Custom Design of Facility Layouts for Multiproduct Facilities Using Layout Modules

Shahrukh A. Irani and Heng Huang

***Abstract*—The functional, flowline and cellular layouts are traditional facility layouts that have been discussed extensively in the literature. However, due to the large variety of parts usually produced in jobshops, none of these layouts by themselves provide a satisfactory layout solution. This paper introduces a new idea in layout design that decomposes a layout into a network of "layout modules." Each module has a unique material-flow pattern, which, in some cases, corresponds to that of one of the traditional layouts. The use of layout modules allows for a single facility to have different types of layout configurations designed for different portions of its material-flow network. This paper introduces a systematic method for implementation of this design approach and demonstrates its application using a sample of routings obtained from industry.**

***Index Terms*—Cluster analysis, facility layout, jobshop, layout modules, string matching.**

I. INTRODUCTION

THE FUNCTIONAL, flowline, and cellular layouts are traditional facility layouts that have been discussed in the literature and implemented in industry [1]. The majority of facility layout projects focus on "fitting" one of these three traditional layouts to the material-flow network created from the routings of products being produced in a multiproduct facility, such as a jobshop. Few research papers ask the more fundamental question. What is the best layout design for a multiproduct facility that suits the variety of product routings? Could a combination of these layouts be better suited for the facility? This paper proposes a new approach to design of layouts for multiproduct manufacturing facilities—the basic idea is to use each of the three traditional layouts as building blocks that are assembled into a network to custom design a layout for the facility.

II. OUTLINE OF THIS PAPER

This paper is organized as follows. Section III reviews the contemporary literature on: 1) new types of facility layouts beyond the functional and cellular layouts and 2) the use of operation sequences as input data for the design of facility layouts. Section IV introduces the concept of layout modules and the flow pattern characteristic of each module. Section V presents a mathematical formulation of the problem of designing a facility layout as a network of layout modules. Section VI proposes a heuristic solution approach for solving the problem. Section VII presents a case study showing the input and output for the proposed approach, as well as major intermediate results in the solution process. Section VIII describes the advantages of designing a facility layout using layout modules.

III. LITERATURE REVIEW

A review of the literature shows that a fundamental requirement for the design of modern facility layouts is the distribution of identical machines at multiple locations in the facility. As early as 1952, Ireson [2] recognized that a manufacturing facility layout may need to have a combination of the product grouping and process specialization characteristic of cellular and functional layouts, respectively. From a historical perspective, the pioneering paper on production flow analysis by Burbidge [3] was the first to specifically discuss machine duplication to divide and distribute the departments in a functional layout in order to design a cellular layout. Holographic layouts [4] and fractal layouts [5] extend the idea of the traditional functional layout since they distribute identical machines at multiple locations on the factory floor. A related idea, that of giving flexibility to a jobshop layout by distributing identical machines at several nonadjacent locations on the shopfloor, is discussed in [6]. Hybrid cellular layouts [7], [8] represent a fusion of several ideas of partial conversion of a functional layout to a cellular layout, functional grouping of several shared machine types, limited physical duplication of shared machines, and intercell flows. In one variation of hybrid layouts, an existing layout was replaced by a combination of both manufacturing cells and individual workcenters [9]. Cascading cells [10] and remainder cells [11] are other examples of hybrid layouts. Holonic layouts [12] have random arrangements of machines with no specific cell boundaries and distribute multiple machines of any type throughout the facility.

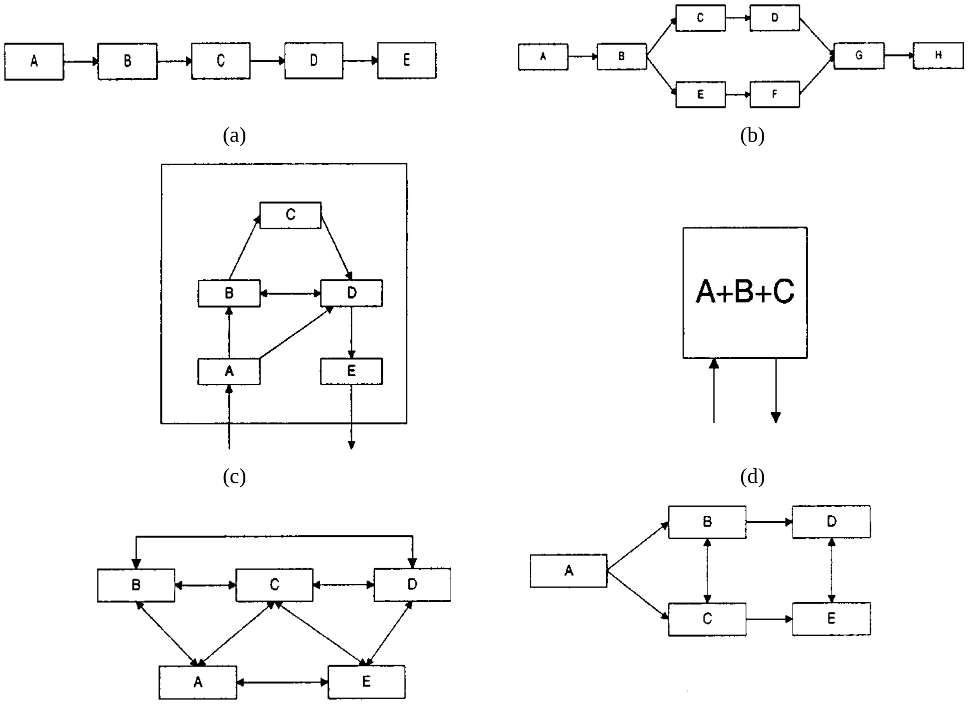
However, the literature also indicates that the strategic duplication of machines in the facility can be achieved only by using the operation sequences of the products as input. If the operation sequences are aggregated and the resulting from–to chart used as input, then at best a single type of layout—functional layout—can be designed for a facility. Vakharia and Wemmerlov [13] presented a method to design a set of independent or interacting flowline cells with minimum intercell flows. Askin and Zhou [14] proposed an enhanced algorithm to solve the same problem using longest common subsequences. Ho *et al.* [15] presented a layout design technique that exploits the similarity of product assembly sequences in a product family

---

Manuscript received November 2, 1998; revised May 25, 1999 and January 26, 2000. This work was supported by the National Science Foundation under Grant DMI-9734815 and Grant DMI-9796034, and by a SEED Grant from Ohio State University. This paper was recommended for publication by Associate Editor P. Banerjee and Editor P. Luh upon evaluation of the reviewers' comments.

