This article was downloaded by: *[Auburn University]*
On: *28 August 2008*
Access details: *Access Details: [subscription number 791576173]*
Publisher *Taylor & Francis*
Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK

IIE Transactions
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713772245

Determining aisle structures for facility designs using a hierarchy of algorithms

Oguzhan Alagoz [a]; Bryan A. Norman [b]; Alice E. Smith [c]
[a] Department of Industrial and Systems Engineering, University of Wisconsin-Madison, Madison, WI, USA [b] Department of Industrial Engineering, University of Pittsburgh, Pittsburgh, PA, USA [c] Department of Industrial and Systems Engineering, Auburn University, AL, USA

Online Publication Date: 01 November 2008

To cite this Article Alagoz, Oguzhan, Norman, Bryan A. and Smith, Alice E.(2008)'Determining aisle structures for facility designs using a hierarchy of algorithms',IIE Transactions,40:11,1019 — 1031

To link to this Article: DOI: 10.1080/07408170802167621

URL: http://dx.doi.org/10.1080/07408170802167621

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: http://www.informaworld.com/terms-and-conditions-of-access.pdf

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

IIE Transactions (2008) **40**, 1019–1031
Copyright © "IIE"
ISSN: 0740-817X print / 1545-8830 online
DOI: 10.1080/07408170802167621

Determining aisle structures for facility designs using a hierarchy of algorithms

OGUZHAN ALAGOZ[1], BRYAN A. NORMAN[2,*] and ALICE E. SMITH[3]

[1]*Department of Industrial and Systems Engineering, University of Wisconsin-Madison, 3162 Engineering Centers Building, Madison, WI 53706, USA*
E-mail: alagoz@engr.wisc.edu
[2]*Department of Industrial Engineering, University of Pittsburgh, 1048 Benedum Hall, Pittsburgh, PA 15261, USA*
E-mail: banorman@engr.pitt.edu
[3]*Department of Industrial and Systems Engineering, Auburn University, 3301 Shelby Center, Auburn University, AL 36849-5346, USA*
E-mail: smithae@auburn.edu

Received August 2006 and accepted December 2007

This paper introduces a tractable methodology for finding practical aisle structures for a facility for a given flexible bay block layout. The proposed methodology begins with a heuristic to identify candidate horizontal and vertical aisles using user guidance for the general form of the aisles. There then follows an enumeration algorithm that determines the final aisle structure. Using the calculated aisle structure, a non-linear programming model adjusts departmental areas and shapes to accommodate straight aisles. Finally, input/output points are sited using a genetic algorithm. Together, these algorithms specify a reasonable aisle structure and define the material flow through the facility. Two variations of the problem are solved—one with a limit on the total aisle distance and one with a cost per unit aisle length. The effectiveness of the proposed methodology is demonstrated on test problems with 20 and 50 departments.

Keywords: Aisle construction, facility layout design, genetic algorithms, aisle, heuristic

1. Introduction

Facilities design has three interrelated components; structure, layout and the determination of a (network) system to support material flow interaction, e.g., material handling systems (Chhajed *et al.*, 1992). Many techniques for developing layouts have been developed and most focus on minimization of material handling costs—usually in terms of rectilinear distance between departmental centroids (Armour and Buffa, 1963; Tam, 1992; Tretheway and Foote, 1994; Tate and Smith, 1995; Tam and Chan, 1998). These techniques give good results so far as they go, however, their performance may be poor in terms of applicability because the layouts usually ignore aisles. Or, if there are aisles, they include dead ends, short aisles and other characteristics that would make them difficult to physically implement.

In this paper we propose a methodology for generating practical aisle structures, along with their Input/Output (I/O) locations, for a given block layout. The methodology uses a hierarchy of algorithms, each chosen for its specific properties for the optimization subproblem at hand.

2. Literature review

Research concerning I/O station placement in facility design includes Das (1993), who introduced a Mixed-Integer Programming (MIP) approach that locates I/O points in cells assuming that the I/O points must be on the axes of the cells. Rajasekharam *et al.* (1998) adapted the MIP formulation proposed by Das by incorporating a Genetic Algorithm (GA) based two-step heuristic procedure. Other more general cases involving I/O point locations are considered by Montreuil and Ratliff (1988), Banerjee and Zhou (1995), and Kim and Goetschalckx (2005). Arapoglu *et al.* (2001) formulated and compared several heuristic approaches to optimally locate I/Os for each department within a facility layout such that material handling costs are minimized. They used contour distance, which is the distance that is actually traveled between I/Os when going along the perimeters of departments.

---

[*] Corresponding author

None of the previously mentioned research explicitly considers how to construct actual aisle structures within a facility. Certainly by using a distance measure in the objective function, aisles are implied. In the case of rectilinear distance between centroids, the implied aisles are numerous and traverse department interiors. In the case of the contour distance, every path used is an aisle. Because of the computational difficulty in selecting aisles, there are very few studies that have considered the construction of aisles in the design phase. The study of Tretheway and Foote (1994) is an exception. They developed a heuristic able to solve the facility layout problem including aisle location using the scatter diagram developed by Drezner (1987) to arrange the facilities. However, because of computational limitations, the largest problem they could solve had 12 departments. Additionally, their paper does not include a cost analysis of different aisle pattern configurations. Benson and Foote (1997) considered the placement of aisles and I/O locations after the relative locations of the departments and a general aisle structure was selected. However, the aisle structures they proposed include multiple short aisles and aisles with dead ends. Wu and Appleton (2002a) used a slicing tree layout and a GA to design floorplans. The shape, size and the I/O locations for each department were given and fixed prior to the floorplan/aisle design phase. They minimized material handling costs between I/Os using the shortest path and the aisle structure consists of all shortest paths chosen for use. They demonstrated their method on problems ranging from four to 11 departments. In a similar study, Wu and Appleton (2002b) used a simulated annealing algorithm and a slicing tree layout to design a block layout with an aisle structure.

There have also been studies that have considered material flow networks. In the work of Chhajed *et al.* (1992), a flow network was designed for a manufacturing system layout using rectilinear distance between I/Os. They solved the problem optimally and also developed heuristics. However, in their methodology they assumed that the I/O locations are given and their aisle construction does not explicitly consider important design features such as avoiding dead ends and aisles that are too short. Other studies concerning finding material flow paths include the following: Maxwell and Muckstadt (1982), Egbelu and Tanchoco (1986), Gaskins and Tanchoco (1987), Sinriech and Tanchoco (1995, 1997), Liu and Chen (1999), Aiello *et al.* (2002), Kaspi *et al.* (2002) and Ying-Chin and Ping-Fong (2004). These all focus on automated guided vehicle path design. These problems usually assume a given block layout and try to find the shortest route along the aisles along which the materials should flow. Moreover, they generally assume that the objective function is to minimize the total travel distance of the materials transported; therefore, they usually do not consider the properties of the path, which may result in impractical aisle structures. For further coverage of material flow system design please refer to the review articles by Sinriech (1995) and Asef-Vaziri and Laporte (2005).

Our approach is distinct from the previous ones in that it uses a flexible cost of aisles, and requires no specific *a priori* choice of aisle structure or I/O location on the part of the user. The user sets two parameters which guide the selection of aisles relative to minimum aisle length. The method produces practical facility designs with relatively few, relatively straight aisles. Tompkins *et al.* (1996) cite properties of practical aisles as the avoidance of curves, jogs, or non-right-angle intersections. They also affirm that aisles should be straight and lead to doors. Our method adheres to these principles. Our hierarchy of algorithms is quite tractable and we have solved problems of 50 departments, a size which is rarely reached in the facility design literature.

