [b]

[a] Department of Industrial Engineering, Anadolu University, Eskisehir, Turkey

[b] Department of Industrial and Systems Engineering, Auburn University, Auburn, Alabama, USA

Available online: 27 Feb 2012

To cite this article: Haluk Yapicioglu & Alice E. Smith (2012): A bi-objective model for the retail spatial design problem, Engineering Optimization, 44:3, 243-266

To link to this article: http://dx.doi.org/10.1080/0305215X.2011.647812

Engineering Optimization
Vol. 44, No. 3, March 2012, 243–266

A bi-objective model for the retail spatial design problem

Haluk Yapicioglu[a] and Alice E. Smith[b]*

[a]*Department of Industrial Engineering, Anadolu University, Eskisehir, Turkey;* [b]*Department of Industrial and Systems Engineering, Auburn University, Auburn, Alabama, USA*

(*Received 29 December 2010; final version received 28 November 2011*)

In this article, a model for the design of the block layout of a retail store is presented as a bi-objective optimization problem. The approach consists of placing departments in a racetrack configuration within the store subject to area and shape constraints, where the first objective function considers the area allocated to each department, contiguity of the departments to the aisle network and resulting revenue generated, and the second objective function considers adjacency requirements among departments. Owing to the complex nature of the model, two metaheuristic search methods, a multi-objective tabu search and the most well known multi-objective genetic algorithm are used separately to solve the problem. The performance of these two heuristics is evaluated and compared, with results suggesting that the multi-objective tabu search is a better choice because of its ability to exploit the neighbourhood structure of the model.

Keywords: multi-objective optimization; tabu search; genetic algorithm; NSGA-II; hybrid optimization; store design; racetrack aisle

1. Introduction

In its broadest definition, the facility layout problem is defined as locating a number of departments within a given area to optimize a performance metric. The two most widely used performance measures are minimizing the total travel distance among departments and maximizing adjacency among departments. Although there is a large body of literature addressing facility layout problems, articles addressing service facilities are very few and problem specific, such as Elshafei's hospital layout article (1977). In this article, the design of the block layout of retail stores in a bi-objective context is of interest. The most basic distinction between manufacturing facilities and retail facilities is that in the latter the moving objects inside the facility are mostly humans. Hence, traditional performance measures do not reflect the needs of the retail facility layout problem. When the facility layout literature is examined, there is a lack of studies regarding retail stores. Only a few articles address the retail store layout problem. These include the work by Peters *et al.* (2004), where they consider the grid layout for a retail store and propose a model for the problem, and earlier work by the authors (Yapicioglu and Smith forthcoming).

*Corresponding author. Email: smithae@auburn.edu

ISSN 0305-215X print/ISSN 1029-0273 online

http://dx.doi.org/10.1080/0305215X.2011.647812
http://www.tandfonline.com

The research objectives of this article are: (1) to develop a bi-objective model that captures the notion of revenue generation with respect to layout in retail stores; (2) to quantify the aisle area allocation; and (3) to design and develop appropriate solution procedures. The organization of the article is as follows. In Section 2, a brief literature survey on previous work on the space–revenue relationship in retail settings and the notion of impulse purchase is provided. These two fields of study are fundamental in creating one of the objective functions, the revenue. Section 3 introduces the mathematical model for the retail facility space allocation problem. In Section 4, the solution approaches are presented. To solve the model, a special solution representation and decoding/encoding mechanisms are developed and utilized in a tabu search and an elitist non-dominated sorting genetic algorithm framework. The article concludes with future research directions, provided in Section 5.

2. Background

In today's retailing world, most stores use one of three layout types: the grid, the free form and the racetrack (Levy and Weitz 2001). Among the three, a primary advantage of the racetrack layout is that it allows shoppers to move along a store's merchandise in a quick and comprehensive manner. To illustrate this, an example racetrack layout is provided in Figure 1.

Next, area revenue relationships, the notion of customer traffic density and impulse purchases are discussed.

2.1. *Shelf space allocation problems*

The relationship between the space allotted to merchandise and profit and/or revenue has drawn the attention of researchers for 50 years. Brown and Tucker (1961) articulated that the law of diminishing marginal returns applies to the space–revenue relationship in retail settings. Lee (1961) also asserted that the effect of increasing shelf space must be of diminishing returns. However, neither of these studies specified a functional form on the definition of the diminishing returns. The assumption of diminishing marginal returns was validated by the field studies of Frank and Massy (1970), Curhan (1973) and Dreze *et al.* (1994). Corstjens and Doyle (1981)

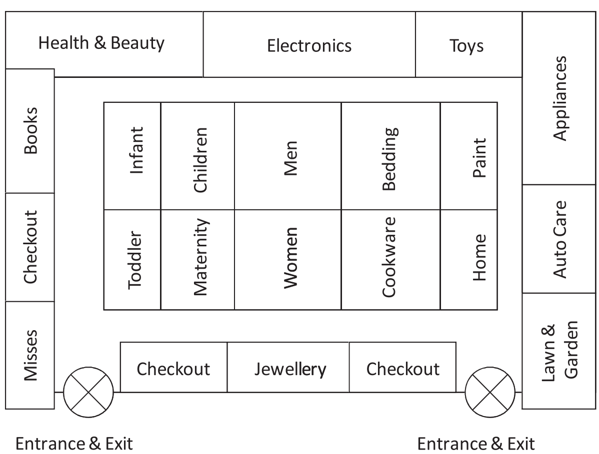

Figure 1. An example racetrack layout.

formulated the demand (q) for product i based on the space elasticity as follows:

$$q_i = r_i s_i^{\beta_i} \prod_{\substack{j=1 \\ j \neq i}}^{K} S_j^{\delta_{ij}}. \tag{1}$$

where β_i is space elasticity for product i, S_i is shelf space allocated to product i, δ_{ij} is cross-elasticity between product i and product j, and r_i is demand multiplier for product i.

Based on the definition above, Corstjens and Doyle also performed a field study which suggested that the space elasticities for different products fluctuate between 0.06 and 0.25 and cross-elasticities between −0.01 and −0.05. Considering the cross-elasticities of secondary importance, Irion *et al.* (2004) simplified the Corstjens and Doyle model by removing the cross-elasticities from the formulation. They formulated the demand function as:

$$q_i = r_i s_i^{\beta_i} \tag{2}$$

The simplified model by Irion *et al.* (2004) has also been widely accepted by researchers. Using Brown and Tucker's classification of product types, Irion *et al.* (2004) suggested [0.06, 0.1], [0.16, 0.20] and [0.21, 0.25] as the intervals of space elasticity to an increase in shelf space for unresponsive, moderately responsive and responsive products, respectively.

Lee's work (1961) defined retail space in two dimensions: an area, not a one-dimensional shelf space as with most of the studies, some of which are cited above. Dreze *et al.* (1994) also investigated the effect of space allocated to the sales in actual area, not as the length of shelf space. They empirically determined that the area allocated to a product is significant in determining the sales, using a Gompertz model provided below.

$$q_i = r_i e^{\beta_i \exp(\omega_i S_i)}. \tag{3}$$

where q_i is the demand for product i, r_i, β_i and ω_i are the parameters of the Gompertz model, and S_i is the area occupied by product i. More specifically, r_i is referred to as the horizontal asymptote where $q_i \leq r_i$, and $\beta_i < 0$ and $\omega_i < 0$ together control the increase in demand as a function of the increase in area.

2.2. *The effect of impulse purchases and customer traffic density*

The sales of a store broadly fall into two categories: planned purchases and unplanned purchases. Planned purchases are the portion of the sales that the customers enter the store with the intention of buying. Impulse purchases can be defined as those purchases that are not planned before the shopping event occurs (Kollat and Willet 1967). Decisions leading to these unplanned purchases are made within the store, and are affected by in-store stimuli (Wilkinson *et al.* 1982). The notion of impulse purchase has been studied by Bellenger *et al.* (1978), Wilkinson *et al.* (1982), Kollat and Willet (1967) and Fiore *et al.* (2000). The marketing literature suggests that different lines of products have different impulse purchase rates. For example, Bellenger *et al.* (1978) found that the percentages of impulse purchase were 27% for women's lingerie and 62% for costume jewellery.

The customer traffic within a store is another issue which has been investigated by marketing researchers. The customer traffic density throughout a store is not uniform; certain zones of a store have denser traffic than others. Empirical studies (Larson *et al.* 2005), stochastic models (Farley and Ring 1966) and agent-based modelling applications (Batty 2003) support this assertion.

3. Model development

The model presented here considers two criteria in evaluating the quality of a layout. These are the revenue generated by departments and the layout efficiency in terms of desired adjacencies. The revenue generated by the departments depends on the area allocated to the departments (area effect) and the location of the departments within the store (location effect). The layout efficiency measures how well the departments are placed in desired positions relative to each other (adjacency criterion). Definitions and mathematical formulations for the revenue and the layout efficiency are found in the subsequent sections.