The authors are with the Department of Industrial, Welding and Systems Engineering, Ohio State University, Columbus, OH 43210 USA.

Publisher Item Identifier S 1042-296X(00)05060-6.



(a) (b) (c) (d) (e) (f)

Fig. 1. Types of layout modules. (a) Flowline module. (b) Branched flowline module. (c) Cell module. (d) Machining center module. (e) Functional layout module. (f) Patterned flow module.

to design a network-type layout for a multiproduct flowline. Moodie *et al.* [16] discussed the case of a design of a network of manufacturing cells using product sequence similarity analysis where all cells have a flowline layout.

This paper describes a novel idea—design of any facility layout as a network of layout modules—by using a hybrid method that integrates the methods for design of functional and cellular layouts. This method extends a string matching and clustering approach for machine grouping and similarity analysis of product routings into a methodology for facility layout design. The idea of "layout modules" is equivalent to that of "directed graph primitives" which originated during a feasibility study to design a cellular layout for a semiconductor fab [17]. Layout modules automatically group machines that occur together in different operation sequences, allowing for the same machines to be duplicated in several locations, depending on the placement of the modules in the final layout. However, instead of assigning complete routings of products into clusters in order to form cells [13], [14], [18], there is no such requirement in this approach to facility layout. In essence, the layout module expands the ideas of "cells" in a cellular layout and "departments" in a functional layout by allowing a module to have a product, process, or part family focus [19].

IV. LAYOUT MODULES: A NEW CONCEPT IN FACILITY LAYOUT

Experience from several machining and fabrication jobshop layout projects undertaken in industry has shown that the traditional layouts are inadequate for layout design. The term "inadequate" means that the overall material-flow network corresponding to the operation sequences of the products being produced by the facility cannot be reduced to or represented as a planar graph, by any *one* of the traditional layouts. Instead, this paper asserts that the material-flow network in any facility layout can be reduced to a planar graph if it is decomposed into a network of layout modules, with each module representing a portion of the entire facility. A layout module is essentially a group of machines connected by a material-flow network that exhibits a flow pattern characteristic of a specific type of layout, such as the flowline, cellular, or functional layout. Layout modules could be categorized as follows.

- *Flowline Module [Fig. 1(a)]:* A flowline module is a linear arrangement of machines such that all intermachine moves for consecutive pairs of operations on any product moving through the line would be forward, either in sequence or bypass. In case of backtracks, due to multiple nonconsecutive operations on the same machine, a decision could be taken to: 1) modify the linear shape into linear segments with circular/loop segments separating pairs of consecutive linear segments; 2) retain the linear shape but utilize a bidirectional material handling system for backtrack moves; or 3) duplicate the same machine at multiple locations to convert backtrack moves into forward moves.
- *Branched (Convergent/Divergent) Flowline Module [Fig. 1(b)]:* A branched flowline module results when a set of products has operation sequences with one or more substrings of operations common to all of them. At several points, the flowline will split into parallel branches, each branch containing machines unique to a particular product (or products). These branches will merge later into a single line wherever all product(s) require the same substring of operations in their operation sequences.

- *Cell Module [Fig. 1(c)]:* Similar to a flowline module, a cell module is a set of dissimilar machines which, if placed together, could produce a family of parts or products without the products requiring to visit any additional departments or machines external to the module. Although the parts in a family may not use all the machines and/or have the same sequence of operations, their operation sequences have high commonality of machine requirements and high similarity of operation sequences.
- *Machining Center Module [Fig. 1(d)]:* A machining center module is a variation of the cell module. It is a single multifunction automated machine that combines the different manufacturing processes available on multiple machines in a traditional multimachine cell module. This type of module is capable of producing a family of parts (or products) with the added advantage of eliminating the intermachine handling and setup delays in a traditional cell module that would require the products to complete a multistep sequence of operations on single-function machines.
- *Functional Layout Module [Fig. 1(e)]:* A functional layout module is analogous to the process-focused department in a traditional functional layout in which material flows are random. The random flows are due to the absence of any flow dominance or patterns in the sequences in which the different machines within the module are used by different parts.
- *Patterned Flow Module [Fig. 1(f)]:* The material-flow network in a patterned flow module exhibits a flow dominance and precedence hierarchy, characteristic of a directed acyclic digraph (DAG). This module could be further decomposed into a network of flowline modules and branched flowline modules.