3. Problem description

We consider the problem of constructing an aisle network for a given block layout assuming that all departments have rectangular shapes within a rectangular building. We also assume that each department has a single I/O point and, using the properties of the contour distance and the flexible bay construct (Tate and Smith, 1995) I/Os are located optimally only where a department intersects another department or the facility exterior (see Arapoglu (2000) for proof). Finally, we assume that aisles take no area in the facility (that is, the area taken by aisles $\ll$ the facility area), and that aisles are undirected and uncapacitated. An overview of the proposed methodology is provided in Fig. 1 and details are provided in Section 4.

The objective function is based on the sum of material handling costs and aisle costs and is as follows:

$$\min \sum_{i \in D} \sum_{j \in D} f_{ij} d_{ij} + \sum_{(m,n) \in A} y_{m,n} a_{m,n} C, \tag{1}$$

where D is the set of all departments, A is the set of all arcs in the layout, f_{ij} is the material flow between departments i and j and $y_{m,n}$ is a binary variable that is equal to one if arc (m, n) is used in the final aisle structure and is zero otherwise. In addition d_{ij} is the shortest contour distance using the aisle structure between the output point (O) of department i and the input point (I) of department j. Note that d_{ij} is a function of the $y_{m,n}$ variables, that is, once the decision variables $y_{m,n}$ are determined, d_{ij} can be easily calculated. Also $a_{m,n}$ is the length of arc (m, n) and C is the unit cost per aisle length.

The set of departments and material flows are given. Arcs are all departmental borders (i.e., perimeters) in the layout. The shortest distances between the departments are calculated with respect to the aisle distance, i.e., the distance that the materials actually follow from the output point of one department to the input point of another department. We use two cost formulations. In the first, we assume there is zero cost per unit aisle length ($C = 0$) but there is a limit on the total aisle length. In the second, we assume a unit cost per unit aisle length ($C > 0$), which is included in the

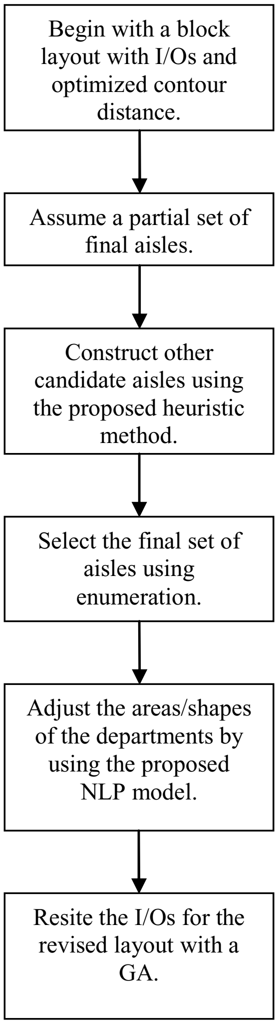


Fig. 1. Flowchart of the algorithm hierarchy.

objective function. Depending on the designer's information and preferences, either could be used with equal ease.

4. Hierarchy of algorithms

We start by finding a set of possible aisles. Figure 2 shows a block layout that has been constructed using the flexible layout methodology of Tate and Smith (1995) and Norman *et al.* (2001). (The numbers arbitrarily label the departments.) We assume that the orientation of the layout is as it is shown in the figure. That is, the boundaries of the bays are assumed to be vertical and the bottom and top boundaries of the layout are horizontal. We do not lose generality with this assumption since we can change the orientation or assume horizontal and vertical aisles are interchangeable.

4.1. *Heuristic for constructing candidate aisles*

The first stage of the hierarchy is a constructive heuristic for choosing candidate horizontal aisles in the facility. The general idea is to examine the layout and identify department borders that are closely aligned and can be adjusted to serve as straight aisles. For example, in Fig. 2 the top borders of departments 8, 19 and 10 are almost collinear and suggest a reasonable horizontal aisle running across the second through fourth bays. Similarly, the top borders of departments 1, 3 and 14 suggest a horizontal aisle. We always assume that any final solution will contain these four aisles: (i) the right-hand side of the first bay; (ii) the left-hand side of the last bay; and (iii) and (iv) the two perimeters at the top and bottom of the facility between the first and last bays. These are shown with solid lines in Fig. 2(a) to indicate presence in the final design. We choose these four aisle segments since including them in the final layout reduces the likelihood that the layout will include dead ends for the other horizontal and vertical aisles. We further assume that all vertical bay boundaries are candidate vertical aisles due to the nature of the flexible bay construct, which inherently results in straight, long vertical aisles. In Fig. 2(a), we have a total of three candidate vertical aisles, which are shown with dashed lines to denote candidate aisles which may or may not be in the final design.

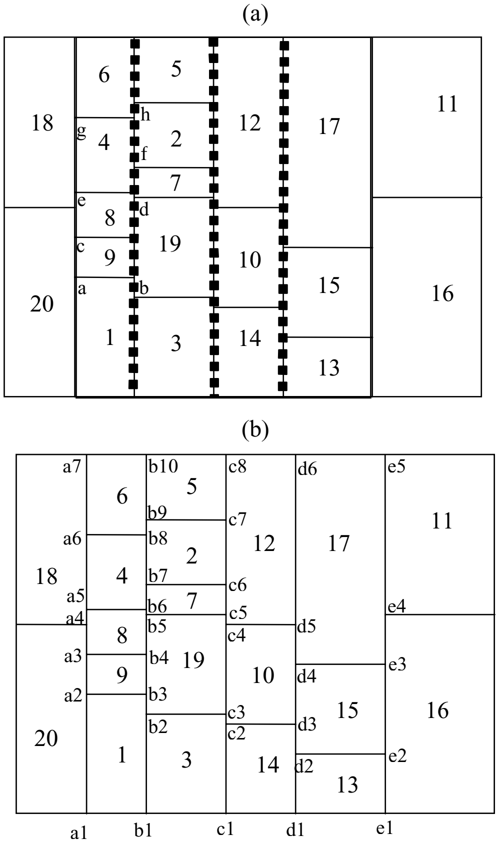


Fig. 2. A flexible bay block layout for the Armour–Buffa 20 department problem.

Next, we group the horizontal arcs in the layout to find candidate horizontal aisles. The notation is given below and the algorithm is presented in Fig. 3.

```
Step 0.
for $i$ =  1 to $B_{2,0}$ do
        $AL_i = B_{2,i}$
        $Avg_i = h_{2,i}$
        $AUD \leftarrow AUD + AL_i$
$k = B_{2,0}$

Step 1.
for $bay$ = 3 to $lastbay$-1 do
        for $j$ = 1 to $B_{\text{bay},0}$ do
                for $i$ = 1 to $k$ do
                        if ($AL_i \in AUD$) and ($used_{i,\text{bay}} = 0$) then
                                select $AL_i$ closest to arc $B_{\text{bay},j}$, let $m = i$
                if ($|Avg_m - h_{\text{bay},j}| < H \times Allow_h$) and ($h_{\text{bay},j+1} > Avg_m$) then
                        add arc $B_{\text{bay},j}$ to $AL_m$
                        $used_{m,\text{bay}} = 1$
                        calculate new $Avg_m$
                else
                        $k \leftarrow k+1$
                        $AL_k = B_{\text{bay},j}$
                        $Avg_k = h_{\text{bay},j}$
                        $used_{k,\text{bay}} = 1$
                        $AUD \leftarrow AUD + AL_k$
                        if ($w_{\text{bay}} + w_{\text{bay}+1} + \ldots + w_{\text{lastbay}-1}$ ) < $WL \times Allow_w$
                                $AUD \leftarrow AUD - AL_k$
        for $i$ = 1 to $k$ do
                if $used_{i,\text{bay}}$ =0 then
                        if $w_i < WL \times Allow_w$ then
                                $AUD \leftarrow AUD - AL_i$
```

Fig. 3. The heuristic for constructing CHAs.