3.1. *The effect of area on revenue*

The main motivation behind the use of the Gompertz model in the work by Dreze *et al.* (1994) is the assumption of 'bounded unit sales' for a given product. The assumption implies that there is a limit to the sales of a product, and once that limit (r_i) is reached, increasing the allotted area, S_i, cannot increase the sales, q_i. This assumption, however, is not valid here. A department defines a set of different merchandise lines and products that are grouped together, not a single item on a shelf. An increase in a department's area provides room for a more diversified set of products, which in turn creates a potential for increased sales. For this reason, the use of the power model ($q_i = r_i S_i^{\beta_i}$) is more suitable than the Gompertz model to define the relationship between the department area and the department revenue. The area effect on the revenue of a department is formulated as:

$$\bar{R}_i = r_i A_i^{\beta_j}. \tag{4}$$

and the total expected revenue for the whole store as:

$$\bar{R} = \sum_{i=1}^{n} \bar{R}_i + R_a. \tag{5}$$

where $R_a = r_a A_a^{\beta_a}$. R_a is the monetary contribution of aisle space to the overall revenue of the store that stems from the use of aisle space as a promotional display area.

3.2. *The effect of location on revenue*

As mentioned in Section 2.2, impulse purchases play an important role in generating revenues. In addition, different departments have different impulse purchase likelihoods. Combining these facts with varied customer traffic inside a store, a mechanism is needed that drives the departments with high impulse purchase likelihoods to be placed in more frequently visited parts of the store.

With this motivation in mind, two sets of ratings are defined: one is for the different zones of the store, denoted by **z**, with respect to traffic density; the other is for the departments, denoted by **d**, with respect to the likelihood of impulse purchases. Using these classifications, the store area is divided into three zones: high traffic zones ($z_k = 1$), medium traffic zones ($z_k = 2$) and low traffic zones ($z_k = 3$). An example partitioning of the retail space is depicted in Figure 2. In the same manner, departments are grouped into three classes: high impulse purchase departments ($d_i = 1$), medium impulse purchase departments ($d_i = 2$) and low impulse purchase departments ($d_i = 3$). These categories can then be used to measure the deviation of the departments' actual location from their most advantageous location, which is formulated as ($z_k - d_i$). If department i would benefit from a high traffic zone ($d_i = 1$) and is placed in a low traffic zone ,($z_k = 3$) the deviation would be positive, which indicates a missed revenue opportunity. On the other hand, if

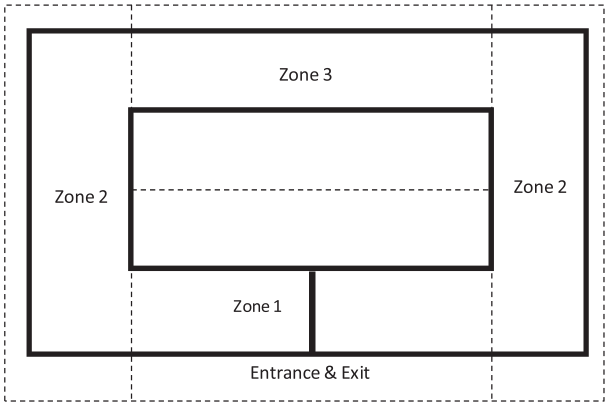

Figure 2. Partition of the store area into zones with respect to traffic density.

department i would not benefit from a high traffic zone ($d_i = 3$) and is placed in a high traffic zone ($z_k = 1$), the deviation would be negative. Situating such a department in a high traffic zone would not be a missed revenue opportunity for this department. Thus, for department i, the deviation from its desired location can be represented as $\max(0, z_k - d_i)$.

The revenue of a department is then formulated by taking the location effect into account as follows:

$$R_i = \frac{r_i A_i^{\beta_i}}{1 + \sum_{k=1}^{K} y_{ik}[\max(0, z_k - d_i)]} \tag{6}$$

where

$$y_{ik} = \begin{cases} 1 & \text{if department } i \text{ is in zone } k \\ 0 & \text{otherwise} \end{cases}$$

and the total revenue generated by the store is defined as:

$$R = \sum_{i=1}^{n} R_i + R_a \tag{7}$$

3.3. *Layout efficiency*

The layout efficiency, denoted by ε, defines the extent to which desired departmental adjacencies are satisfied by a given layout and formulated as:

$$\varepsilon = \frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(c_{ij}^{+}x_{ij}) - \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(c_{ij}^{-}(1 - x_{ij}))}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} c_{ij}^{+} - \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} c_{ij}^{-}} \tag{8}$$

where

$$x_{ij} = \begin{cases} 1, & \text{if department } i \text{ is adjacent to department } j \\ 0, & \text{otherwise} \end{cases} \qquad \forall i, j;\ i > j$$

In measuring the layout efficiency, the well-known closeness ratings concept from the manufacturing facilities layout literature is used (see for example Heragu 1997 or Tompkins *et al.* 2003). Traditionally, closeness ratings reflect the level of interaction between departments. In retail, one would place complementary departments near each other, such as women's jewellery

Table 1. Adjacency ratings used.

Rating	Definition	
A	Absolutely necessary	125
E	Especially important	25
I	Important	5
O, U	Ordinary closeness	1
X	Undesirable	−25
XX	Prohibited	−125

near women's fragrances or housewares near towels and bedding. If, for a given pair of departments, adjacency is not desired or prohibited, negative closeness ratings are used. An example might be such diverse departments as women's underwear and men's suits. Six closeness ratings are defined as detailed in Table 1. The purpose of the layout efficiency metric as defined here is to evaluate the quality of the layout from the adjacency satisfaction point of view.

3.4. *Overall model*

The authors' previous studies using a single objective version of this model (Yapicioglu 2008, Yapicioglu and Smith forthcoming) showed that R and ε are conflicting objectives and motivated the bi-objective optimization approach. A rectangular store area with polygon departments is considered. There is a single racetrack aisle with the same aspect ratio as the store and a single entry located midway through one of the longer sides. Departments are specified by a minimum area, a maximum aspect ratio and a specific revenue model according to area using the exponential form described earlier. The aisle is specified by a minimum and a maximum width as well as a revenue function analogous to the departments'. Given the definition of performance metrics from the previous sections, the complete bi-objective model is provided below:

$$\max \begin{cases} z_1 = R \\ z_2 = \varepsilon \end{cases} \tag{9}$$

s.t.

$$\frac{P_i}{4\sqrt{A_i}} \leq \bar{\alpha}_i \forall i, \tag{10}$$

$$A_i^L \leq A_i \forall i, \tag{11}$$

$$A_a^L \leq A_a \tag{12}$$

$$\sum_{i=1}^{n} A_i + A_a = A \tag{13}$$

$$\sum_{i=1}^{n} A_i y_i = A^0 \tag{14}$$

$$\sum_{i=1}^{n} A_i (1 - y_i) = A^I \tag{15}$$

$$W^L \leq W^a \leq W^U \tag{16}$$

$$y_i = \begin{cases} 1, & \text{if department } i \text{ is in the outer rim} \\ 0, & \text{otherwise} \end{cases} \quad \forall i.$$

Here, P_i, A_i and $\bar{\alpha}_i$ are the perimeter, the area and the maximum aspect ratio allowed for department i, respectively. A_i^L and A_a^L are the lower bounds for the area of department i and the aisle. Constraints (10) impose the aspect ratio requirements for each department and constraints (11) and (12) ensure that the areas allocated to the departments and to the aisles are larger than their lower bounds. Constraint (13) ensures that the total department and aisle area cannot exceed the area of the store (A). Constraints (14) and (15) ensure that the area allocated to the departments to the outer rim and to the inner region do not exceed the area available to the outer rim (A^0) and the inner region (A^I), respectively. Finally, constraint (16) guarantees that the width of the aisle (W^a) is within the prespecified limits (W^L and W^U).

4. Solution approach

Owing to the nonlinear nature of the revenue function, a two-stage optimization procedure is devised. In the first stage, the area allotments are determined by solving the following model:

$$\max \bar{R} = \sum_{i=1}^{n} \bar{R}_i + R_a \tag{17}$$

s.t

$$\sum_{i=1}^{n} A_i + A_a \leq A \tag{18}$$

$$A_i \geq A_i^L \forall i. \tag{19}$$

At this stage, retail space beyond the minimum needed for each department is allocated. Since the objective function of the model is not linear, a nonlinear optimization package can be used to solve the model. The objective function is denoted by $\bar{R}$ since the model given by Equations (17)–(19) disregards the zone requirements of the departments. Depending on the level of satisfaction of zone requirements, the actual revenue figure might be lower than the value generated by solving this model. Determining the areas of the departments prior to the metaheuristic search (tabu or genetic algorithm) enables the metaheuristic to focus on the combinatorial part of the problem (the permutation of departments).

Next, using the area allotments obtained from the model given above, two multi-objective metaheuristics, the multi-objective tabu search (MTS) from Kulturel-Konak *et al.* (2006) and the elitist non-dominated sorting genetic algorithm II (NSGA-II) from Deb *et al.* (2002), are used to optimize the layout of the departments and aisle. MTS was selected because a tabu search can exploit a combinatorial neighbourhood, such as the one here, effectively. NSGA-II was selected because it is the benchmark for multi-objective genetic algorithms and has shown strong performance on a variety of problems. A brief introduction to tabu search is provided in Section 4.1, followed by applications of tabu search in multi-objective settings in Section 4.2. The details of the tabu search implementation are presented in Section 4.3, and NSGA-II and its implementation are covered in Section 4.4.