V. DESIGN OF A MODULAR LAYOUT

The following nonlinear integer program is a formulation of the problem of designing a facility layout as a network of layout modules:

$$\text{minimize} \sum_{k=1}^{K} E_k P_k + \sum_{l=1}^{L} \sum_{m=1}^{l-1} \sum_{i=1}^{Q} C_{ilm} y_{ilm} \tag{1}$$

subject to

$$\sum_{l=1}^{L} x_{ijl} = 1, \qquad \text{for each } (i,\, j) \tag{2}$$

$$\sum_{i=1}^{Q} \sum_{j=1}^{N_i} q_i F_{ijk} x_{ijl} \le T_k n_{kl}, \qquad \text{for each } (k,\, l) \tag{3}$$

$$\sum_{l=1}^{L} n_{kl} = M_k + P_k, \qquad \text{for each } k \tag{4}$$

$$y_{ilm} = \sum_{j=1}^{N_i - 1} \left\lceil \frac{q_i}{B_i} \right\rceil x_{ijl} x_{i(j+1)m}, \quad \text{for } l \in \{1, \cdots, L\};\ m \in \{1, \cdots, L\};\ i \in \{1, \cdots, Q\} \tag{5}$$

$$x_{ijl} = \{0, 1\}, \qquad \text{for each } (i,\, j,\, l)$$

$$n_{kl} \ge 0, \qquad \text{for each } (k,\, l)$$

$\lceil z \rceil$ denotes the smallest number that is larger than or equal to z.

Variables:

n_{kl}	number of machines of type k assigned to module l;
P_k	extra number of machines of type k required to be purchased;
x_{ijl}	$= 1$ if the jth operation of product i is performed in module l $= 0$ otherwise;
y_{ilm}	annual number of trips for transporting product i between modules l and m.

Parameters:

B_i	batch quantity of product i;
C_{ilm}	cost per trip of transporting product i between modules l and m;
E_k	annual cost incurred by purchasing an extra machine of type k;
F_{ijk}	time required to perform the jth operation of a unit of product i on a machine of type k;
K	number of machine types;
L	number of modules in the layout;
M_k	existing number of machines of type k;
N_i	number of operations in the routing of product i;
Q	number of products;
q_i	annual production quantity of product i;
T_k	annual available time on a machine of type k.

The objective function incorporates the standard material handling costs for travel between modules and the costs of purchasing extra machines to increase available capacity on any machine type in any module. Constraint set (2) ensures that each operation in the routing of each product is done in any one module. The implicit assumption is that a particular operation does not have alternative machines on which it could be performed. Constraint sets (3) and (4) ensure that the integer allocation of machines of the same type to one or more modules is constrained by the available number of machines of that type on the shop floor, i.e., acquisition of extra machines incurs a capital expense. Constraint set (5) is core to the problem—for each pair of consecutive operations in the routing of a product, we need to track the movement of the product between the two modules in which the operations are performed. We need to check if both these operations were performed in the same module, or not. If not, then that would result in an intermodule trip that implicitly incurs work-in-process and material handling costs. The ideal solution would be to completely process each product in a dedicated flowline module. Since that would entail significant investment in extra machines, the practical strategy would be to maximize the number of consecutive operations in each routing that are performed within the same module.

A study of the above formulation shows that the problem of designing a facility layout as a network of layout modules consists of several NP-complete subproblems. For example, determining the size and number of each layout module that will comprise the new layout is analogous to the clustering, capacitated clustering, graph partitioning, and partition into cliques problems [20], [21]. Given the digraph connecting the set of machines assigned to a module, determining which of the layout modules in Fig. 1 it best represents is the subgraph isomorphism problem [20]. For example, the simplest case where the population of routings is clustered into K modules with the same configuration—flowline—is analogous to the shortest common superstring problem [20]. Having obtained a decomposition of the existing facility into a network of modules, determining the optimal layout for this network is analogous to the quadratic assignment problem or the maximum weight planar subgraph problem [20] of designing a block (or functional) layout for a facility.

It is unlikely that we will be able to solve such a complex problem in an efficient way. Consequently, we made several simplifications in the model without losing the important underlying structure. For example, we ignored the detailed layout-related costs and eliminated the need to design an actual block layout with aisle structure. The heuristic procedure developed to solve this simplified problem is described next.

VI. HEURISTIC SOLUTION PROCEDURE FOR DESIGN OF MODULAR LAYOUTS

The input data required for a facility layout study is usually comprised of the following: 1) a set of products; 2) an operation sequence for each product; and 3) a production quantity for each product. The strategy for generating layout modules is based on the methods of string matching and clustering used extensively in genetics, molecular chemistry, and the biological sciences [22]. We first define a common_substring and a residual_substring as follows.

Common_substring: substring of consecutive operations that is common to two or more operation sequences.

Residual_substring: the remaining substring(s) of operations in an operation sequence after the common_substring(s) are extracted from it.

For example, between operation sequences $X = (A, B, C, D, E, F)$ and $Y = (A, B, G, H, E, F)$, the common_substrings are (A, B) and (E, F), while the residual_substrings in X and Y are (C, D) and (G, H), respectively.

The heuristic solution procedure for custom design of any facility layout as a network of layout modules is described next.

Stage 1: Find common_substrings, if any, between all pairs of operation sequences. The algorithm for comparing two operation sequences $X = (x_1, x_2, \cdots x_m)$ and $Y = (y_1, y_2, \cdots, y_n)$ and finding all the common_substrings in them is illustrated in Fig. 2. The algorithm runs in $O(mn)$.

Stage 2: Measure similarity between each pair of unique common_substrings. Find all unique common_substrings in the set of common_substrings discovered in Stage 1. Compute similarity measures for all pairs of unique common_substrings. The measures commonly used for string grouping are numerical measures of similarity or dissimilarity (distance). The majority of them, such as the Levenshtein distance [23], seek only the dissimilarities between two strings ignoring the common_substrings. These techniques, however, are not efficient in situations where the numbers of operations in two sequences differ significantly. In such cases, the matching segments in the two sequences are overridden by the unmatched residual segments in the longer sequence. In particular, one sequence could be completely contained in another sequence, and yet the two sequences could be dissimilar according to the Levenshtein string edit distance measure. Consider the following operation sequences: $O_1 = (A, J)$, $O_2 = (A, K)$, $O_3 = (A, J, B, C, D, E)$, $O_4 = (A, F, G, H, I, J)$.