AL_i = potential horizontal aisle i;
k = index of last AL_i;
$B_{i,0}$ = number of arcs in bay i excluding the arcs at the bottom and top of the layout;
$B_{i,j}$ = jth arc in bay i;
$h_{i,j}$ = height of jth arc in ith bay;
w_i = total width of AL_i;
Avg_i = average height of AL_i;
AUD = set of undeleted potential horizontal aisles;
WL = width of the facility excluding right-most and left-most bays;
H = height of the facility;
$Allow_h$ = height allowance, percentage of H, which is used to add an arc to AL_i (user selected parameter);
$Allow_w$ = width allowance, percentage of WL, which is used to make a decision about including potential aisles in the final layout (user selected parameter);

$$used_{i,j} = \begin{cases} 1 \text{ if } AL_i \text{ is used in bay } j, \\ 0 \text{ otherwise.} \end{cases}$$

The algorithm traverses the facility starting from the left side and groups the horizontal arcs that exist at department boundaries to create Candidate Horizontal Aisles (CHAs) considering each arc's proximity to any existing CHAs. In Fig. 2(a), there are four CHAs in the left-most bay, arcs (a, b), (c, d), (e, f) and (g, h). The algorithm begins with the lowest (closest to the bottom edge of the facility) CHA (a, b) and selects the closest horizontal arc (which is the top boundary of department 3) from the set of horizontal arcs in the next bay to the right. The average height (distance from the lower edge of the facility) of each CHA is kept at each step (here, the mean of the heights of (a, b) and top boundary of department 3). If the difference between the average height of a CHA and the height of the closest arc is within a specified range, then that arc is added to that CHA and the average height of that CHA is updated. If there is an arc above the current arc whose height is less than the mean height of the CHA, then we also add that upper arc to the CHA. Two cases can arise while the algorithm is traversing the facility.

Case 1: An arc cannot be added to any of the CHAs that have already been constructed.

In this case, if the total length of the remaining bays is enough to form a new aisle (that is, if ($w_{\text{bay}} + w_{\text{bay}+1} + \cdots + w_{\text{lastbay}-1}) > WL \times Allow_w$), then a new CHA is constructed and that arc is the first element of the CHA.

Case 2: A CHA does not include any arc in the current bay.

In this case, that CHA is checked to see whether it exceeds the pre-specified threshold width, $WL \times Allow_h$, for horizontal aisles. If yes, then it is retained as a candidate aisle. Otherwise, it is deleted. Deletion of CHAs with respect to this threshold value eliminates aisles with short lengths from the final solution, thus giving more practical solutions.

The step-by-step execution of the algorithm for the example problem in Fig. 2(b) is shown below. Note that the letters with indices in Fig. 2(b) are the node numbers (arbitrarily assigned) in the network. $Allow_h$ and $Allow_w$ have been set by the user at the values below.

$$H = 2 \quad Allow_h = 0.15 \quad H \times Allow_h = 0.30$$
$$WL = 1.9 \quad Allow_w = 0.50 \quad WL \times Allow_h = 0.95$$

Step 0:

$$AL_1 = \{(a2, b3)\}\ Avg_1 = 0.6667$$
$$AL_2 = \{(a3, b4)\}\ Avg_2 = 0.8889$$
$$AL_3 = \{(a5, b6)\}\ Avg_3 = 1.1111$$
$$AL_4 = \{(a6, b8)\}\ Avg_4 = 1.5556$$
$$AUD = AL_1,\ AL_2, AL_3, AL_4$$
$$k = 4$$

Step 1: $bay = 3$

$B_{3,1} = (b2, c3)$ and is closest to AL_1 and $used_{1,3} = 0$
so $Avg_1 - h_{(b2,c3)} = 0.1212 < H\times\ Allow_h = 0.30$
and $h_{(b6,c5)} > Avg_1$ so
$AL_1 = \{(\text{a2, b3}), (\text{b2, c3})\}\ Avg_1 = 0.5408$
$used_{1,3} = 1$

$B_{3,2} = (\text{b6, c5})$ and is closest to AL_3 and $used_{3,3} = 0$
so $Avg_3 - h_{(b6,c5)} = 0.0202 < H\times\ Allow_h = 0.30$
and $h_{(b7,c6)} > Avg_3$ so

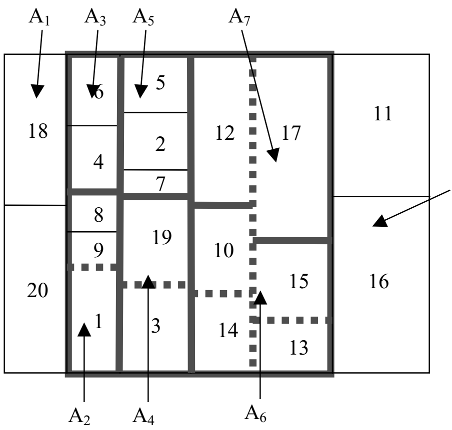


Fig. 4. The layout for the Armour–Buffa 20 department problem after stage one of the hierarchy.

$AL_3 = \{(a3, b4), (b6, c5)\}$ $Avg_3 = 1.1010$ $used_{3,3} = 1$

$B_{3,3} = (b7, c6)$ and is closest to AL_3 but $used_{3,3} = 1$ so we cannot add this arc to AL_3 hence $k = 5$; $AL_5 = \{(b7, c6)\}$ $Avg_5 = 1.2727$ $used_{5,3} = 1$ $AUD = \{AL_1, AL_2, AL_3, AL_4, AL_5\}$ $w_3 + w_4 + w_5 = 1.49 > WL \times Allow_h = 0.95$ and thus the possible new aisle, AL_5, may be long enough after including additional horizontal segments and therefore it is not removed from AUD

$B_{3,4} = (b9,c7)$ and is closest to AL_4 and $used_{4,3} = 0$ so $Avg_4 - h_{(b9,c7)} = 0.0808 < H \times Allow_h = 0.30$ and $h_{(b10,c8)} > Avg_4$ so $AL_4 = \{(a6, b8), (b9, c7)\}$ $Avg_4 = 1.1010$ $used_{4,3} = 1$

Since $used_{2,3} = 0$ and $w_2 = 0.405 < 0.50$, AL_2 is removed from AUD, that is
$AUD = \{AL_1, AL_3, AL_4, AL_5\}$

The algorithm works similarly for $bay = 4$ and 5. The final set of CHAs and Candidate Vertical Aisles (CVAs) is shown by the shaded (used in the final design) and dashed lines (not used in the final design) in Fig. 4. The CHAs are the set of all horizontal aisles that may be included in the final solution. However, some of these aisles or some of the CVAs may be eliminated to meet the constraint on the total aisle length or to minimize the cost of the design. This is determined by the next algorithm in the hierarchy.

4.2. *Enumeration for choosing final aisles*

In this stage we find the set of aisles from the CHAs and CVAs to be deleted by considering the sum of material handling costs and aisle costs, or by imposing a constraint on the total aisle distance. CHAs and CVAs are either kept or deleted as a whole. We made this assumption to make the calculations more tractable and to ensure that we will not have short aisles in the final solution. The search space and search effort are small enough so that we can use an enumerative approach. In order to calculate the effect of deleting or keeping the aisle candidates in different solutions, it is necessary to resite the I/O points of the departments since this will affect the distance that material travels. We use the GA of Arapoglu *et al.* (2001) for reassigning the I/O locations and determining the final material handling costs. Deleted candidate aisles are assigned an infinite distance value. The sections below explain the enumeration under the two alternative formulations to limit aisle length.

4.2.1. *Total Aisle Length Constraint*

Let L be the limit for total aisle length in the facility (user set). Our aim is to find the best combination of aisles to be included in the final solution given that their total length does not exceed L. For each aisle segment, we have two alternatives: include it in the final layout or do not include it. Thus, if there are k candidate aisles then there are 2^k total possible solutions. The enumeration examines at most 2^k solutions, but we can eliminate some of these through feasibility checks and bounding mechanisms. We applied three mechanisms to eliminate some candidates.