4.1. *Tabu search*

Tabu search was introduced to the combinatorial optimization literature by Glover (Glover 1989, 1990, Glover and Laguna 1997, Reeves 1995). As the name suggests, the procedure is based on ‘prohibiting’ or disallowing certain moves in a systematic way to ensure that recently visited solutions are not reached again (cycling avoidance). Avoiding cycling gives tabu search the ability to search the solution space more comprehensively, without being trapped in local optima. Tabu search has successful implementations in many combinatorial, NP-hard optimization problems including quadratic assignment (Taillard 1991), unequal area facility layout (Kulturel-Konak *et al.* 2004), vehicle routing (Taillard *et al.* 1997), redundancy allocation (Kulturel-Konak *et al.* 2003), job shop scheduling (Barnes and Chambers 1995) and weighted maximal planar graphs (Osman 2006).

4.2. *The use of tabu search in multi-objective settings*

The first known application of tabu search in a multi-objective setting was by Hertz *et al.* (1994). In their article, they considered a single objective at a time, and treated the remaining objective functions as constraints. Later, Hansen (1997, 2000) proposed a multi-objective tabu search algorithm, TAMOCO, where a set of current solutions with their respective tabu lists are employed. Each current solution uses a set of differing weights to evaluate the quality of the solution with respect to the objective functions considered. In this respect, TAMOCO resembles the vector evaluated genetic algorithm (VEGA) of Schaffer (1985).

In recent years, there have been studies using multi-objective tabu search variants in continuous optimization problems. Gandibleux and Freville (2000) proposed a multi-objective tabu search algorithm for the 0-1 multi-objective knapsack problem. In their approach, they use a very efficient method, called decision space reduction, using the special structure of the 0-1 knapsack problem, which is not applicable to this model. Jaeggi *et al.* (2008) incorporated a local search algorithm, the Hooke and Jeeves method, to their multi-objective tabu search application. This approach was not employed in this study, as the continuous part of the formulation is solved exactly using an exact nonlinear optimizer. In this article, the multi-objective tabu search approach devised by Kulturel-Konak *et al.* (2006) is used. In their approach, called MTS, instead of having a set of current solutions, a single current solution is kept. The quality of a solution is evaluated with respect to one of the objective functions randomly selected from the set of objective functions. At every iteration, the updated solution is compared with the non-dominated solutions kept in an external repository. They demonstrated that MTS performed better than TAMOCO in a series of redundancy allocation problems.

4.3. *Multi-objective tabu search*

To solve the model, one approach used is the MTS algorithm as explained in Kulturel-Konak *et al.* (2006). The solution representation, the decoding/encoding mechanisms and move operators are given in subsequent sections. In order to enforce the aspect ratio constraints (Equatio 10), a penalty function in multiplicative form is used.

Solution representation and decoding

The solution representation is depicted in Figure 3. It is an extension of the flexible bay structure (FLEXBAY) representation which is used extensively in the facility layout literature. The standard FLEXBAY representation was introduced by Tong (1991). In FLEXBAY, the allocation of departments is done by allowing guillotine cuts in one direction to obtain bays, and perpendicular cuts within bays to obtain departments.

The encoding is composed of the sequence of departments, **p**, followed by two baybreaks. The departments from the beginning of the permutation up to the first baybreak, b_1, are assigned to

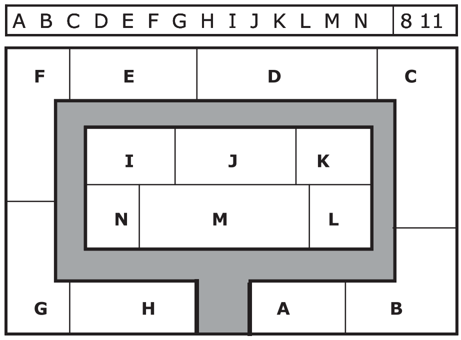

Figure 3. Solution representation and the corresponding layout.

the outer bay, which is placed outside the aisle. The departments starting from the first baybreak up to the second baybreak are assigned to the upper bay, while the remaining departments are assigned to the lower bay. The lower bay and the upper bay comprise the inner section of the store (inside the aisle). Assignment of the departments to the outer bay starts from the lower midpoint of the rectangular store area and continues counterclockwise. The placements inside the aisle are done using boustrophedon ordering starting from the upper left corner. The initial permutation and the corresponding baybreak combinations are generated randomly.

Move operator

Two different move operators are used:

Swap. The swap operator exchanges the places of the two departments located in the ith and jth positions in the permutation $\mathbf{p}$ where $i = 1, \ldots, n-1$ and $j = i+1, \ldots, n$.

Re-partite. The re-partite operator is used to change the number of departments allocated to each bay. After each swap operation, the newly obtained permutation is coupled with every possible partition combination to obtain new solutions using sets $\mathbf{b}_1$ and $\mathbf{b}_2$. For instance, as illustrated in Figure 3, partitioning 8–11 indicates that there are eight departments in the outer bay, three departments in the upper bay, and the remaining departments in the lower bay. Definitions of sets $\mathbf{b}_1$ and $\mathbf{b}_2$ are provided in the candidate list strategies section.

Tabu list entries

In the tabu list, the most recent department pairs swapped are kept. To apply dynamic tabu tenure, a lower bound and an upper bound on the tabu tenure are specified. Every 20 iterations, the size of the tabu list is determined according to an integer uniform random number between these lower and the upper bounds. The lower and the upper bounds on the tabu tenure were determined experimentally and are set to five and eight, respectively.

Neighbourhood definition

The neighbourhood of a given solution is considered in two different dimensions: permutation and partitioning. In the first stage, partitioning is kept constant and from a given permutation, all other permutations that can be reached by a single swap operation are considered. The number of solutions that can be reached using the swap operator by assuming no change in the partitioning is $n(n-1)/2$.

(a)

A	B	C	D	E

(b)

2	4

(c)

B	A	C	D	E
C	B	A	D	E
D	B	C	A	E
E	B	C	D	A
A	C	B	D	E
A	D	C	B	E
A	E	C	D	B
A	B	D	C	E
A	B	E	D	C
A	B	C	E	D

(d)

1	2
1	3
1	4
2	3
2	4
3	4

Figure 4. The initial solution: (a) permutation, (b) partitioning, (c) permutation neighbourhood, and (d) partitioning neighbourhood.

In the second stage, the permutation is kept constant, and the search is performed in the partition neighbourhood. The number of solutions that can be reached by changing the partition is $(n-1)(n-2)/2$ for a given permutation. Thus, the number of different solutions that can be reached from any given solution is $n(n-1)^2(n-2)/4$ provided that a complete neighbourhood is searched for both swap and partitioning. For example, if $n = 20$, then the size of the neighbourhood is 32,490. An example solution and corresponding neighbourhood structure are provided in Figure 4. The size of the neighbourhood is reduced by candidate list strategies, as explained in the following sections.

Objective functions

Equations (20) and (21) present the two objective functions used. These fitness functions are: the adjacency score (F_a) and the revenue (F_r). Both objective functions include a common term, the penalty function, which is to reduce the quality of solutions that have departments violating the departmental aspect ratio constraint. In Equations (20) and (21), the penalty term is $(n - s/n)^\kappa$, where n is the number of departments, s is the number of departments violating the aspect ratio, and κ is the exponent controlling the severity of the penalty function. The value of the exponent κ is determined experimentally.

$$F_a = \varepsilon \left(\frac{n-s}{n}\right)^\kappa \tag{20}$$

$$F_r = R \left(\frac{n-s}{n}\right)^\kappa \tag{21}$$

For the purposes of this study, two departments are considered to be adjacent if they share a common edge. For the departments that are separated by the aisle, two departments are said to be adjacent if those two departments are across from each other. For instance, in Figure 3, department L is adjacent to departments A, B and C, in addition to departments K and M.

With these settings, the pseudo-code of the neighbourhood generation is given as follows:

```
Procedure tabuSearchNeighbourhoodGeneration (Solution CURRENT){
    Initialize neighbourhood array nArray;
    for (i = 0; i < n − 1; i + +){
        for (j = i + 1; j < n; j + +){
            Departments dP ← swap departments i, j of CURRENT;
```

```
                Determine sets b1 and b2 using A^i and A_a;
                for (b1 ∈ b1){
      for (b2 ∈ b2, b1 < b2){
                          nArray.add(Solution(dP, b1, b2));
                    }end for
                }end for
            }end for
      }end for
      sort nArray;
      return nArray;
}end procedure
```

Aspiration criteria

If a solution within the neighbourhood dominates a solution from the set of non-dominated solutions, a move to that solution is allowed even if the move is tabu.

Candidate list strategies

For this problem, $n(n-1)^2(n-2)/4$ distinct solutions can be reached from any given solution. Of these solutions, $n(n-1)/2$ are due to distinct permutations, and for each distinct combination there are $(n-1)(n-2)/2$ baybreak combinations. As mentioned in the section on the solution representation, the location of the first baybreak, b_1, determines the number of departments inside and outside the aisle. The location of b_1, thus, determines the width of the aisle. As the value of b_1 increases, the number of departments outside the aisle increases, and the aisle runs closer to the centre, making it wider. In contrast, as the value of the b_1 decreases, the number of departments outside the aisle decreases, and it runs further from the centre, becoming narrower. Hence, there are natural limits over the values that b_1 can assume that are imposed by the lower and upper bounds on the aisle width. Given the total area of the departments inside the aisle and the aisle area, the width of the aisle can be calculated as:

$$W_a = \frac{1}{2}(\sqrt{(A^I + A_a)/\alpha^f} - \sqrt{A^I/\alpha^f}). \tag{22}$$

where $A^I = \sum_{i=b_1}^{n} A_i, A_a$, is the aisle area and α^f is the facility's aspect ratio. Equation (22) is derived using Euclidean geometry. Given the area of a rectangle and the ratio of the longer side of the rectangle to the shorter side, one can calculate the lengths of both sides. Using this formulation, the aisle width is calculated as half of the difference between the lengths of the shorter side of the two rectangles whose areas are given by A^I and $A^I + A_a$, respectively.