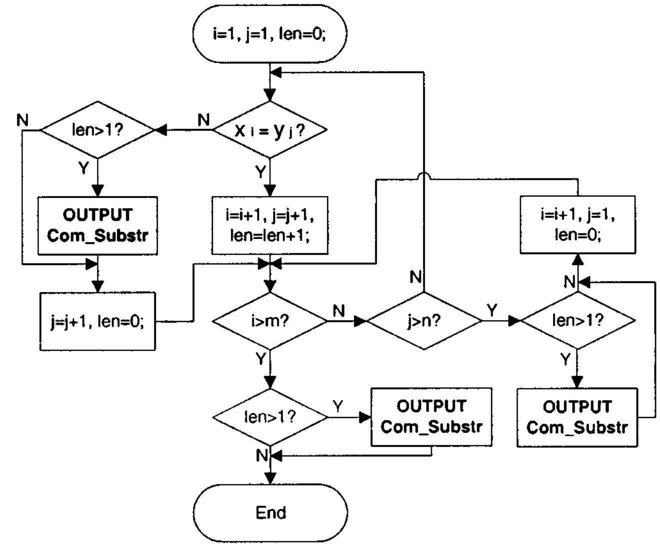


Fig. 2. Algorithm for finding common_substrings in two operation sequences.

If using the Levenshtein measure, the distance between O_1 and O_2 is 1, the distance between O_1 and O_3 is 4, and the distance between O_1 and O_4 is 4. Accordingly, O_1 is more similar to O_2 than to O_3 and O_4. O_1 would therefore be grouped with O_2 but not with O_3 or O_4, although O_1 is a substring of O_3 and a subsequence of O_4. This grouping of operation sequences could result in unnecessary duplication of machines between clusters.

Next, Askin and Zhou [14] proposed a similarity coefficient based on the longest common subsequence (LCS) between two operation sequences. This measure eliminates the unintended influence of long unmatched residual subsequences. However, as discussed in [22], no subsequence measure can provide substring information. The LCS-based measure focuses on the order of operations in the sequences but ignores the gaps between matching operations in the sequences. This could result in grouping of parts that will experience significant bypass travel distance between consecutive operations in their routings if they are produced on the same flowline. For example, using Askin and Zhou's similarity measure, the similarity between O_1 and O_3 is equal to the similarity between O_1 and O_4, which means O_1 and O_4 may be grouped together although the travel distance from operation A to operation B will increase significantly if the two parts are produced on the same flowline.

To avoid such undesirable clustering due to the numerical similarity/dissimilarity measures discussed earlier, we present a new similarity measure for string clustering called merger coefficient. This measure evaluates the feasibility of merging or absorbing one operation sequence completely into another operation sequence. In order to calculate the merger coefficient between two operation sequences, the merger distance and the interruption distance need to be defined first. In terms of the three types of transformations for deriving one operation sequence from another operation sequence—substitution, deletion, and insertion [18]—the merger distance for the absorption of sequence x into sequence y, denoted by $md(x, y)$, is defined as the smallest number of substitutions and insertions required to derive x from y. Keeping $md(x, y)$ fixed, the smallest number of deletions required between two consecutive basic transformations, between two consecutive matching operations, and between two consecutive transformation and matching operation, is defined as the interruption distance for the absorption of x into y, denoted by $id(x, y)$. For example, the merger distance and the interruption distance for absorbing O_1 into O_4 are 0 and 6, respectively.

Once the merger distances and interruption distances between any two operation sequences x and y have been identified, the merger coefficient between x and y, denoted by $mc(x, y)$, can be calculated as follows:

$$
\begin{aligned}
&mc(x, y)\\
&\quad = mc(y, x)\\
&\quad = \begin{cases}
1 - \dfrac{md(y,x) + \frac{id(y,x)}{N_x}}{N_y + 1}, & \text{if } N_x > N_y\\[2ex]
1 - \dfrac{md(x,y) + \frac{id(x,y)}{N_y}}{N_x + 1}, & \text{if } N_x < N_y\\[2ex]
\max\left(1 - \dfrac{md(x,y) + \frac{id(x,y)}{N_y}}{N_x + 1},\right. & \\[2ex]
\qquad\quad \left. 1 - \dfrac{md(y,x) + \frac{id(y,x)}{N_x}}{N_y + 1}\right), & \text{if } N_x = N_y
\end{cases}
\end{aligned}
$$

where N_x and N_y represent the number of operations in sequences x and y, respectively. Note that $O < mc(x, y) \leq 1$. The higher the merger coefficient between two operation sequences, the more similar are they. The computation of the merger coefficient runs in $O(N_x N_y)$ [24]. Unlike the Levenshtein distance and Askin and Zhou's LCS-based similarity coefficient, the merger coefficient is the only measure that does an exact comparison of strings to detect common_substrings of operations. Table I compares the three measures for their ability to measure the exact substring similarity/dissimilarity between sequence O_1 and sequences O_2, O_3, and O_4.