1. *Feasibility check*: This check eliminates solutions that are infeasible. Infeasible solutions occur when a department has no access to any aisle. For example, in Fig. 4, if all of the five CVAs and CHAs are deleted, then department 19 does not have access to an aisle.
2. *Total aisle length constraint*: This mechanism checks whether or not a solution exceeds the limit on the total aisle length. If it exceeds L, then that alternative is discarded.
3. *Dominating solutions*: If we assume aisles have zero cost, there is no need to find solutions which have aisle lengths significantly less than L. In other words, assume that the total aisle length is L'. If there is an aisle, which is not included in the current solution, with length less than $L - L'$, then if we include that aisle, we still satisfy the constraint and have an equal or better cost.

4.2.2. *Cost per unit aisle length objective*

In this formulation, the final solution will make a tradeoff between aisle costs and transportation costs. It may be argued that transportation costs are operational costs and aisle costs are strategic, or one time, costs, however, a designer might estimate unit costs appropriate to these two aspects of material flow. In this case, we can only apply the feasibility checking mechanism to the problem since the second and third mechanisms described above do not apply. In this case, there are more alternative solutions and therefore more computations to perform.

4.3. *Non-linear programming model and I/O resiting*

From the enumeration, we know which CHAs and CVAs to keep and which to delete in the final solution, however,

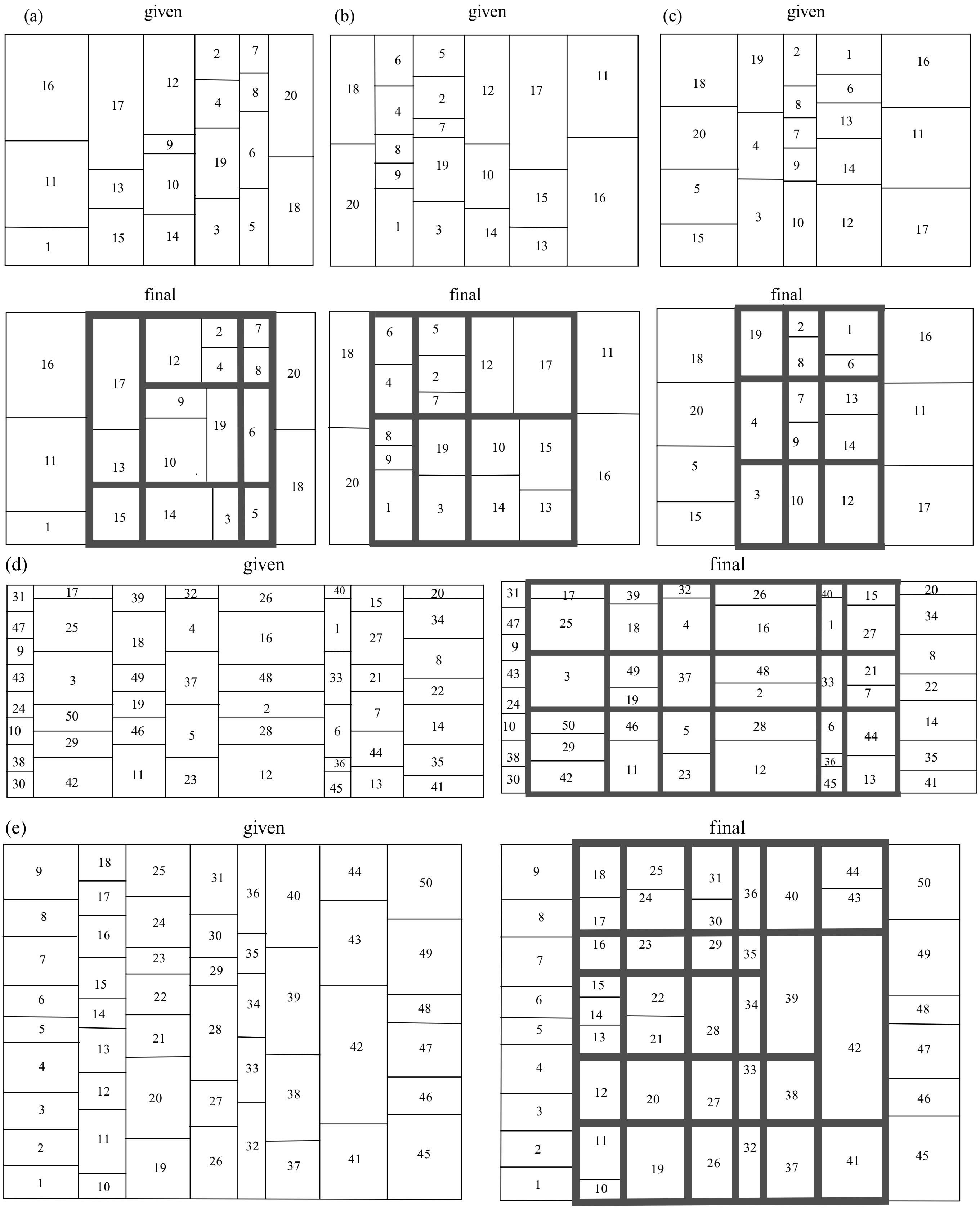


Fig. 5. Original and final layouts for five different problems: (a) AB20_01; (b) AB20_20; (c) AA20_03; (d) AA50_01; and (e) AA50_02.

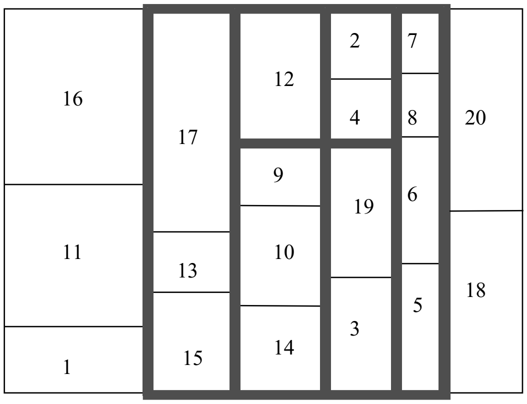
the aisles may not be straight. Our next step, is to adjust the areas and shapes of the departments to correct for this effect. (Note that departmental shape changes that cause violation of aspect ratio constraints are not allowed.) A Non-Linear Programming (NLP) model is used to minimize the maximum area reduction of any department. Although individual departments are adjusted, the bounding rectangle of the facility is not altered. This objective can be changed to minimizing the maximum absolute area deviation or minimizing the total area deviation or any other related objective. However, the minimax objective is preferable because it guarantees restricting the area reduction in all departments. The increase in areas of departments is usually less problematic than area reduction because equipment can always fit into a larger space, but not necessarily into a smaller space.

To apply the NLP approach, the departments within the facility are grouped into "clusters" based on the aisles chosen by the enumeration. The aisles break the facility into regions with all departments in a region defining a cluster. There are eight clusters for the example problem shown in Fig. 4 where the aisles that are solid comprise the final aisle network. The eight clusters of departments are: (18, 20), (4, 6), (1, 8, 9), (2, 5, 7), (3, 19), (12, 17), (10, 13, 14, 15) and (11, 16). The areas of the clusters are used in the NLP approach rather than department areas to simplify the NLP. Note that the percent area change of a cluster as a result of the NLP is identical to that of the departments within the cluster. It can be easily shown that if the area of a cluster is reduced by $\alpha\%$ then to minimize the maximum area reduction of any department within the cluster each department's area needs to be reduced by $\alpha\%$.

Continuing with the example, we need to find the optimal locations for one horizontal and four vertical aisles. Recall that the top and bottom horizontal segments will necessarily be aisles and their locations are known so there is no need to include them in the NLP. In addition, the solid lines on the far left and far right will also necessarily be aisles but their location is not preset. They may move left or right to reduce the maximum area reduction of the departments in the layout.