Then, the set of feasible first baybreaks for the sequence of departments, $\mathbf{p}$, is defined as:

$$\mathbf{b}_1 = \{k | W_a \geq W_a^L, W_a \leq W_a^U\} \tag{23}$$

Finally, for each $k \in \mathbf{b}_1$, the set of second baybreaks is defined as:

$$\mathbf{b}_2 = \{m | m > k, m < n, k \in \mathbf{b}_1, b_1 < b_2\} \tag{24}$$

where n is the number of the departments in the permutation. With the formulae provided in Equations (23) and (24), the number of baybreak combinations considered is reduced substantially.

In choosing the values of W_a^L and W_a^U, the natural lower and upper bounds on the aisle width imposed by the aisle area and the department areas can be of help by using two extreme cases defined below.

Consider a department sequence where only the smallest department is outside the aisle. This is the case where the width of the aisle is the smallest. Thus, a lower bound on the aisle width can be defined as:

$$\underline{W_a} = \frac{1}{2}\left[\sqrt{\frac{\sum_{i=1}^{n} A_i - \min(A_i) + A_a}{\alpha^f}} - \frac{\sum_{i}^{n} A_i - \min(A_i)}{\alpha^f}\right] \tag{25}$$

By the same token, consider a department sequence where only the smallest department is inside the aisle, and the rest of the departments are placed outside the aisle. This is the case where the aisle width is at its widest. Hence the upper bound on the aisle width is given by:

$$\bar{W}_a = \frac{1}{2}\left(\sqrt{\frac{\min(A_i) + A_a}{\alpha^f}} - \sqrt{\frac{\min(A_i)}{\alpha^f}}\right) \tag{26}$$

Any feasible layout has to satisfy the condition $\underline{W_a} \leq W_a \leq \bar{W}_a$, so the lower and upper bounds of the aisle width, W_a^L and W_a^U also have to satisfy $\underline{W_a} \leq W_a^L \leq W_a^U \leq \bar{W}_a$.

The values $\underline{W_a}$ and $\bar{W}_a$ might yield extreme values that are impractical (*e.g.* $\underline{W_a} = 2$ ft and $\bar{W}_a = 50$ ft). In such a case, tightening the interval (W_a^L, W_a^U) to reflect the decision maker's preferences would further reduce the size of $\mathbf{b}_1$, leading to a smaller baybreak neighbourhood.

Diversification and intensification strategies

Diversification and intensification strategies are used to move the search from a different solution space area if there is no improvement for a certain number of moves in the non-dominated solution set. The three approaches investigated in this study are:

(1) If no update is performed to the non-dominated solution set for 50 consecutive iterations, the search is diversified using a newly generated random solution.
(2) If no update is performed to the non-dominated solution set for 50 consecutive iterations, the search is intensified using a random solution picked *from the non-dominated solution set.*
(3) If no update is performed to the non-dominated solution set for *100 consecutive iterations*, the search is intensified using a random solution picked from the non-dominated solution set.

Whenever the search is diversified or intensified, tabu list and non-improvement counter are reset, but the non-dominated solution repository is retained. Among the settings above, option (3) gave the best results among the computational trials.

Termination criteria

To terminate the search from a given initial solution, the number of moves that do not lead to an update of the non-dominated solution set is recorded. If an update to the non-dominated solution set has not been made for a certain number of consecutive moves, the search terminates. Every time the non-dominated solution set is updated, the non-improving move counter starts from zero again. For the 12 department example, the search is terminated after 1000 moves without an update to the non-dominated solution set. Similarly, the termination criterion of 10K moves was used for the 20 department problem. Given the determination of sets $\mathbf{b}_1$ and $\mathbf{b}_2$ and the definition of the neighbourhood generation procedure, the bi-objective tabu search algorithm to solve the retail store design problem is given as follows:

```
Procedure MultiObjectiveTabuSearch{
        initialize CURRENT,Tenure, minTenure, maxTenure;
        initialize iter = 0, iterDiversification, iterMax;
        initialize paretoSet
        BEST_ADJ ← CURRENT;
        BEST_REV ← CURRENT;
        initialize tabuList, nArray;
        do {
                select fitnessMode(F_r, F_a)
                nArray ← tabuSearchNeighbourhoodGeneration(CURRENT);
                Tenure = minTenure + U(0, 1)(maxTenure − minTenure)
                for(i = 0; i < nArray.size(); i + +){
                        if (fitMode = = F_r)
                                if(nArray [i].fitness > BEST_REV.fitness){ //Aspiration
Criteria
                                        CURRENT ← nArray [i];
                                        BEST_REV ← CURRENT;
                                        add BEST_REV to paretoSet;
                                        update tabuList;
                                        iter = 0;
                                        break;
                                }end if
                                if(nArray [i] is not a tabu){
                                        add BEST_REV to paretoSet;
                                        update tabuList;
                                        break;
                                }end if
                        end if }
                        if (fitMode == F_a)
                                if(nArray [i].fitness > BEST_ADJ.fitness){ //Aspiration
Criteria
                                        CURRENT ← nArray [i];
                                        BEST_ADJ ← CURRENT;
                                        add BEST_ADJ to paretoSet;
                                        update tabuList;
                                        iter = 0;
                                        break;
                                }end if
                                if(nArray [i] is not a tabu){
                                        CURRENT ← nArray [i];
                                        add BEST_ADJ to paretoSet;
                                        update tabuList;
                                        break;
                                }end if
                        end if }
                }end for
                perform dominance check on paretoSet
                if (iter > iterDiversification){
                        pick a solution from paretoSet randomly, DIVER;
                        CURRENT ← DIVER;
```

```
                reset **tabuList**;
            }end if
            iter++;
    }while iter < iterMax
    print **paretoSet**;
}end procedure
```

4.4. *Elitist non-dominated sorting genetic algorithm: NSGA-II*

The elitist non-dominated sorting genetic algorithm, NSGA-II, was proposed by Deb *et al.* (2002) and is the most well-known multi-objective genetic algorithms. NSGA-II uses the basic operators of genetic algorithms, such as population, parent selection, crossover and mutation. However, none of these operators utilizes the true values of the objective functions. Instead, NSGA-II defines two metrics to identify the quality of a solution. The first metric is the rank, assigned to solutions based on their location with respect to the theoretical Pareto front. The closer the solution is to the theoretical Pareto front, the higher is the rank. The second metric is the crowding distance, which is a measure of how crowded the Pareto archive is around a solution. With these two metrics, a solution is preferred to another if it has a higher rank, or if the two solutions have the same rank, then a solution with a larger crowding distance is preferred to the other, as the larger crowding distance is assumed to be an indication of a less explored region of a Pareto front. The other details of NSGA-II are as follows.

The solution representation and decoding used with the NSGA-II is the same as that used with MTS. An initial population of size $n = 100$ is created randomly. Once the initial population is created, the rank and the crowding distance of the individuals are calculated as per Deb *et al.* (2002). Although Deb *et al.* (2002) proposed a methodology for constraint handling to be used with NSGA-II, the fitness functions given by Equations (20) and (21) are utilized within NSGA-II, so as to make a fair comparison between the two optimization methods.

In parent selection, binary tournament selection is used. First, two individuals from the parent population are randomly selected. If one of the individuals has a higher rank than the other, then that individual is selected as one of the parents. If both individuals have the same rank, then the individual with the larger crowding distance wins the tournament. The same selection process is repeated to select the next parent. Once the parents are selected, two offspring are created from each parent pair. For this, partially mapped crossover (PMX) (with a crossover rate of 0.9) as defined by Goldberg and Lingle (1985) with a mutation operator (with a probability of 0.005) is used for the permutation part of the solution representation. The mutation operator is a simple swap, where two randomly selected members of the permutation are interchanged. To ensure permutation feasibility, a repair mechanism is employed. After offspring permutation is determined, the baybreak set from one parent or the other is assigned to an offspring with equal probability. With these settings, an offspring population of n individuals is created at every generation.

To select the next generation of individuals, the parent population and the offspring population are merged into a single population of a size $2n$. Next, the ranks of the individuals are calculated and the individuals are sorted in decreasing order of their ranks. The first n individuals compose the next generation. The search terminates after 12,500 generations. This number of generations is determined to allow the NSGA-II to run for the same number of function evaluations as that of MTS, on average. To achieve this, MTS is run for 30 times and the number of function evaluations is recorded. Then, the average of number of function evaluations is calculated and divided by 100, the population size of NSGA-II, to arrive at the number of generations. When the search terminates, individuals with the highest rank constitute the Pareto archive identified by NSGA-II.

