Dynamic Layout Algorithms : A State-of-the-art Survey

Jaydeep Balakrishnan
Faculty of Management
University of Calgary
Calgary, Alberta T2N1N4, CANADA

Chun Hung Cheng
Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong
Shatin HONG KONG

Published in *OMEGA*, 26, 4, 1998, 507-521

Address for Correspondence:

Jaydeep Balakrishnan
Faculty of Management
University of Calgary
Calgary Alberta T2N 1N4 CANADA
Ph: (403) 220 7844 Fax: (403) 282 0095
Email: balakris@ucalgary.ca

Dynamic Layout Algorithms : A State-of-the-art Survey

Abstract

It has been over a decade since Rosenblatt published his seminal paper on modelling the dynamic facility layout problem (DFLP). Since then, there have been improvements to Rosenblatt□s original dynamic programming model. Alternate solution methods have also been proposed. However, no comprehensive review of the research in the DFLP has been undertaken. In this paper we categorize the different works of research that have followed and discuss them. They include improved and more flexible solution methods, fathoming procedures, bound determinations and, method comparisons.

Keywords: dynamic facility layout, optimization, heuristics, modelling, survey

Dynamic Layout Algorithms : A State-of-the-art Survey

1. Introduction

This paper investigates the design of facility layouts based on multiperiod planning horizons. During this horizon, the material handling flows between the different departments in the layout may change. This necessitates a more sophisticated approach than the much researched static facility layout problem (SFLP) approach. The dynamic facility layout problem (DFLP) extends the SFLP by assuming that the material handling flows can change over time. This in turn might require layout rearrangement.

In the static problem, given a group of departments, material flow between each pair of departments, and the cost per unit of flow per unit distance, the departments have to be arranged into a layout such that the sum of the costs of flow between the departments in the layout is minimized. The material flows between pairs of departments or relative material flows are assumed to be constant over time.

The dynamic approach to layout corrects the above deficiency. In the dynamic approach, the layout plan is based on a multiperiod time horizon. During this time if the material flow changes warrant it, layout rearrangements may be planned in one or more periods. The analysis is based on the trade offs between the costs of excess material handling if a layout is not rearranged when required and the costs of such rearrangements. The cost of layout rearrangement would include out-of-pocket moving expenses as well as the cost of operational disruptions.

The next section explains the importance of the dynamic layout problem. Section 3 explains the dynamic problem through an example adopted from Rosenblatt [29]. Section 4 discusses the various solution approaches for the DFLP. A summary of these is given in Table 2. Finally, Section 5 concludes the paper.

2. Importance of Dynamic Layout

In an environment where material handling flows do not change over a long time, a static layout analysis would be sufficient. In today's market based and dynamic environment, such flows can change quickly. Page [26] reports that on average, 40% of a company□s sales come from new products, i.e., products that have only recently been introduced. Any change in product mix can result in changes in flow and thus affect layouts.

In a dynamic environment, the static approach may still be used, though there are disadvantages. One method is to use a short planning horizon so that during this horizon the flows are fairly constant. The disadvantage is that after the short horizon, if the relative flows change, the layout may have to be rearranged to maintain the effectiveness of the layout. However, rearranging layouts frequently without prior planning can result in operational disruptions and excess rearrangement costs. Another approach is to use a long planning horizon and disregard the changes in flow. The total flow over the planning horizon can be determined by adding the material flows in each period to get the total flow. There will be no rearrangement costs, but this may result in the layout being inefficient throughout the horizon. The dynamic approach to layout corrects these deficiencies by striking a balance between the material handling and layout rearrangement costs and planning future layout rearrangements if necessary.

However, dynamic layout analysis may not be justified in every situation. When the cost of layout rearrangement is negligible, dynamic layout analysis is not necessary. The layout can be redesigned as necessary when demand changes require it, without any prior planning. Examples of such layouts can be found in Hirano [13] . At the other extreme, if the rearrangement costs are prohibitive, such as in the case of very heavy machinery, we may use the same layout for the entire planning horizon. In this situation also a dynamic analysis is not necessary. This paper concentrates on the intermediate settings where the costs of layout rearrangement are neither negligible nor prohibitive. Examples of such layouts can be found in Kouvelis et al. [16]. They describe examples in flexible manufacturing systems (FMS) and semiconductor manufacturing.