The notation used for the NLP model is as follows:

I = set of all clusters;
J = set of all CVAs after enumeration;
K = set of all CHAs after enumeration;
x_j = distance between CVA_j and CVA_{j-1}, note x_1 equals the distance between CVA_1 and the left hand side of the layout and $x_{|J|+1}$ equals the distance between $\text{CVA}_{|J|}$ and the right hand side of the layout, ($j = 1, .., 5$ for the example);
nh_k = distance between CHA_k and CHA_{k-1}, note nh_1 is equal to the distance between CHA_1 and the bottom of the layout and $nh_{|K|+1}$ is equal to the distance between $\text{CHA}_{|K|}$ and the top of the layout, ($k = 1, 2, 3$ for the example);



MAR=0.6% cost =38.9% increase

Fig. 6. The effect of changing $Allow_h$ from 0.15 to 0.05 on the aisle structure of the AB20_01 problem ($L = 1.6$).

$A_{i,o}$ = original area of cluster i;
SD_i = set of departments in cluster i;
xd_r = final width of department r;
hd_r = final height of department r;
AR = maximum aspect ratio for all departments (this can easily be relaxed to an aspect ratio constraint for each department);
MAR = maximum area reduction (negative);
H = height of the facility;
W = width of the facility.

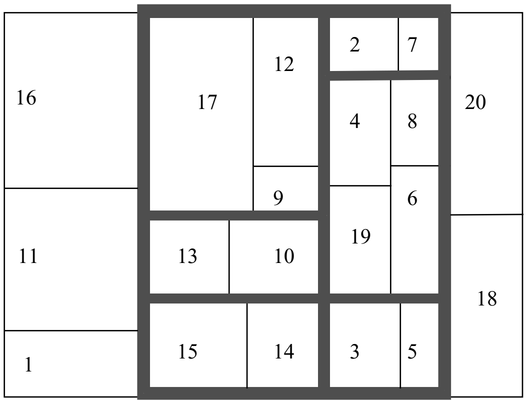
The model is as follows:

max MAR

subject to

$$MAR \leq \frac{nh_k x_j}{A_{i,0}} - 1, \quad \forall i \in I, \tag{2}$$

$$\max\left(\frac{hd_r}{xd_r}, \frac{xd_r}{hd_r}\right) \leq AR, \quad \forall r \in SD_i, \quad i \in I, \tag{3}$$



MAR=3.3% cost =6.7% increase

Fig. 7. The effect of changing $Allow_w$ from 0.50 to 0.30 on the aisle structure of the AB20_01 problem ($L = 1.6$).

$$\sum_{k \in K} nh_k = H, \tag{4}$$

$$\sum_{j \in J} x_j = W, \tag{5}$$

$$nh_k \geq 0, \quad \forall k \in K, \tag{6}$$

$$x_j \geq 0, \quad \forall j \in J. \tag{7}$$

Minimizing the maximum area reduction focuses on finding the least negative (that is, maximum in the numeric sense) area reduction among the departments, hence the max MAR objective. Constraint (2) ensures that the maximum reduction is greater than or equal to the reductions in all clusters. Constraint (3) guarantees that the aspect ratios of the new departments do not exceed the original aspect ratio constraint(s). Note that the hd_r and xd_r variables are proportional to the nh_k and x_j variables with respect to their original areas through direct geometric relationships, but to simplify the formulation and focus on its most important aspects these are not shown explicitly. Constraints (4) and (5) state that the total height and width of the clusters do not exceed the height, H, and width, W, of the facility, respectively. Constraints (6) and (7) are non-negativity constraints.

The specific coordinates of the departments are found by first determining precisely where the aisles are located—which is done by the NLP. Then the relative positions of the departments are preserved. For example, in Fig. 5(a) for the cluster with departments 13 and 17, department 17 remains above department 13. Similarly, in the cluster with departments 2, 4 and 12, department 12 is to the left of departments 2 and 4 and department 2 is above department 4. Departments that were in a separate bay prior to the aisle construction are kept in their bay in the cluster. For example, in Fig. 5(b) departments 10 and 14 are kept in the left half of their cluster while departments 13 and 15 are kept in the right half.

The NLP gives the new heights and widths of the clusters and the departments. For the largest layout problems that we solved (50 departments), the NLP has at most 82 constraints and 11 variables, in other words, a modestly sized model. After revising the department areas using the NLP, we need to resite the I/O points since the areas and the locations of the arcs have been changed. For this purpose, we again use the GA of Arapoglu *et al.* (2001). Alternatively an exact algorithm such as that from Kim and Kim (1999) might be used if the problem size and computational resources allow. At each step of the enumeration (Section 4.2), we run the GA using the original areas and shapes of the departments. If we had to run the GA after the area/shape adjustments are done (earlier in this section), we would have to run the NLP model for each alternative, which would increase the CPU time drastically. Furthermore, as we show in the computational results, running the GA before or after performing the area/shape changes generally gives the same decision (with regard to adding/deleting arcs) because the two are so similar.

5. Computational tests

Several unequal area problems of 20 and 50 departments with flexible bay layouts are solved with the method described in Section 4. The problems are from Armour and Buffa (1963) and Arapoglu *et al.* (2001). The GA settings used for I/O resiting are the same as in Arapoglu *et al.* (2001). All algorithms were coded in C and the computational experiments were conducted on a Pentium III 864 MMX computer.

5.1. *Constraint on total aisle length, L*

Figure 5 shows the before and after layouts for five different problems, three of 20 departments and two of 50 departments. The following values for the aisle algorithm parameters are used: $Allow_w = 0.50$ and $Allow_h = 0.15$. The value for L is found by multiplying the total outer aisle length, *perimeter*, (which is composed of the four aisle segments shown in bold in Fig. 2(a)) with a parameter called *coef*; thus, L equals *perimeter* multiplied by *coef*. The value for L changes slightly for different problems and is indicated in Table 1. The resulting designs are practical with straight, long aisles enabling material flow throughout the facility for most types of material handling systems. The magnitude of the area changes is small in all cases. In practice, the aisle structure selection process would be performed by decision makers examining alternate layouts that result from different values of L, and perhaps $Allow_w$ and $Allow_h$.

The results in Table 1 also show that our methodology can be used to construct good aisle structures while not significantly changing the material handling costs used in the original block layout optimization for the problems considered.

Table 1. The final solutions for the test problems when $Allow_w = 0.50$, $Allow_h = 0.15$

Problem	L *value*	MAR^* (%)	$\frac{FC_b - OC}{OC}$ (%)	$\frac{FC_a - FC_b}{FC_b}$ (%)	$\frac{FC_a - OC}{OC}$ (%)
AB20_01	1.6	2.6	9.4	−2.5	6.7
AB20_20	1.6	4.9	13.7	−6.7	6.1
AB20_03	1.6	3.1	1.3	0.5	1.8
AB20_04	1.8	3.6	0.6	0.0	0.6
AB20_05	1.6	4.1	0.8	5.4	6.3
AA50_01	2.4	8.2	6.2	−4.3	1.7
AA50_02	3.0	14.4	2.4	−3.6	−1.3

*is the maximum percent area reduction of any department (also, any cluster).

OC is the original cost of the given layout.

FC_b is the interim cost calculated after enumeration but before making the area changes by NLP.

FC_a is the final cost calculated after making the area changes by NLP.