TABLE I
COMPARISON OF THE THREE SIMILARITY/DISSIMILARITY MEASURES

	O_2	O_3	O_4	
O_1	1	4	4	Levenshtein Distance [23]
	0.5	1	1	LCS-based Similarity Coefficient [14]
	0.67	1	0.75	Merger Coefficient

Stage 3: Generate basic layout modules (flowline, branched flowline, and patterned flow modules). From the unique common_substrings, remove any substring that is completely contained in another substring, i.e., the shorter one of a pair of substrings for which the merger coefficient is one. Perform a cluster analysis of the remaining common_substrings to group similar common_substrings into several clusters using the merger coefficients. Since the cluster analysis problem is an NP-complete problem, a variety of heuristics such as single linkage cluster analysis, average linkage cluster analysis, and complete linkage analysis have been used to solve it [25]. In prior studies by the authors relating to cellular manufacturing and facility layout, average linkage cluster analysis was found to consistently extract the desired clusters from the test data [26]. Based on the clusters of common_substrings, generate a flowline, branched flowline, or patterned flow module to represent each cluster of common_substrings. Each cluster of common_substrings is then assimilated into a directed graph (digraph). The criterion for merging two clusters of common_substrings is that the resulting digraph generated from the combined cluster must be acyclic [27]. A directed acyclic graph (DAG) exhibits no cycles or strongly connected components consisting of more than two nodes—this is characteristic of the material-flow networks in the flowline, branched flowline, and patterned flow modules. Strongly connected components in a directed digraph can be found in $O(m + n)$, where m and n are the number of edges and vertices, respectively [28]. The common_substrings that do not get clustered in this stage will be retained as flowline modules. The algorithm for clustering of common_substrings and generating basic layout modules is described next.

Given T as the number of all operation types and N as the number of common_substrings, a common_substring can be represented by a $T \times T$ matrix A, where

$$
A[i, j] = \begin{cases} 1, & \text{if there is a flow from operation } i \text{ to operation } j\\ 0, & \text{otherwise.} \end{cases}
$$

Representing each common_substring by a $T \times T$ matrix $A_k(k = 1, 8, N)$, the merger of two or more substrings can then be represented by the matrix $\sum_k A_k$, where

$$
\left(\sum_k A_k\right)[i, j] = \begin{cases} a > 0, & \text{if there is a flow from operation } i \text{ to operation } j\\ 0, & \text{otherwise.} \end{cases}
$$

The following procedure for forming layout modules by aggregating a cluster of common_substrings is described below.

1) Let each common_substring A_k be a cluster, which will result in the set of clusters $\mathbf{C} = \{C_k\} = \{A_k\}$, $k = 1, \cdots, N$.
2) Find all strongly connected components consisting of more than two operations, if any, contained in each common_substring $A_k(k = 1, \cdots, N)$ and store them in a corresponding set of strongly connected component $\boldsymbol{S_k}$. The strongly connected components in $\boldsymbol{S_k}$ are allowed to occur in a basic layout module generated from A_k.

3) Do until no clusters can be merged:
    - Step 1: Mark every pair of clusters as untested.
    - Step 2: Calculate similarity for each pair of clusters using the average-linkage method [25].
    - Step 3: From the untested clusters, select the pair C_{k1} and C_{k2} that has the highest similarity.
    - Step 4: Test mergeability of C_{k1} and C_{k2}. If $(C_{k1}+C_{k2})$ contains strongly connected components that consist of more than two operations and do not belong to $(\boldsymbol{S_{k1}} \cup \boldsymbol{S_{k2}})$, C_{k1} and C_{k2} are not mergeable, go to Step 3; else, C_{k1} and C_{k2} are mergeable, replace C_{k1} and C_{k2} by $(C_{k1}+C_{k2})$ in cluster set $\boldsymbol{C}$ with $(\boldsymbol{S_{k1}} \cup \boldsymbol{S_{k2}})$ as its corresponding strongly connected component set, mark $(C_{k1}+C_{k2})$ as untested, go to Step 2.

4) Output each of the clusters in C as a layout module.

Stage 4: Generate functional layout modules and machining center modules, if necessary. If two layout modules have many common operations, they may be merged into a more complex module (a function layout module or machining center module) to reduce machine duplication. The commonality between layout modules M_i and M_j is defined as

$$\frac{n_{ij}}{\min(n_i,\, n_j)}$$

where n_{ij} is the number of distinct operations common to both modules; n_i and n_j are the number of distinct operations contained in M_i and M_j, respectively. Given a user-defined threshold level of commonality $V(0 \leq V \leq 1)$ for merging layout modules, the algorithm for merging layout modules is presented as follows.

1) Calculate the commonality between each pair of layout modules.
2) Find the pair of layout modules with highest commonality. If the commonality is higher than the threshold level V, then aggregate the two modules into one, go to (1); else, stop.

The selection of the threshold level is a specific decision problem that requires the user to perform the classical tradeoff between intermodule material-flow costs and machine duplication among the modules to eliminate the flows [29].

Stage 5: Express the original operation sequences in terms of the layout modules and/or residual machines. Given an operation sequence $(x_1, x_2, \cdots, x_m)$ and layout modules $M_1, \cdots, M_n$:

1) let $i = 1$; create a null operation sequence as the new sequence;
2) if $i \leq m$, find the layout module M_j which contains the longest substring that matches $(x_i, x_{i+1}, \cdots, x_k)$, where $k \leq m$; else, go to (4);
3) if $k \leq 1$, then put x_i as a residual machine at the end of the new sequence, $i = i + 1$, go to (2); else, put M_j at the end of the new sequence, $i = k + 1$, go to (2);
4) output the new sequence as a "new" version of the original operation sequence.

Based on the "new" versions of the original operation sequences, a digraph representation (or from–to Chart) between layout modules and residual machines in the facility layout can be generated. If one or more layout modules and/or residual machines exhibit no flow from/to other modules and machines, then they could be merged into a cell module.