3. The Dynamic Facility Layout Problem (DFLP)

A rectangular layout with six locations is shown in Figure 1a. The locations are all equal in size. There are six departments that have to be assigned to the six locations in each of the five periods in the planning horizon. The assignment of departments to locations for one period shown in Figure 1a is designated by the sequence 1-2-3-4-5-6. Figure 1b shows another possible assignment and is designated by the sequence 2-6-3-4-1-5

This is only one out of the 6! or 720 combinations or sequences that exist for this layout. Each combination represents a different static layout and is a candidate layout in each period of the five period dynamic problem. Each layout is also denoted by a sequence indicated in Figures 1a and 1b. We are interested in determining the assignment of departments in this layout over five periods. A period could represent any length of time such as a year, a quarter, or a month. The varying relative flows between pairs of departments along with the shifting cost for each department are shown in Table 1

In this example, the shifting cost depends only on the department shifted and not on the distance of the move. This assumption is valid where the fixed costs of the move (such as removing the machine, reinstalling it, and the cost of lost production) dominate the variable costs such as the cost to move the machine unit distances.

1	**2**	**3**
4	**5**	**6**

Figure 1a: An example layout

2	**6**	**3**
4	**1**	**5**

Figure 1b: An example layout

Period 1

To -- From	**1**	**2**	**3**	**4**	**5**	**6**
1	0	63	605	551	116	136
2	63	0	635	941	50	191
3	104	71	0	569	136	55
4	65	193	622	0	77	90
5	162	174	607	591	0	179
6	156	13	667	611	175	0
Period 2						
1	0	175	804	904	56	176
2	63	0	743	936	45	177
3	168	85	0	918	138	134
4	51	94	962	0	173	39
5	97	104	730	634	0	144
6	95	115	983	597	24	0
Period 3						
1	0	90	77	553	769	139
2	168	0	114	653	525	185
3	32	35	0	664	898	87
4	27	166	42	0	960	179
5	185	56	44	926	0	104
6	72	128	173	634	687	0
Period 4						
1	0	112	15	199	665	649
2	153	0	116	173	912	671
3	10	28	0	182	855	542
4	29	69	15	0	552	751
5	198	71	42	24	0	758
6	62	109	170	90	973	0
Period 5						
1	0	663	23	128	119	50
2	820	0	5	98	141	66
3	822	650	0	137	78	91
4	826	570	149	0	93	151
5	915	515	53	35	0	177
6	614	729	178	10	99	0

Shifting cost for departments

887	964	213	367	289	477

Table 1: Material Flow and Shifting Costs

The layout also shows flow dominance. Some departments have higher material handling inflows than the others. These flow dominant departments change during the planning horizon. For example, in Table 1, departments 3 and 4 have high material inflows during the first period whereas in the fifth period, it is departments 1 and 2 that have the high material inflows. This results in the dynamic nature of the problem and implies that the optimal static layout may also change during the planning horizon.

The optimal static layouts, based on the notation in Figures 1a and 1b, for the problem in Table 1 are 135642, 142536, 153246, 164253 and 326415 in periods 1 through 5 respectively. If the layout rearrangement cost was negligible, the optimal solution would have been to rearrange the layout to the optimal static layout in each period. However, layout rearrangement incurs costs. Based on the entire planning horizon, if the savings in the cost of material handling due to rearrangements is greater than the cost of shifting the departments, we might rearrange the layout.

So the dynamic problem involves selecting a static layout for each period and then deciding whether to change to a different layout in the next period. If the shifting costs are relatively low, the layout configuration would tend to change more often to retain material handling efficiency. The reverse is true for high shifting costs.

4. Discussion of the Dynamic Layout Literature

4.1 Equal Sized Departments

4.1.1 Deterministic Material Flow

A majority of the research that has been done in the DFLP assumes equal department sizes and deterministic material flow . Often, solving the DFLP includes solving the SFLP. For most realistic SFLPs, where a large number of departments are present, obtaining the optimal solution will not be possible. So various heuristic algorithms have been proposed which can solve fairly large static layout problems in reasonable amounts of time. Heuristic algorithms can be classified into two major types - construction type algorithms where a solution is constructed from scratch and, improvement type algorithms, where an initial layout is improved. CRAFT [2], is a popular improvement algorithm that uses pair-wise interchange (exchange). For a comprehensive review of the static layout literature, see Kusiak and Heragu [18]. More recent work can be found in Meller and Gau [21].