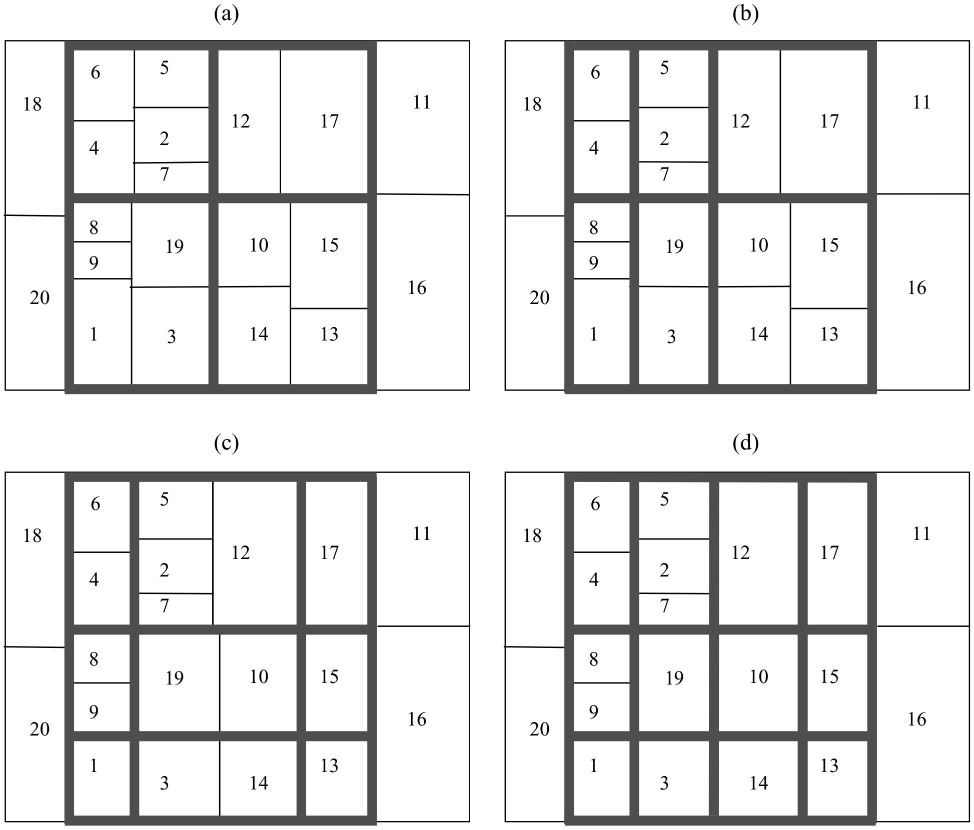


Fig. 8. The effect of changing L on the aisle structure of the AB20_20 problem: (a) $L = 1.4$; (b) $L = 1.6$; (c) $L = 1.8$; and (d) $L \geq 2.0$.

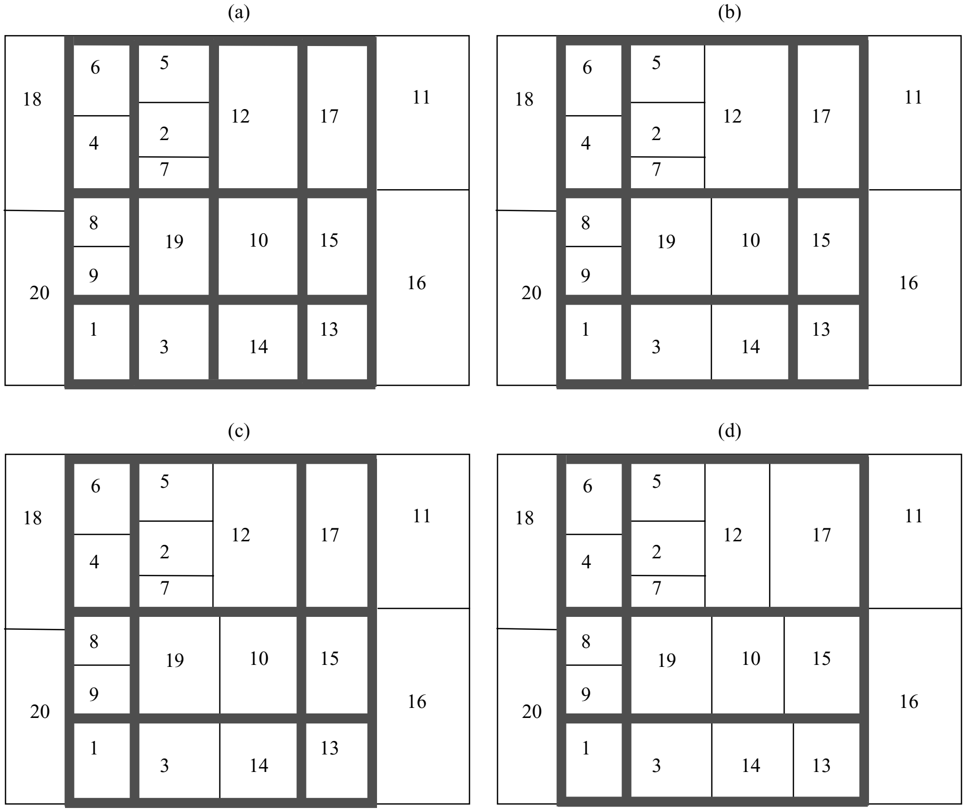


Fig. 9. The effect of changing C on the aisle structure of the AB20_20 problem: (a) $C = 0$; (b) $C = 8$; (c) $C = 16$; and (d) $C = 24$.

Table 2. The effect of change in the value of L on the performance of the enumeration, maximum area deviation and costs ($Allow_w = 0.50$ and $Allow_h = 0.15$)

Problem	*L (coef) value*	*CPU time (min)**	*Total nodes#*	*Percent of nodes calculated*	*MAR (%)*	*Cost change (%)***
AB20_01	1.4	1.7	64	21.9	3.3	20.2
	1.6	2.1	64	28.1	2.6	9.4
	1.8	1.2	64	14.1	6.1	8.0
	2.0	0.9	64	14.1	6.1	8.0
	>2.0	0.3	64	1.6	6.3	3.8
AB20_20	1.4	0.9	32	21.9	4.9	34.6
	1.6	1.1	32	28.1	4.9	13.7
	1.8	0.8	32	15.6	7.6	0.9
	>1.8	0.4	32	3.1	7.8	0.0
AB20_03	1.4	2.1	64	29.7	1.5	3.4
	1.6	1.3	64	18.8	3.1	1.3
	1.8	0.8	64	9.4	3.1	0.6
	>1.8	0.3	64	1.6	4.1	0.0
AB20_04	1.4	1.5	64	18.8	3.2	7.4
	1.6	1.8	64	23.4	3.2	7.4
	1.8	1.6	64	21.9	3.6	0.6
	2.0	0.9	64	9.4	7.5	0.0
	>2.0	0.4	64	1.6	7.5	0.0
AB20_05	1.4	0.6	16	25.0	3.7	2.1
	>1.4	0.3	16	6.3	4.1	1.6
AA50_01	1.4	4.6	512	1.2	0.0	36.5
	1.6	6.4	512	1.8	5.5	17.2
	1.8	14.9	512	5.1	6.1	13.1
	2.0	37.9	512	13.5	7.8	8.0
	2.2	35.1	512	11.7	8.2	7.7
	2.4	25.5	512	9.0	8.2	6.2
	2.6	21.1	512	7.0	10	4.5
	2.8	7.5	512	2.3	10.2	4.7
	>2.8	1.7	512	0.2	11.4	3.4
AA50_02	1.4	1.5	1024	0.1	0.0	47.7
	1.6	6.7	1024	1.2	4.6	26.7
	1.8	20.5	1024	4.0	4.7	15.0
	2.0	55.6	1024	11.6	6.7	12.5
	2.2	76.4	1024	15.9	8.4	8.6
	2.4	67.4	1024	14.3	12.8	7.6
	2.6	37.5	1024	7.8	9.5	6.7
	2.8	9.5	1024	1.9	13.3	4.7
	3.0	5.5	1024	1.0	14.4	2.4

*CPU time includes only enumeration.

**The costs are at the end of enumeration, the change is calculated with respect to the original cost of the given layout.

#Total number of nodes in the enumeration tree.