TABLE II
ROUTINGS OF PARTS

Routing #	Part #	Operation Sequence
1	1	1,2,3,4,5,6,7,8,9,10
2	2	1,2,3,11,4,8,10
3	3	12,2,13,3,2,9,10
4	4	12,2,6,3,10
5	5	12,6,2,3,2,4,10
6	6	1,2,8,9,2,4,10
7	7	2,3,5,4,6,7,6,7,10
8	8	2,3,5,4,6,10
9	9	1,2,14,4,5,6,9,10
10	10	1,2,3,4,5,6,7,10
11	11	12,2,3,9,10
12	12	1,2,13,3,6,5,9,10
13	13	1,2,3,5,4,8,6,8,10
14	14	12,2,3,5,6,2,10
15	15, 16	1,2,3,4,5,8,6,5,7,10
16	17, 24	12,2,3,10
17	18	1,2,3,5,6,10
18	19	12,2,3,5,6,9,10
19	20	12,2,3,8,10
20	21	1,2,3,4,5,6,7,5,10
21	22	1,2,5,6,4,9,10
22	23	12,2,10
23	25	12,2,3,5,4,6,9,10

TABLE III
COMMON_SUBSTRINGS SUITABLE FOR CLUSTER ANALYSIS

No.	Common-Substrings
S1	8,9
S2	1,2,3,4,5,6,7
S3	4,8
S4	8,10
S5	3,2
S6	2,13,3
S7	3,10
S8	2,4,10
S9	6,2
S10	2,3,5,4,6
S11	6,7,10
S12	6,10
S13	5,6,9,10
S14	6,5
S15	8,6
S16	1,2,3,5
S17	12,2,3,5,6
S18	2,10

VII. CASE STUDY

The proposed method for generating modular layouts was tested using data obtained from a local sheet metal fabrication jobshop. The operation sequences of the parts are listed in Table II. After finding all the unique common_substrings

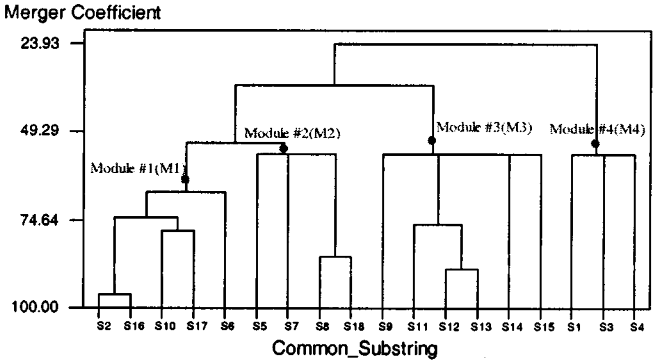
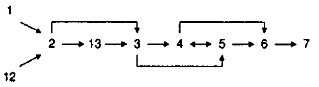
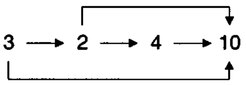
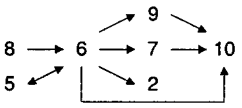
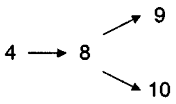


Fig. 3. Dendrogram for average linkage cluster analysis of the common_substrings.

TABLE IV
BASIC LAYOUT MODULES

Module #	Cluster of Common_ Substrings	Acyclic Digraph for the Layout Module
M1	S2, S16, S10, S17, S6	
M2	S5, S7, S8, S18	
M3	S9, S11, S12, S13, S14, S15	
M4	S1, S3, S4	

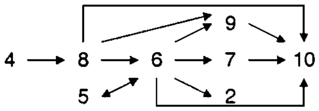


Fig. 4. Digraph for module 5 ($M5$).

TABLE V
REEXPRESSION OF THE OPERATION SEQUENCES USING LAYOUT MODULES

Routing #	Operation Sequence	Module Sequence
1	(1,2,3,4,5,6,7),(8,9,10)	M2 → M5
2	(1,2,3),11,(4,8,10)	M2 → 11 → M5
3	(12,2,13,3),2,(9,10)	M2 → 2 → M5
4	(12,2),6,(3,10)	M2 → 6 → M1
5	12,(6,2),(3,2,4,10)	12 → M5 → M1
6	(1,2),(8,9),(2,4,10)	M2 → M5 → M1
7	(2,3,5,4,6,7),(6,7,10)	M2 → M5
8	(2,3,5,4,6),10	M2 → 10
9	(1,2),14,(4,5),(6,9,10)	M2 → 14 → M2 → M5
10	(1,2,3,4,5,6,7),10	M2 → 10
11	(12,2,3),(9,10)	M2 → M5
12	(1,2,13,3),(6,5),(9,10)	M2 → M5 → M5
13	(1,2,3,5,4),(8,6),(8,10)	M2 → M5 → M5
14	(12,2,3,5,6),(2,10)	M2 → M1
15	(1,2,3,4,5),(8,6,5),(7,10)	M2 → M5 → M5
16	(1,2,3,4,5),(8,6,5),(7,10)	M2 → M5 → M5
17	(12,2,3),10	M2 → 10
18	(1,2,3,5,6),10	M2 → 10
19	(12,2,3,5,6),(9,10)	M2 → M5
20	(12,2,3),(8,10)	M2 → M5
21	(1,2,3,4,5,6,7),5,10	M2 → 5 → 10
22	(1,2),(5,6),4,(9,10)	M2 → M2 → 4 → M5
23	(12,2),10	M2 → 10
24	(12,2,3),10	M2 → 10
25	(12,2,3,5,4,6),(9,10)	M2 → M5

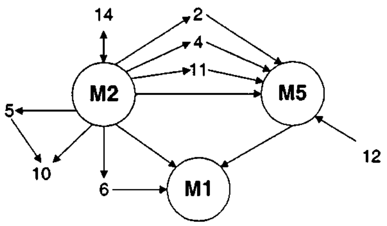


Fig. 5. Digraph representation of the material flows in the facility.

between all pairs of operation sequences and removing the substrings that could be completely absorbed into other substrings, the common_substrings suitable for grouping are shown in Table III. Fig. 3 shows the average linkage cluster analysis dendrogram for grouping the common_substrings using merger coefficients. Table IV shows the four basic layout modules generated from the cluster analysis. A threshold level of commonality $V = 0.7$ was chosen for merging the layout modules. Since the commonality between $M3$ and $M4$ is $0.75 > V$, $M3$ and $M4$ are merged into a new module named $M5$. Fig. 4 shows the digraph for $M5$. The final layout modules obtained after merging modules are $M1$, $M2$, and $M5$. Table V shows the reexpression of the original operation sequences in terms of these three modules and residual machines. Based on Table V, a digraph representation of the material flows in the modular layout for the facility is shown in Fig. 5. If a detailed block layout is desired for the facility, the from–to chart in Fig. 5 would be input to a standard layout algorithm [30].