Less literature is available in the dynamic facility layout area. The following formulation of the DFLP is adopted from Balakrishnan et al. [4]:

$$\text{Min} \sum_{t=2}^{P}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{m=1}^{N} A_{tijm} Y_{tijm} + \sum_{t=1}^{P}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{m=1}^{N} F_{tijkm} X_{tij} X_{tkm}$$

$$subject\,to$$

$$\sum_{i=1}^{N} X_{tij} = 1 \qquad j = 1,..N \quad t = 1,...P$$

$$\sum_{j=1}^{N} X_{tij} = 1 \qquad i = 1,..N \quad t = 1,...P$$

$$Y_{tijm} = X_{(t1)ij} \, x \, X_{tim} \qquad i, j, m = 1,..N \quad t = 2,..P$$

i,k : *Departments in the layout*

j,m : *Locations in the layout*

Y_{tijm} : *0,1 variable for shifting* i *from* j *to* m *in period* t

A_{tijm} : *Fixed cost of shifting* i *from* j *to* m *in period* t
(where $A_{tijj} = 0$*)*

X_{tij} : *0,1 variable for locating* i *at* j *in period* t

F_{tijkm} : *Cost of material flow between* i *located at* j
and k *located at* m *in period* t

P : *Number of periods in the planning horizon*

N : *Number of departments in the layout*

The objective is to minimize the sum of the layout rearrangement costs (first term) and the material handling costs (second term) over the planning horizon. (1) requires every department to be assigned. (2) requires every location to have a department assigned to it. (3) assigns Y_{tijm} a value of 1 only if a department has been shifted in the period. This formulation is nonlinear and can be solved optimally only for small problems.

This formulation is an extension of the well-known quadratic assignment problem (QAP) [15] formulation for the SFLP. Although, many solution techniques for the DFLP involve solving embedded SFLPs, the optimum solution to the dynamic problem might not involve any of the static optimum solutions.

Sahni & Gonzalez [31] show that the QAP formulation for the SFLP is a NP-complete problem.

Thus the DFLP is also a very difficult problem to solve optimally. We would have to explicitly or implicitly evaluate $(n!)^t$ combinations where t is the number of periods, which even for a six department, five period problem is very large (1.93×10^{14}). Thus most realistic problems will have to be solved suboptimally, i.e., not all the possible layouts in a period are explicitly or implicitly evaluated. In this case, there is no guarantee of optimality for the dynamic problem.

The optimum solution for the problem shown in Table 1, based on the notation in Figures 1a and 1b, is to employ layout 246135 in the first two periods, then shift to layout 246153 for period 3. In period four, layout 264153 will be used and finally in period 5, layout 214653 will be employed. Thus, the layout is changed three times during the five periods. None of these layouts are statically optimal in their respective periods. The total cost of this plan is $71187. Had the static optimal solutions been used, the savings in material handling cost would have been more than offset by the increased layout rearrangement costs, and the total five period cost would have increased to $75384.

4.1.1.1 Dynamic Programming Approaches

Rosenblatt□s model of the DFLP is adopted from Ballou [5] who considers the dynamic nature of demand in locating warehouses. Location decisions are generally made based on horizons of twenty years or more. Thus, the changes in demand during that time (dynamic nature) can be important. In the dynamic plan, the warehouse location for each period is different from the location that would have been used if that period's demand was considered in isolation (static location). The objective here is to trade off the costs of warehouse relocation against the distribution costs saved.

Sweeney and Tatham [32] provide a fathoming procedure for the dynamic facility location problem that allows us to eliminate some of the static locations or states that would be required in each period. The rule is applicable to dynamic layout also. One disadvantage of this method is that a good feasible solution is required before the rule can be applied effectively.

The DFLP is similar to dynamic warehouse location. The trade off here is between the material handling flow cost within the facility and the shifting costs for the departments that may need to be relocated within the facility. Rosenblatt uses DP to solve a six department problem optimally.

Let:

L_i *Layout i*

A_{ij} *Rearrangement (sum of department shifting costs) cost when changing from layout L_i to layout $L_{j,}$. This cost is independent of the period in which it occurs.*

F_{it} *Material handling cost for layout L_i in period t. This is obtained from the SFLP solution.*

C^*_{it} *Minimum total costs (material handling and shifting) for all periods up to t where L_i is used in period t.*

The combination of layouts with the minimum total cost is chosen based on the following recursive relationship:

$$\begin{array}{l}\text{Minimum cost}\\ \text{up to layout } j\\ \text{in present}\\ \text{period } t\end{array} = \begin{array}{l}\text{Minimum}\\ \text{of (for}\\ \text{every}\\ \text{static}\\ \text{layout } i)\end{array} \left[\begin{array}{l}\text{Best solution for } i\\ \text{up to period}\\ (t\text{-}1) + \text{rearrange-}\\ \text{ment cost to}\\ \text{layout } j\end{array}\right] + \begin{array}{l}\text{Material handling}\\ \text{cost for layout } j \text{ in}\\ \text{present period } t\end{array}$$

or

$$C^{*}_{jt} = Min_{i}\,\{C^{*i}_{i(t-1)} + A_{ij}\} + F_{jt}$$

The DP is solved using backward recursion. Each period in the planning horizon forms a stage and each static layout forms a state. He reports that in his experiments, the fathoming procedure of Sweeney and Tatham did not help in eliminating any state.