As shown in Table 1, the differences between the final costs using the optimized aisles and the original costs using all department perimeters as aisles are less than 7%. Another aspect is that the largest cost difference between FC_a and FC_b is less than 7%, which shows that the costs before and after the department area changes are similar. *This is important because it indicates that block layout can be done, using the contour distance metric, prior to aisling, resizing and I/O resiting without significantly sacrificing the overall solution quality*. Additional computational work is needed to verify or dispute whether this is a general property of typical flexbay block layouts and their subsequent aisle design. This also supports the decision to simplify the aisle structure evaluation procedure and not require recalculation of the area changes for each alternative.

5.2. *Effects of altering parameter values*

We next consider the effects of changing the user set parameters on the final designs. If we increase $Allow_h > 0.15$, the results are unchanged. On the other hand, if we set $Allow_h = 0.05$, we obtain fewer and shorter horizontal aisles. This

Table 3. The effect of change in the value of C on the performance of enumeration, maximum area deviation and costs ($Allow_w = 0.50$ and $Allow_h = 0.15$)

Problem	*C value*	*CPU time (min)**	*Total nodes*	*Percent of nodes calculated*	*MAR (%)*	*Final cost***
AB20_01	0	5.1	64	68.8	6.1	358.1
	8	5.1	64	68.8	2.6	473.5
	16	5.1	64	68.8	2.6	569.6
	24	5.1	64	68.8	3.3	650.6
AB20_20	0	2.6	32	71.9	7.8	334.9
	8	2.6	32	71.9	7.6	445.1
	16	2.6	32	71.9	7.6	552.0
	24	2.6	32	71.9	5.4	656.0
AB20_03	0	5.4	64	81.3	4.1	330.5
	8	5.4	64	81.3	1.5	411.0
	16	5.4	64	81.3	1.5	475.8
	24	5.4	64	81.3	1.5	540.6
AB20_04	0	5.4	64	75.0	7.5	340.6
	8	5.4	64	75.0	3.2	445.0
	16	5.4	64	75.0	3.2	524.2
	24	5.4	64	75.0	3.2	603.4
AB20_05	0	1.7	16	93.8	4.1	287.7
	8	1.7	16	93.8	3.7	347.3
	16	1.7	16	93.8	3.7	405.6
	24	1.7	16	93.8	3.7	454.6
AA50_01	0	157.6	512	54.5	11.4	39,418.4
	121	157.6	512	54.5	7.8	52,049.0
	242	157.6	512	54.5	5.5	60,889.5
	363	157.6	512	54.5	5.5	68,875.5
AA50_02	0	233.3	1024	47.7	23.6	40,461.2
	121	233.3	1024	47.7	4.7	60,795.3
	242	233.3	1024	47.7	4.7	75,254.8
	363	233.3	1024	47.7	0.0	88,364.0

*CPU time includes only enumeration.

**The costs are calculated at the end of enumeration and the costs of the aisles are included in the objective function value.

lessens the running time because there are very few horizontal aisle options to consider, reducing the size of the enumeration tree. The effect of decreasing $Allow_h$ on the aisle structure of the AB20_01 problem is shown in Fig. 6. As $Allow_h$ decreases, the number of deleted arcs increases. Therefore, material handling costs increase dramatically due to the circuitous routes that ensue from the limited number of horizontal aisles. On the other hand, the final solution MAR is nominal. Increasing the value of $Allow_w$ above 0.50 also produces fewer horizontal aisles but the change is not as dramatic and the design is not significantly different. However, when we decrease $Allow_w$ from 0.50 to 0.30 we obtain more candidate horizontal aisles and shorter aisles, creating a layout impractical to implement. Additionally, because of the increase in the number of candidate aisles, running time increases. This effect is shown in Fig. 7.

The effect of altering the value of L on the layout of the AB20_20 problem is shown in Fig. 8. As L increases, there are more aisles in the final solution and the magnitude of the maximum percent area reduction increases because there are more clusters, each with smaller area, and more departmental areas are changed.

We next measure the effects of changing L on the performance of the enumeration and the bounding/dominance mechanisms used to eliminate some of the candidate aisles. As shown in Table 2, an increase in L changes the performance of the bounding mechanisms and the corresponding CPU time. While there is not a clear-cut relationship, in general, as L increases the percentage of the total possible nodes that must be evaluated is reduced.

5.3. *Using a unit cost per aisle length, C*

We also considered the same set of problems using the second objective function, minimizing total cost given a unit cost, C, for aisles. Results are shown in Table 3 for different values of C. As shown in Fig. 9, as C increases, the total length of the aisles decreases and therefore, the magnitude of the maximum area reduction decreases. The objective function values increase but this is due to the reduction in aisles, creating greater travel distances, plus the effects of the aisle cost increasing due to the cost per unit aisle changing. Changes in C had no effect on the CPU time because we can apply only the first of the bounding mechanisms in this version of the problem. All of these effects are logical and consistent with what would be expected. A user could run the algorithm multiple times with different values of C to see how the aisle network changes and then choose the final design based on the results.

6. Concluding remarks

In this paper, we have presented a hierarchy of algorithms for finding practical aisle structures for a given flexible bay block layout without significantly changing the areas of the departments or the material flow costs. A heuristic is used to find CHAs and candidate aisles are reduced to the final aisle set through enumeration. The areas and shapes of departments are adjusted to create straight aisles by applying a NLP approach. Finally, I/O points are resited using a previously developed GA. The method, therefore, uses exact optimization where tractable (selection of final aisles from candidates and adjusting department size and shape) and heuristics where necessary (identifying candidate aisles and resiting I/Os). Computational effort is manageable even for large problems. Overall, the CPU times for the different problems range from a few minutes for the 20-department problems to a few hours for the 50-department problems. This includes identifying candidate aisles, selecting optimum aisles, adjusting department areas and resiting I/Os.

The method does depend on a few values set by the designer and we have tested the effects of these values. We envision a designer trying several sets of parameter values to develop a suite of candidate designs. It is also possible for a designer to look at both objectives—adhering to a maximum aisle distance or setting a unit cost per aisle length. In summary, this approach is not designed to act as a black-box tool. Useful interaction with the designer is expected as (s)he explores a variety of designs. At the end, the design selected will be implementable and fairly complete, with departments shaped, sized and located, aisles specified and I/O locations for each department situated. It will also be an optimal or near-optimal design with respect to material handling costs and aisle consideration (either length limit or unit cost). Additionally, the material handling costs will use the actual distance traveled from an output station of one department along the aisle to the input station of the next department in the routing. Given this envisioned use of the approach, the computational effort of the entire design process (after block layout) is probably in the range of half a day to a day. Since facility design is a strategic endeavor that is done infrequently and not under great time constraints, these computational times are well within reason.

Although the examples in this paper are based on a flexible bay representation, the method could be modified for use on a non-flexible bay layout, but in that case the candidate aisle construction heuristic would need to be modified to first create aisles running horizontally or vertically and then to create aisles in the other direction. A potential problem with starting without any vertical aisles is that there is no guarantee that there will be a feasible layout with reasonable aisle structures. In particular, if the user chooses high values for the $Allow_w$ parameter, our method may not be able to generate a sufficient number of aisles to permit flow of materials between all departments.

The current research could be extended by incorporating several modifications. First, penalty costs for half-length aisles can be used so solutions with longer aisles are selected. Second, different objective functions can be used in the NLP model such as minimizing the total area deviation or minimizing the maximum area change rather than minimizing the maximum area reduction. Third, alternative manners to site I/Os other than the GA or siting multiple I/O points for each department could be considered. Finally, deletion of a portion of the horizontal and vertical aisles can be considered, such as having a vertical aisle that runs only halfway through the facility, which may give different solutions. These latter two alterations would increase the search space and computational time fairly significantly. Of course, future work could also consider the aspects of the aisles themselves in terms of area, direction and capacity. These will complicate things significantly, but are still workable within the framework defined in this paper.