VIII. ADVANTAGES OF THE PROPOSED APPROACH

This paper proposed a generic approach for design of any facility layout using operation sequences. It extends current thinking on input data requirements and methods for facility layout. In addition, it adds a new type of layout—modular layout—to a new generation of facility layouts that are beyond the three traditional layouts that continue to be studied or implemented in industry. Compared to the traditional layouts, modular layouts have more flexibility for designing a layout to suit the material flows specific to the facility since they identify various flow patterns for different portions of the facility. For example, given two operation sequences (A, B, C, D, E) and (A, B, F, G, H), a flowline (A, B, C, D, E, F, G, H) could be formed using Askin and Zhou's method [14]. Here, subsequence (C, D, E) is forced to precede subsequence (F, G, H), although there is no flow between them. When implementing this flowline, (A, B, C, D, E) and (C, D, E, F, G) are indifferent, i.e., the machine adjacencies in the original operation sequences are missing and superfluous flows are introduced in the flowline. However, by applying the concept of layout modules to the same pair of operation sequences, a branched flowline will be formed that will not have superfluous flows, because two flowlines corresponding to the strings (C, D, E) and (F, G, H) would be attached in parallel to the string (A, B).

Unlike traditional layouts, modular layouts do not assume flow patterns (flowlines) or criteria for machine grouping (process departments or cells). Instead, the flow patterns and machine groupings are determined by the flows implicit to the operation sequences, whereby various types of practical layouts can be designed. However, a modular layout could become a traditional layout due to the flow characteristics of the facility. This is reflected in the following scenarios addressed by the proposed design methodology.

- When only one layout module is generated after Stage 3, the final layout is a flowline, branched flowline, or patterned flowline layout.
- When more than one layout module is generated after Stage 2 but all of them are merged in Stage 4, the final layout would be a functional layout.
- When all layout modules are cell modules, the final layout could be a cellular or hybrid cellular layout.

IX. CONCLUSION

This paper proposed a new methodology for design of facility layouts for multiproduct facilities. The methods discussed in this paper extend the state of the art in the theory and practice of facility layout techniques by using operation sequences, not from–to charts, as input data. The underlying concept for the methodology is that the actual material-flow network for a facility can be replaced by an equivalent network of different layout modules, where each layout module is equivalent to one of the traditional layouts that have a well-recognized flow pattern. Since layout modules are developed based on an analysis of operation sequences, this new approach to facility layout would allow the facilities planner to customize the layout for any facility based on the unique composition of the product mix processed in that facility.

REFERENCES

[1] J. Tompkins and J. White, *Facilities Planning*. New York: Wiley, 1984.

[2] W. G. Ireson, *Factory Planning and Plant Layout*. Englewood Cliffs, NJ: Prentice-Hall, 1952.

[3] J. L. Burbidge, "Production flow analysis," *Prod. Eng.*, vol. 42, no. 12, pp. 742–752, June 1963.

[4] B. Montreuil and P. Lefrancois, "Organizing factories as responsibility networks," in *Progress in Material Handling Research*, R. J. Graves *et al.*, Eds. Ann Arbor, MI: Material Handling Institute, Braun-Brumfield, 1996, pp. 375–411.

[5] U. Venkatadri, R. Rardin, and B. Montreuil, "A design methodology for fractal layout organization," *IIE Trans.*, vol. 29, pp. 911–924, 1997.

[6] D. B. Webster and M. B. Tyberghein, "Measuring flexibility of jobshop layouts," *Int. J. Prod. Res.*, vol. 18, no. 1, pp. 21–29, 1980.

[7] S. A. Irani, "A flow decomposition approach for integrated layout design of virtual group technology flowlines," Ph.D. dissertation, Dept. of Ind. and Manufac. Eng., The Pennsylvania State Univ., College Park, PA, 1990.

[8] J. Shukla, "Hybrid and progressive cellular layouts for jobshops," M.S. thesis, Dept. Mech. Eng., Univ. of Minnesota, Minneapolis, 1995.

[9] G. Harhalakis, T. Lu, I. Minis, and R. Nagi, "A practical method for design of hybrid-type production facilities," *Int. J. Prod. Res.*, vol. 34, no. 4, pp. 897–918, 1996.

[10] R. Tilsley, F. A. Lewis, and D. F. Galloway, "Flexible cell production systems: A realistic approach," *CIRP Ann.*, vol. 25, no. 1, pp. 269–271, 1977.

[11] D. Shunk and R. Reed, "An analytical approach to measure the effects of group technology," in *Proc. 3rd Int. Conf. on Production Systems*, 1975, pp. 627–634.

[12] R. G. Askin, N. H. Lundgren, and F. Ciarallo, "A material flow based evaluation of layout alternatives for agile manufacturing," in *Progress in Material Handling Research*, R. J. Graves *et al.*, Eds. Ann Arbor, MI: Material Handling Institute, Braun-Brumfield, 1996, pp. 71–90.