5. Computational experience

Since the model proposed in this study is novel, two test problems with 12 and 20 departments were created. In addition, three facility sizes, 16×24, 17×25.5 and 18×27, were defined. Area requirements for the departments and adjacency matrices were created randomly. With these settings, there is a total of six problems. (Data for the test cases are available from the authors upon request.) The performance of MTS and NSGA-II is compared on the 12-department problems. The results show that MTS outperforms NSGA-II decisively and, therefore, the 20-department problem is addressed by only the MTS. The MTS results are presented first, followed by a comparison with those from the NSGA-II.

5.1. *Multi-objective tabu search algorithm*

The MTS approach from Kulturel-Konak *et al.* (2006) changes the objective function at every iteration of the algorithm according to the respective probabilities of objective functions. By the same token, p_a and p_r are defined as the probabilities of using either F_a or F_r as the fitness function at a given iteration of the bi-objective tabu search ($p_a + p_r = 1$). For instance, if $p_a = p_r = 0.5$ is used, both objective functions are equally likely to be active at a given iteration of the search. The authors' previous experience suggests that maximizing F_a is more difficult than maximizing F_r, hence five settings of p_a and p_r are used as defined in Table 2, all of which favour working on the F_a objective (except the one with equal probabilities).

For each problem instance, the best combination of p_r, p_a pairs and the number of non-dominated solutions are tabulated in Table 3. In selecting the best (p_r, p_a) pairs, two criteria are considered: the coverage of the Pareto archive obtained by different configurations of p_r and p_a, and how many non-dominated solutions for each configuration are dominated by those found by the best configuration. For instance, in problem 1, with configuration $p_r = p_a = 0.5$ the non-dominated solution set dominates three and one non-dominated solutions found with configurations $p_r = 0.1$, $p_a = 0.9$ and $p_r = 0.2\, p_a = 0.8$, respectively, and has one more non-dominated solution compared to non-dominated sets of configurations $p_r = 0.1, p_a = 0.7$ and $p_r = 0.4, p_a = 0.6$.

Table 2. Multinomial probability mass function settings.

p_r	p_a
0.1	0.9
0.2	0.8
0.3	0.7
0.4	0.6
0.5	0.5

Table 3. Best probability mass function settings by problem.

Problem	n	A	p_r	p_a	No. of Pareto efficient solutions
1	12	16×24	0.5	0.5	7
2	12	17×25.5	0.1, 0.3, 0.5	0.9, 0.7, 0.5	4
3	12	18×27	0.2, 0.4	0.8, 0.6	9
4	20	16×24	0.4	0.6	12
5	20	17×25.5	0.1	0.9	10
6	20	18×27	0.3	0.7	8

Table 4. Pareto efficient solutions summary.

Problem	n	Area	F_r Min.	F_r Max.	F_a Min.	F_a Max.	CPU time (s)
1	12	16 × 24	10,693	12,056	0.446	0.756	219
2	12	17 × 25.5	11,195	11,804	0.580	0.697	528
3	12	18 × 27	11,260	12,691	0.529	0.720	424
4	20	16 × 24	14,261	15,989	0.738	0.810*	15,093
5	20	17 × 25.5	14,768	16,494	0.750	0.821*	10,711
6	20	18 × 27	14,815	16,915	0.772	0.832*	17,561

Note: For the $n = 20$ problems, the best F_a values found improve upon the best solutions reported earlier (Yapicioglu 2008); these values are marked with an asterisk (*).

Table 5. Comparison with upper bounds, $n = 12$.

$n = 12$	16 × 24	17 × 25.5	18 × 27
$(F_r)\hat{}$	12,782	13,117	13,404
$(F_a)\hat{}$	0.948	0.948	0.948
(F_r)	12,056	11,804	12,691
Deviation	5.68%	10.01%	5.32%
(F_a)	0.756	0.697	0.720
Deviation	20.24%	26.47%	24.05%

Table 6. Comparison with upper bounds, $n = 20$.

$n = 20$	16 × 24	17 × 25.5	18 × 27
$(F_r)\hat{}$	15,989	16,494	16,915
$(F_a)\hat{}$	0.935	0.935	0.935
(F_r)	15,989	16,494	16,915
Deviation	0.00%	0.00%	0.00%
(F_a) (Bi-obj)	0.810	0.821	0.832
Deviation	13.40%	12.20%	11.02%
(F_a) (Single obj)	0.809	0.819	0.820
Deviation	13.58%	12.41%	12.19%

In Table 4, the coverage of the Pareto efficient solutions (*e.g.* the minimum and maximum values of F_r and F_a) are provided. For all problem instances, the best reported solutions from Yapicioglu (2008) for the single objective counterparts of the same problem are reached. Moreover, for the $n = 20$ problems, the best F_a values found improve upon the best solutions reported earlier (Yapicioglu 2008). These values are denoted by (*) in Table 4. The reasons for this situation are twofold: the bi-objective versions of the problems are allowed to run for more iterations, and the occasional switches to F_r during the course of the search serve as a useful diversification strategy for the fitness function F_a.

In Tables 5 and 6, the overall performance of the MTS algorithm is summarized with respect to the upper bounds calculated for $n = 12$ and $n = 20$, respectively. For $n = 12$, the best known solutions from the single objective formulations for all cases have been identified. For $n = 20$, although better solutions are discovered in terms of , the largest improvement is only around 1%.

Finally, in Figures 5–10 pictorial representations of the Pareto archives obtained for the six cases are provided, followed by two example layouts for each test problem in Figures 11–16. The previously best known solutions in terms of F_a are also depicted for each test problem in Figures 11–16. It can be seen in all cases that there is a trade-off between increased revenue and improved adjacency. What is particularly striking is that there is a point on most graphs where F_a

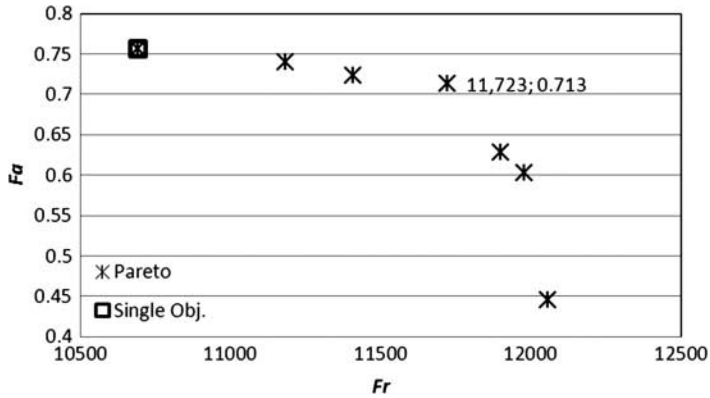

Figure 5. Pareto archive, $n = 12$, $A = 16 \times 24$, $p_r = 0.5$, $p_a = 0.5$.

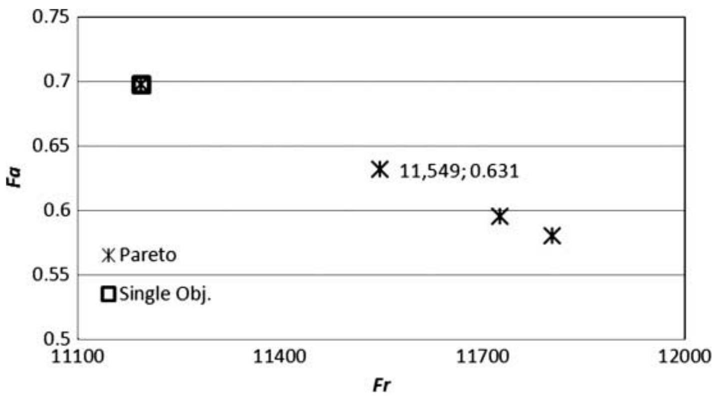

Figure 6. Pareto archive, $n = 12$, $A = 17 \times 25.5$, $p_r = 0.1, 0.3, 0.5$, $p_a = 0.9, 0.7, 0.5$.

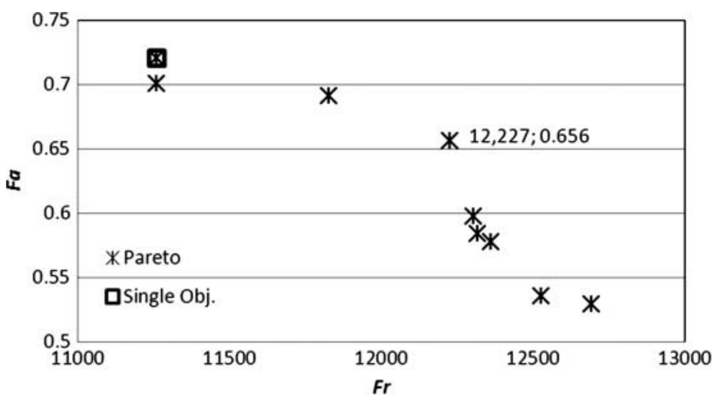

Figure 7. Pareto archive, $n = 12$, $A = 18 \times 27$, $p_r = 0.2, 0.4$, $p_a = 0.8, 0.6$.

falls off rapidly against F_r. Therefore, a good choice of layout might be the layout with the largest F_r which occurs just before the significant decrease in F_a. Among the Pareto efficient solutions, the choice of the best solution should be made by the decision maker; however, the authors' choice of 'best' is indicated in Figures 5–10 by labelling the objective function values.