Acknowledgements

The authors gratefully acknowledge the support of US National Science Foundation grant DMI 99-08322. The authors wish to thank Dr. Russell Meller as well as two anonymous referees for their suggestions and insights, which have improved the paper.

References

Aiello, G., Enea, M. and Galante, G. (2002) An integrated approach to the facilities and material handling system design. *International Journal of Production Research*, **40**, 4007–4017.

Arapoglu, R.A. (2000) Simultaneous layout design in facility layout. PhD Dissertation, University of Pittsburgh, Pittsburgh, PA.

Arapoglu, R.A., Norman, B.A. and Smith, A.E. (2001) Locating input and output points in facilities design—a comparison of constructive, evolutionary and exact methods. *IEEE Transactions on Evolutionary Computation*, **5**, 192–203.

Armour, G.C. and Buffa, E.S. (1963) A heuristic algorithm and simulation approach to relative location of facilities. *Management Science*, **9**, 294–309.

Asef-Vaziri, A. and Laporte, G. (2005) Loop-based facility planning and material handling. *European Journal of Operational Research*, **164**, 1–11.

Banerjee, P. and Zhou, Y. (1995) Facilities layout design optimization with single loop material flow path configuration. *International Journal of Production Research*, **33**, 183–204.

Benson, B. and Foote, B.L. (1997) Door FAST: a constructive procedure to optimally layout a facility including aisles and door locations based on an aisle flow distance metric. *International Journal of Production Research*, **35**, 1825–1842.

Chhajed, D., Montreuil, B. and Lowe, T.J. (1992) Flow network design for manufacturing systems layout. *European Journal of Operationals Research*, **57**, 145–161.

Das, S.K. (1993) A facility layout model for flexible manufacturing systems. *International Journal of Production Research*, **31**, 279–297.

Drezner, Z. (1987) An heuristic procedure for the layout of a large number of facilities. *Management Science*, **33**, 907–915.

Egbelu, P.J. and Tanchoco, J.M.A. (1986) Potential for bi-directional guide-path for automated guided vehicle based systems. *International Journal of Production Research*, **24**, 1075–1097.

Gaskins, R.J. and Tanchoco, J.M.A. (1987) Flow path design for automated guided vehicle systems. *International Journal of Production Research*, **25**, 667–676.

Kaspi, M., Kesselman, U. and Tanchoco, J.M.A. (2002) Optimal solution for the flow path design problem of a balanced unidirectional AGV system. *International Journal of Production Research*, **40**, 389–401.

Kim, J.G. and Goetschalckx, M. (2005) An integrated approach for the concurrent determination of the block layout and the input and output point locations based on the contour distance. *International Journal of Production Research*, **43**, 2027–2047.

Kim, J.G. and Kim, Y.D. (1999) A branch and bound algorithm for locating input and output points of departments on the block layout. *Journal of the Operational Research Society*, **50**, 517–525.

Liu, F.H.F. and Chen, J.T. (1999) A heuristic approach to determine bidirectional flow-path for AGV systems. *International Journal of Industrial Engineering: Theory, Applications and Practice*, **6**, 171–181.

Maxwell, W.L. and Muckstadt, J.A. (1982) Design of automated guided vehicle systems. *IIE Transactions*, **14**, 114–124.

Montreuil, B. and Ratliff, H.D. (1988) Optimizing the location of input/output stations within facilities layout. *Engineering Costs Production Economics*, **14**, 177–187.

Norman, B.A., Smith, A.E. and Arapoglu, R.A. (2001) Integrated facilities layout using a perimeter distance measure. *IIE Transactions,* **33**, 337–344.

Rajasekharam, M., Peters, B.A. and Yang, T. (1998) A genetic algorithm for facility layout design in flexible manufacturing systems. *International Journal of Production Research*, **36**, 95–110.

Sinriech, D. (1995) Network design models for discrete material flow systems: a literature review. *International Journal of Advanced Manufacturing Technology*, **10**, 277–291.

Sinriech, D. and Tanchoco, J.M.A. (1995) An introduction to the segmented flow approach for discrete material flow systems. *International Journal of Production Research*, **33**, 3381–3410.

Sinriech, D. and Tanchoco, J.M.A. (1997) Design procedures and implementation of the segmented flow topology (SFT) for discrete material flow systems. *IIE Transactions*, **29**, 323–335.

Tam, K.Y. (1992) Genetic algorithms, function optimization, and facility layout design. *European Journal of Operational Research*, **63**, 322–346.

Tam, K.Y. and Chan, S.K. (1998) Solving facility layout problems with geometric constraints using parallel genetic algorithms: experimentation and findings. *International Journal of Production Research*, **36**, 3253–3272.

Tate, D.M. and Smith, A.E. (1995) Unequal-area facility layout by genetic search. *IIE Transactions*, **27**, 465–472.

Tompkins, J.A., White, J.A., Bozer, Y.A., Frazelle, E.H. Tanchoco, J.M.A. and Trevino, J. (1996) *Facilities Planning*. John Wiley, New York, NY.

Tretheway, S.J. and Foote, B.L. (1994) Automatic computation and drawing of facility layouts with logical aisle structures. *International Journal of Production Research*, **32**, 1545–1555.

Wu, Y. and Appleton, E. (2002a) The optimisation of block layout and aisle structure by a genetic algorithm. *Computers & Industrial Engineering*, **41**, 371–387.

Wu, Y. and Appleton, E. (2002b) Integrated design of the block layout and aisle structure by simulated annealing. *International Journal of Production Research*, **40**, 2353–2365.

Ying-Chin, H. and Ping-Fong, H. (2004) A machine-to-loop assignment and layout design methodology for tandem AGV systems with multiple-load vehicles. *International Journal of Production Research*, **42**, 801–832.

Biographies

Oguzhan Alagoz is currently an Assistant Professor of Industrial and Systems Engineering at the University of Wisconsin-Madison. He received his Ph.D. in Industrial Engineering from the University of Pittsburgh in 2004. His research interests include medical decision making, completely and partially observable Markov decision processes, discrete-event system simulation, scheduling and stochastic programming.

Bryan A. Norman is an Associate Professor of Industrial Engineering at the University of Pittsburgh. He received his Ph.D. degree in Industrial and Operations Engineering from the University of Michigan in 1995, where he was a National Science Foundation Fellowship holder, and has B.S.I.E. and M.S.I.E. degrees from the University of Oklahoma. His research interests primarily focus on the modeling of complex problems in manufacturing and production systems and applied optimization. His areas of application include machine and personnel scheduling, job rotation, assembly line balancing, facility layout, material handling system design, energy modeling and radio frequency identification. His research has been funded by several sources including the National Science Foundation and local industry. He has published his research in *IIE Transactions*, *Naval Research Logistics*, *INFORMS Journal on Computing*, the *International Journal of Production Research*, the *European Journal of Operational Research*, the *Annals of Operations Research* and *Computers & Industrial Engineering*. He is a member of IIE and INFORMS.

Alice E. Smith is Professor and Chair of the Industrial and Systems Engineering Department at Auburn University. Previous to this position, she was on the faculty of the Department of Industrial Engineering at the University of Pittsburgh. She has served as a principal investigator on over $3500 000 of sponsored research. She holds one U.S. patent and several international patents and has authored more than 60 publications in journals which have garnered over 500 citations (Web of Science). She won the E. L. Grant Best Paper Award in 1999 and in 2006, and the William A. J. Golomski Best Paper Award in 2002. She holds editorial positions on *INFORMS Journal on Computing*, *Computers & Operations Research*, *International Journal of General Systems*, *IEEE Transactions on Evolutionary Computation* and *IIE Transactions*. She is a fellow of IIE, a senior member of IEEE and SWE, a member of Tau Beta Pi, INFORMS and ASEE, and a Registered Professional Engineer in Industrial Engineering in Alabama and Pennsylvania.