[13] A. J. Vakharia and U. Wemmerlov, "Designing a cellular manufacturing system: A materials flow approach based on operation sequences," *IIE Trans.*, vol. 22, no. 1, pp. 84–97, 1990.

[14] R. G. Askin and M. Zhou, "Formation of independent flowline cells based on operation requirements and machine capabilities," *IIE Trans.*, vol. 30, pp. 319–329, 1998.

[15] Y. C. Ho, C. C. Lee, and C. L. Moodie, "Two sequence-pattern, matching-based, flow analysis methods for multi-flowlines layout design," *Int. J. Prod. Res.*, vol. 31, no. 7, pp. 1557–1578, 1993.

[16] C. L. Moodie, J. Drolet, Y.-C. Ho, and G. M. H. Warren, "Cell design strategies for efficient material handling," in *Material Flow Systems in Manufacturing*, J. M. Tanchoco, Ed. New York: Chapman & Hall, 1994, ch. 3, pp. 76–101.

[17] S. A. Irani, "Production flow analysis in a semiconductor fab," in *Proc. Joint National Science Foundation and Semiconductor Research Corporation Workshop on Operational Methods in Semiconductor Manufacturing*, Feb. 19–20, 1997.

[18] K. Y. Tam, "An operation sequence based similarity coefficient for part families formations," *J. Manufac. Syst.*, vol. 9, no. 1, pp. 55–68, 1990.

[19] S. A. Irani and H. Huang, "Layout modules: A novel extension of hybrid cellular layouts," in *Proc. 1998 Int. Mechanical Engineering Congress & Exposition, Winter Annu. Meeting of the ASME*, Anaheim, CA, Nov. 15–20, 1998, Paper 98-WA/MH-1.

[20] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: Freeman, 1979.

[21] J. M. Mulvey and M. P. Beck, "Solving capacitated clustering problems," *Eur. J. Operational Res.*, vol. 18, pp. 339–348, 1984.

[22] D. Sankoff and J. Kruskal, *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison*. Reading, MA: Addison-Wesley, 1983.

[23] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Cybern. Contr. Theory*, vol. 10, no. 8, pp. 707–710, 1966.

[24] H. Huang and S. A. Irani, "Facility layout using operation sequences: A new similarity measure for operation sequences," Dept. of Ind., Welding and Syst. Eng., The Ohio State Univ., Columbus, OH, Working Paper, 1999.

[25] M. R. Anderberg, *Cluster Analysis for Applications*. New York: Academic, 1973.

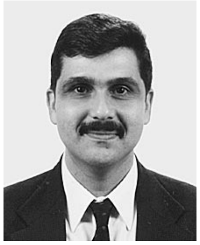
[26] S. T. S. Daita, S. A. Irani, and S. Kotamraju, "Algorithms for production flow analysis," *Int. J. Prod. Res.*, vol. 37, no. 11, pp. 2609–2638, 1999.

[27] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *Data Structures and Algorithms. Reading*. Reading, MA: Addison-Wesley, 1987.

[28] S. S. Skiena, *The Algorithm Design Manual*. New York: Spring-Verlag, 1998.

[29] B. Arvindh and S. A. Irani, "Cell formation: The need for integrated solution of the subproblems," *Int. J. Prod. Res.*, vol. 32, no. 5, pp. 1197–1218, 1994.

[30] L. R. Foulds and J. W. Giffin, "A graph theoretic heuristic for minimizing total transport cost in facilities layout," *Int. J. Prod. Res.*, vol. 23, no. 6, pp. 1247–1257, 1985.



Shahrukh A. Irani is currently an Assistant Professor in the Department of Industrial, Welding and Systems Engineering in the College of Engineering, Ohio State University, Columbus. Previously, from 1990 to 1996, he had been an Assistant Professor of Industrial Engineering in the Department of Mechanical Engineering at the University of Minnesota. His areas of research and teaching interest are cellular manufacturing, facility layout, group technology, design and analysis of manufacturing systems, lean manufacturing, and production control. He has been published in the *International Journal of Production Research*, *IIE Transactions*, *Journal of Manufacturing Systems*, and *ASME Journal of Engineering for Industry*. He is also Editor of the *Handbook of Cellular Manufacturing Systems* (New York: Wiley, 1999). He is actively involved in efforts to incorporate industry experience across the curriculum. In 1995–1996, he worked in industry on a faculty internship that was partially supported by the Grant Opportunities for Academic Liaison (GOALI) program of the National Science Foundation. His industry outreach activities include an industry workshop on cellular Manufacturing Systems which has been sponsored by the Higher Education Manufacturing Productivity Applications Consortium, Minnesota Consortium for Defense Conversion and the General Motors Technical Education Program. In 1996, Current research projects address the design of hybrid cellular layouts, software for production flow analysis, and the integration of pattern recognition algorithms with hypergraph theory for design of facility layouts.

Dr. Irani is a member of SME, IIE, and INFORMS and is a frequent Speaker or Session Chair at their conferences. He is presently serving as the Director of the Facilities Planning and Design (FAPAD) division of the Institute of Industrial Engineers for 1999–2001. He was voted Young Engineer of the Year by the Minnesota Federation of Engineering Societies and the Minneapolis Chapter of the Institute of Industrial Engineers.

Heng Huang received the B.S. degree in industrial instrumentation and automation and the M.S. degree in systems engineering from the Tianjin University, Tianjin, China, in 1994 and 1997, respectively. He is currently working toward the Ph.D. degree with the Department of Industrial, Welding and Systems Engineering at The Ohio State University, Columbus.

He is currently a Graduate Research Associate in the Department of Industrial, Welding and Systems Engineering at The Ohio State University. His research interests include production flow analysis, cellular manufacturing systems, pattern recognition, advanced methods for group technology, and next-generation facility layouts.