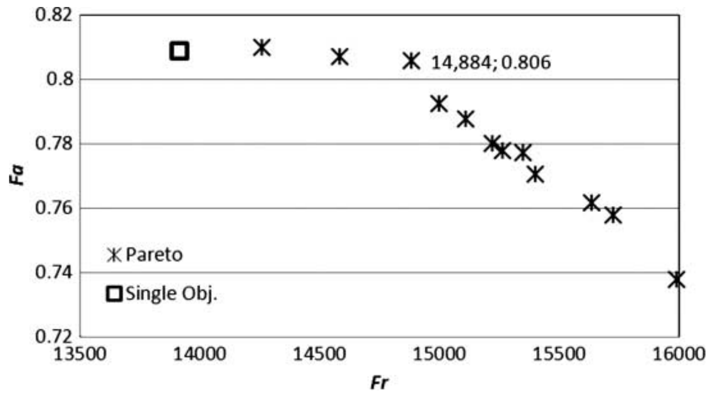

Figure 8. Pareto archive, $n = 20$, $A = 16 \times 24$, $p_r = 0.4$, $p_a = 0.6$.

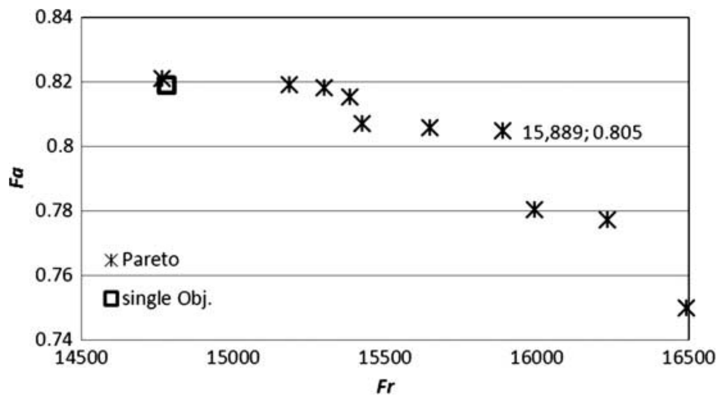

Figure 9. Pareto archive, $n = 20$, $A = 17 \times 25.5$, $p_r = 0.1$, $p_a = 0.9$.

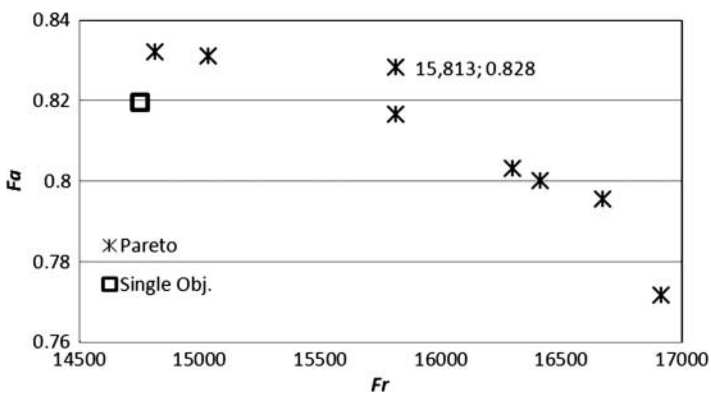

Figure 10. Pareto archive, $n = 20$, $A = 18 \times 27$, $p_r = 0.3$, $p_a = 0.7$. (a) $F_r = 12,056$, $F_a = 0.446$; (b) $F_r = 10,693$, $F_a = 0.756$.

5.2. *Multi-objective tabu search versus non-dominated sorting genetic algorithm II*

In this section, the performance of MTS is compared with that of NSGA-II using both visual inspection and statistical analyses. As mentioned in Section 5.1, one of the MTS parameters

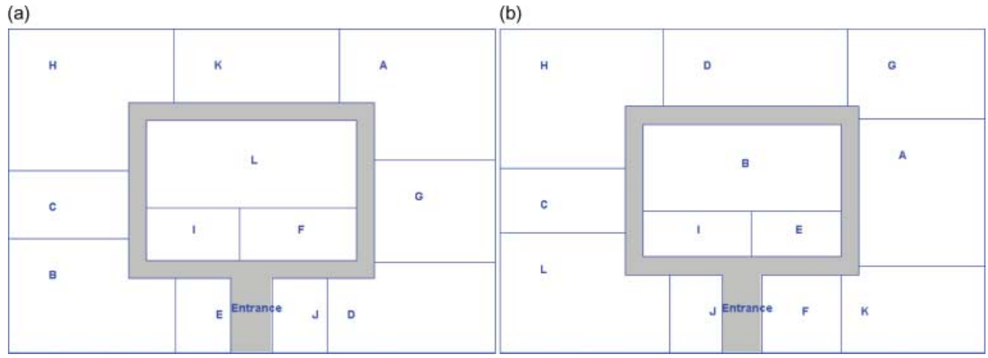

Figure 11. Example layouts, $n = 12, A = 16 \times 24, p_r = 0.5, p_a = 0$. (a) $F_r = 11,804$, $F_a = 0.580$; (b) $F_r = 11,195$, $F_a = 0.697$.

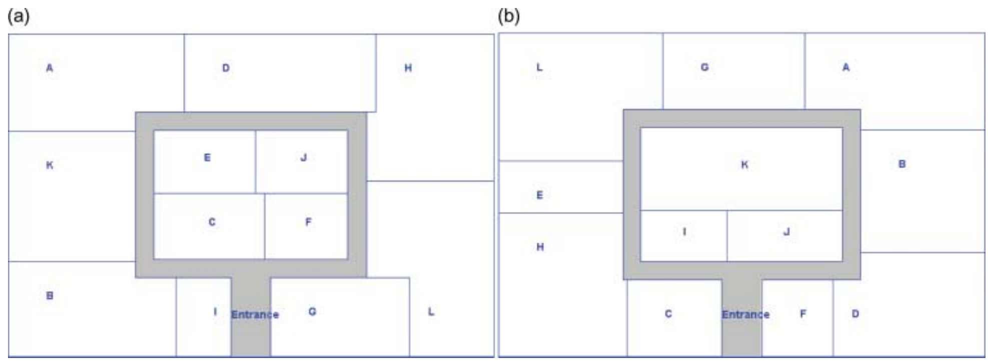

Figure 12. Example layouts, $n = 12$, $A = 17 \times 25.5$, $p_r = 0.1$, 0.3, 0.5, $p_a = 0.9$, 0.7, 0.5. (a) $F_r = 12,691$, $F_a = 0.529$; (b) $F_r = 11,260$, $F_a = 0.720$.

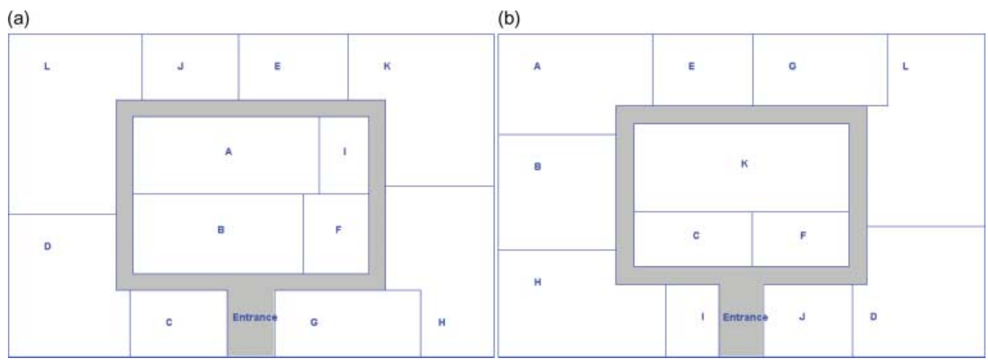

Figure 13. Example layouts, $n = 12$, $A = 18 \times 27$, $p_r = 0.2$, 0.4, $p_a = 0.8$, 0.6. (a) $F_r = 15,989$, $F_a = 0.738$; (b) $F_r = 14,261$, $F_a = 0.810$.

that needs tuning is the multinomial probability mass function (PMF) settings, and these are established through experimentation. Then, using these settings the MTS is run 30 times for $n = 12$ and $A = 16 \times 24$, 17×25.5 and 18×27, respectively. With the implementation details provided in Section 4.4, NSGA-II is run 30 times as well. The results are first analysed from a feasibility point of view. In Table 7, 'All feasible' means that the solutions in the Pareto archive are all feasible, 'Some feasible' means that the Pareto archive contains both feasible and infeasible

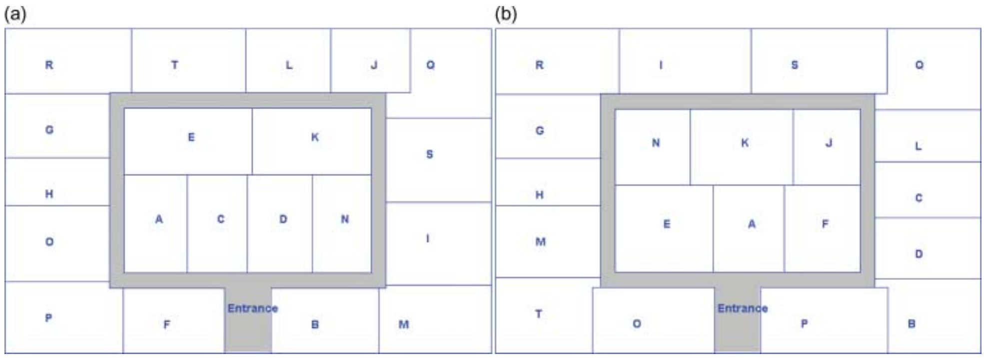

Figure 14. Example layouts, $n = 20, A = 16 \times 24, p_r = 0.4, p_a = 0.6$. (a) $F_r = 16,232, F_a = 0.777$; (b) $F_r = 14,768$, $F_a = 0.821$.

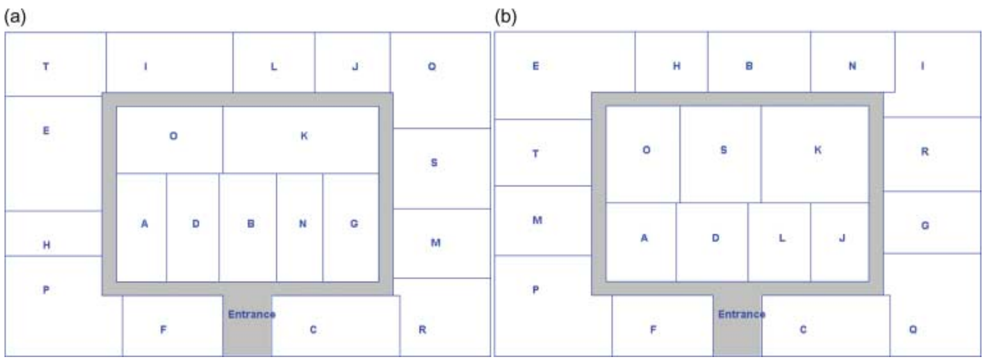

Figure 15. Example layouts, $n = 20$, $A = 17 \times 25.5$, $p_r = 0.1$, $p_a = 0.9$. (a) $F_r = 16,915$, $F_a = 0.772$; (b) $F_r = 14,815$, $F_a = 0.832$.

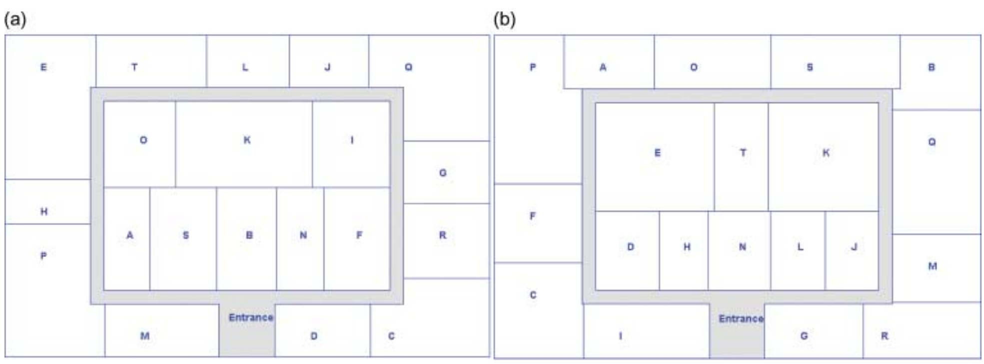

Figure 16. Example layouts, $n = 20, A = 18 \times 27, p_r = 0.3, p_a = 0.7$.

solutions, and 'No feasible' means that all solutions in the Pareto archive are infeasible. As Table 7 suggests, MTS is able to identify Pareto archives that comprise feasible solutions across 30 trials for all problems, whereas NSGA-II falls quite short of that. In particular, in Problem 2, NSGA-II identifies only one Pareto archive where solutions are all feasible and the remaining Pareto archives are all composed of infeasible solutions.

In Figures 17–19, Pareto optimal solutions identified across 30 trials are aggregated for both MTS and NSGA-II. These figures show that MTS consistently reached very similar Pareto archives

Table 7. Summary of feasible solutions identified.

	MTS			NSGA-II		
Problem	All feasible	Some feasible	No feasible	All feasible	Some feasible	No feasible
1	30	0	0	22	1	7
2	30	0	0	1	0	29
3	30	0	0	7	4	19

Note: MTS = multi-objective tabu search; NSGA-II = non-dominated sorting genetic algorithm II.

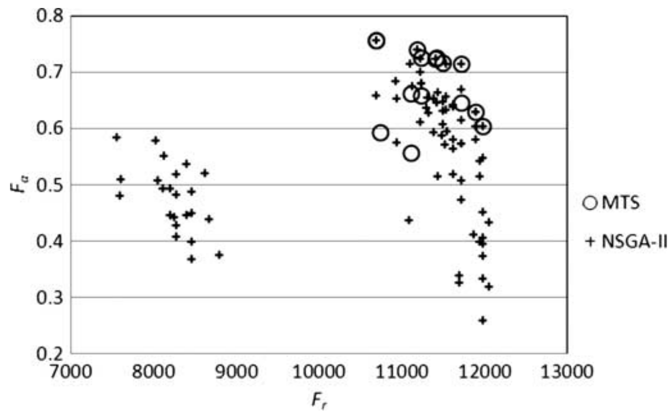

Figure 17. Multi-objective tabu search (MTS) *vs* non-dominated sorting genetic algorithm II (NSGA-II), $n = 12$, $A = 16 \times 24$.

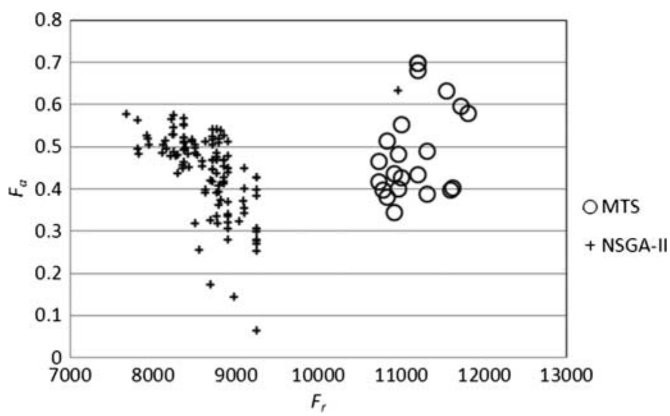

Figure 18. Multi-objective tabu search (MTS) *vs* non-dominated sorting genetic algorithm II (NSGA-II), $n = 12$, $A = 17 \times 25.5$.

where the non-dominated solutions are very close to each other. While in some trials for Problems 1 and 3 NSGA-II identified the same Pareto archive as MTS, the fronts identified by the genetic algorithm varied significantly from trial to trial. For Problem 2, the situation is worse: none of the Pareto archives identified by MTS is identified by NSGA-II. More seriously, except for one trial, NSGA-II failed to identify Pareto solutions without aspect ratio constraints violation for Problem 2 (this is evident in the very poor objective function values because they are penalized values). In the only trial where NSGA-II identified a feasible solution, the Pareto set consisted of only

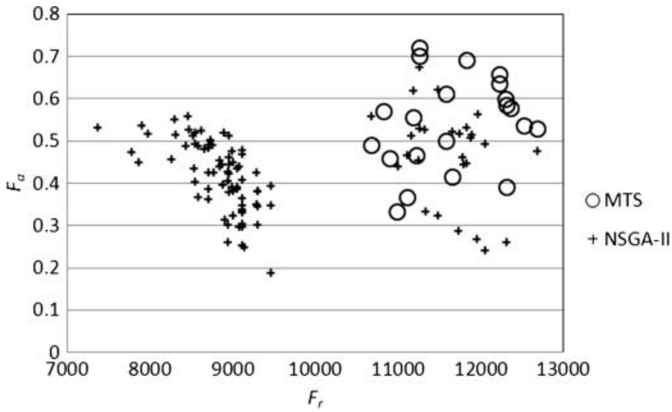

Figure 19. Multi-objective tabu search (MTS) *vs* non-dominated sorting genetic algorithm II (NSGA-II), $n = 12$, $A = 18 \times 27$.

Table 8. Summary of the Mann–Whitney test.

Problem	Objective function	*n*		Median		95% CI	*W*	*p*
		MTS	NSGA-II	MTS	NSGA-II			
1	F_r	15	88	11242	11414	(−300; 437)	783.0	0.9814
	F_a	15	88	0.71346	0.57225	(0.07059; 0.18195)	1225.5	0.0000
2	F_r	22	115	11098	8726	(2280; 2627)	2775.0	0.0000
	F_a	22	115	0.45115	0.46781	(−0.02126; 0.09247)	1714.0	0.2517
3	F_r	21	102	11586	9070	(1963; 2818)	2094.5	0.0000
	F_a	21	102	0.55419	0.44078	(0.06379; 0.16669)	1909.5	0.0000

Note: MTS = multi-objective tabu search; NSGA-II = non-dominated sorting genetic algorithm II; CI = confidence interval.

Table 9. Summary of the Kruskal–Wallis test.

Problem	Objective function	*n*		Median		Average rank		*Z*		*p*
		MTS	NSGA-II	MTS	NSGA-II	MTS	NSGA-II	MTS	NSGA-II	
1	F_r	15	88	11242	11414	52.2	52	0.03	−0.03	0.978
	F_a	15	88	0.71346	0.57225	81.7	46.9	4.17	−4.17	0.000
2	F_r	22	115	11098	8726	126.1	58.1	7.37	−7.37	0.000
	F_a	22	115	0.45115	0.46781	77.9	67.3	1.15	−1.15	0.251
3	F_r	21	102	11586	9070	99.7	64.2	5.33	−5.33	0.000
	F_a	21	102	0.55419	0.44078	90.9	56.0	4.08	−4.08	0.000

Note: MTS = multi-objective tabu search; NSGA-II = non-dominated sorting genetic algorithm II.

one solution, and that solution was dominated by the Pareto optimal solutions identified by MTS (visible in the top right-hand corner just below the front found by MTS).

Visual results depicted in Figures 17–19 clearly show that MTS performs better than NSGA-II. A likely explanation is that the tabu search better leverages the permutation encoding and natural neighbourhood of the model. The performance of MTS is compared with that of NSGA-II using two appropriate non-parametric statistical tests, namely the Mann–Whitney and Kruskal–Wallis tests. The results are summarized in Table 8 for the Mann–Whitney test and in Table 9 for the Kruskal–Wallis test. For Problem 1, the difference between the median values of F_a is highly significant, while in Problem 2, the difference between the median values of F_r is highly significant.

The contrast between the two algorithms is most apparent in Problem 3 where the differences of the median values of both F_a and F_r are highly significant. Considering the analyses of results clearly shows that MTS outperforms NSGA-II on this problem set. MTS can take advantage of the natural neighbourhood of the encoding to more efficiently and effectively search the solution space.

6. Conclusions

In this article, a bi-objective model for the spatial design of a retail store was considered. The proposed model considers the revenue generated and the degree of adjacency satisfaction among departments as equally important factors, subject to area, shape and aisle width constraints. The results show that using a bi-objective formulation has advantages over the single objective formulation because solutions that compromise high revenues with desired adjacency can be identified. The optimization is accomplished by first considering the nonlinear part and optimizing it exactly (that is, the department areas are established exactly). These are then fed into the metaheuristic (tabu search or genetic algorithm) which selects the location of each department within the facility. Note that this separation of the problem into nonlinear and combinatorial components might cause the overall procedure to miss a globally optimal solution. However, the two-part procedure is tractable and effective.

The failure of usually successful NSGA-II in this problem can be explained by its general purpose, whereas the MTS exploits the neighbourhood structure of the permutation encoding. The disruptive nature of crossover in the genetic algorithm also impaired the identification of designs which adhered to the aspect ratio constraint in one of the three problems. Turning to MTS, the results showed that the multinomial probability mass function plays an important role. In general, as the number of departments increases, maximizing the adjacency among departments becomes more difficult. Hence, activating the adjacency objective more frequently over the course of the optimization process is important to obtain a Pareto archive with good coverage. The relative weight placed on the adjacency criterion is problem dependent but can be established fairly easily with a few computational tests.

The next step in this work is to use actual retail data and verify the model and the usefulness of its results. To that end, the authors are seeking retail establishments that will share revenue-generating data and other relevant data with them.

References

Barnes, J.W. and Chambers, J.B., 1995. Solving the job shop scheduling problem with tabu search. *IIE Transactions*, 27, 257–263.

Batty, M., 2003. Agent based pedestrian modeling. Working article, Centre for Advanced Spatial Analysis, University College London. Available from http://eprints.ucl.ac.uk/237/

Bellenger, D.N., Robertson, D.H., and Hirschman, E.C., 1978. Impulse buying varies by product. *Journal of Advertising Research*, 18, 15–18.

Brown, W.M. and Tucker, W.T., 1961. Vanishing shelf space. *Atlanta Economic Review*, 9–13.

Corstjens, M. and Doyle, P., 1981. A model for optimizing retail shelf space allocations. *Management Science*, 27, 822–833.

Curhan, R.C., 1973. Shelf space allocation and profit maximization in mass retailing. *Journal of Marketing*, 37, 54–60.

Deb, K., *et al.*, 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6 (2), 182–197.

Dreze, X., Hoch, S.J., and Purk, M.E., 1994. Shelf management and space elasticity. *Journal of Retailing*, 70, 301–326.

Elshafei, A.N., 1977. Hospital layout as a quadratic assignment problem. *Journal of the Operational Research Society*, 28, 167–179.

Farley, J.U. and Ring, W.L., 1966. A stochastic model of supermarket traffic flow. *Operations Research*, 14, 555–567.

Fiore, A.M., Yah, X., and Yoh, E., 2000. Effects of a product display and environmental fragrancing on approach responses and pleasurable experiences. *Psychology and Marketing*, 17, 27–54.

Frank, R.E. and Massy, W.F., 1970. Shelf position and space effects on sales. *Journal of Marketing Research*, 7, 59–66.

Gandibleux, X. and Freville, A., 2000. Tabu search based procedure for solving the 0-1 multi-objective knapsack problem: the two objectives case. *Journal of Heuristics*, 6 (3), 361–383.
Glover, F., 1989. Tabu search—Part I. *ORSA Journal on Computing*, 1, 190–206.
Glover, F., 1990. Tabu search—Part II. *ORSA Journal on Computing*, 2, 4–32.
Glover, F. and Laguna, M., 1997. *Tabu search*. Boston, MA: Kluwer.
Goldberg, D.E. and Lingle, R., 1985. Alleles, loci and the traveling salesman problem. *In*: *Proceedings of the first international conference on genetic algorithms and their applications*, 24–26 July 1985 Pittsburgh. Hillsdale, NJ: Lawrence Erlbaum, 154–159.
Hansen, M.P., 1997. Tabu search for multiobjective optimization: MOTS. *Article presented at the 13th international conference on multi criteria decision making (MCDM'97)*, January Cape Town.
Hansen, M.P., 2000. Tabu search for multiobjective combinatorial optimization: TAMOCO. *Control and Cybernetics* 29, 799–818.
Heragu, S., 1997. *Facilities design*. Boston, MA: PWS.
Hertz, A., *et al.*, 1994. A multi-criteria tabu search approach to cell formation problems in group technology with multiple objectives. *RAIRO–Operations Research* 28, 303–328.
Irion, J., Al-Khayyal, F., and Lu, J-C., 2004. *A piecewise linearization framework for retail shelf space management models.* Available from: http://www.optimization-online.org/DB_FILE/2004/10/967.pdf
Jaeggi, D.M., *et al.*, 2008. The development of a multi-objective tabu search algorithm for continuous optimisation problems. *European Journal of Operational Research*, 185, 1192–1212.
Kollat, D.T. and Willet, R.P., 1967. Customer impulse purchase behavior. *Journal of Marketing Research*, 4, 21–31.
Kulturel-Konak, S., Smith, A.E., and Coit, D.W., 2003. Efficiently solving the redundancy allocation problem using tabu search. *IIE Transactions*, 35, 515–526.
Kulturel-Konak, S., *et al.*, 2004. Exploiting tabu search memory in constrained problems. *INFORMS Journal on Computing*, 16, 241–254.
Kulturel-Konak, S., Smith, A.E., and Norman, B.A., 2006. Multi-objective tabu search using a multinomial probability mass function. *European Journal of Operational Research*, 169 (3), 918–931.
Larson, J.S., Bradlow, E., and Fader, P., 2005. An exploratory look at supermarket shopping paths. *International Journal of Research in Marketing*. Available from: SSRN: http://ssrn.com/abstract=723821
Lee, W., 1961. Space management in retail stores and implications to agriculture. *In*: W.K. Dolva, ed. *Marketing keys to profits in the 1960s*. Chicago, IL: American Marketing Association, 523–533.
Levy, M. and Weitz, B.A., 2001. *Retailing management*. Boston, MA: McGraw-Hill Irwin.
Osman, I.H., 2006. A tabu search procedure based on a random Roulette diversification for the weighted maximal planar graph problem. *Computers & Operations Research*, 33, 2526–2546.
Peters, B.A., Klutke, G.A., and Botsali, A.R., 2004. Research issues in retail facility layout design. *In*: R.D. Meller, *et al.*, eds. *Progress in material handling research: 2004*. Charlotte, NC: Material Handling Institute, 399–414.
Reeves, C.R., ed., 1995. *Modern heuristic techniques for combinatorial problems*. London: McGraw Hill, 104–109.
Schaffer, J.D., 1985. Multiple objective optimization with vector evaluated genetic algorithms. *In: Proceedings of the first international conference on genetic algorithms and their applications*, 24–26 July Pittsburgh. Hillsdale, NJ: Lawrence Erlbaum, 93–100.
Taillard, É.D., 1991. Robust taboo search for the quadratic assignment problem. *Parallel Computing* 17, 443–455.
Taillard, É.D., *et al.*, 1997. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science* 31, 170–186.
Tompkins, J.A., *et al.*, 2003. *Facilities planning*. Hoboken, NJ: John Wiley & Sons.
Tong, X., 1991. *SECOT: a sequential construction technique for facility design*. Doctoral Dissertation. University of Pittsburgh.
Wilkinson, J.B., Mason, J.B., and Paksoy, C.H., 1982. Assessing the impact of short-term supermarket strategy variables. *Journal of Marketing Research*, 19, 72–86.
Yapicioglu, H., 2008. *Retail spatial design with a racetrack aisle network considering revenue and adjacencies.* Doctoral Dissertation, Auburn University.
Yapicioglu, H. and Smith, A.E., forthcoming. Retail spatial design considering revenue and adjacencies using a racetrack aisle network. *IIE Transactions*, accepted for publication.