For small problems, $N = n!$, where $n!$ is the number of possible static layouts (given n departments) in each period and N is the number of static layouts included in the DP. The solution will be optimal in this case. For larger problems $N < n!$, since using all $n!$ static layouts will result in an intractable problem. The DP procedure still gives the optimal solution for the layouts included. However, as not all the possible static layouts are included, i.e. ($N < n!$), the resultant procedure (with static and dynamic stages) cannot guarantee the optimal solution. N in each period depends on the capability of the software and hardware used to solve the problem. The more powerful these are, the larger N can be. Logically, larger N should lead to better solutions. In most practical sized problems $N << n!$. Rosenblatt discusses different methods of selecting the N layouts when $N < n!$. One method is to choose them randomly. This is computational efficient but the resulting solution quality is usually poor.

Another method suggested by Sweeney and Tatham is to use the N best static layouts from each period. The assumption is that since we are including the best layouts from each period, the overall dynamic solution should be better than if we chose the static layouts randomly. This was borne out by Rosenblatt□s tests, where using the best layouts resulted in better solutions than when using random layouts. But generating the best layouts involves solving N QAPs in each period, each with more constraints than the previous QAP (to prevent the recurrence of the previous static optimal solution). This can be very time consuming and will not be practical for larger problems.

It is also important to have any static layout included in one period to be duplicated in every other period. Otherwise, we may eliminate the possibility of using the same layout in more than one period, which may be preferable when layout rearrangement costs are high.

Though Rosenblatt did not conduct any experiments using large problems, this paper is frequently cited in dynamic layout as it defines the problem clearly and discusses directions for solving practical sized problems in the field.

Balakrishnan et al. extend the dynamic layout problem of Rosenblatt. They consider the existence of budget constraints and conduct a detailed experiment to investigate the problem. Their formulation is called the constrained dynamic plant layout problem (CDPLP). They point out that layout rearrangement requires funds and these funds may be limited. So the dynamic layout problem is solved under this constraint. The simplex based constrained shortest path (CSP) algorithm of Mote et al. [25] is used to solve the problem.

In addition, an experiment involving the number of static layouts, the method of static layout selection, and constraint tightness was undertaken. Problems with fifteen and twenty departments were also solved. Thus, this paper illustrates two important aspects of the dynamic layout problem: 1) When constraints are added to the dynamic layout problem, shortest path algorithms are much faster than dynamic programming in realistic situations; 2) A heuristic approach to the dynamic layout problem can be effective. CRAFT was used in the experiments to generate static solutions. For realistic problems, as it will not be possible to find the *N* best static solutions by solving the QAP, the ability of CRAFT to provide *N* good solutions allows large size DFLPs to be solved effectively.

Another implication of this research is that instead of CRAFT, we could use other algorithms such as SPIRAL [11] which is useful for unequal department sizes. This means that although the Rosenblatt model assumes equal size departments, this drawback can be circumvented by using a procedure such as SPIRAL in the static stage. The dynamic programming stage is of course independent of the shapes or sizes of the departments as it deals with only the costs of material handling and rearrangement.

The use of incomplete dynamic programming [36] for the DFLP, where the rearrangement costs are assumed to be fixed regardless of the departments rearranged, is discussed by Urban [35]. For example, this may occur when the whole facility has to be shut down for any rearrangement and this cost dominates the cost to shift individual departments. Thus, the decision in each period can be reduced to (0,1) depending on whether rearrangement occurs. This then leads to a rearrangement vector for a multiperiod problem. The solution is found in two phases. In the first phase, a series of QAPs have to be solved to find the solution to the static sub-problems. In the second phase, the solutions from the first phase form the arcs on a shortest path formulation with each period represented by a node. For larger problems two heuristics are proposed. The first one, GRASP, uses a construction algorithm with neighbourhood search to come up with good solutions for the QAP. The second algorithm, the initialized multigreedy algorithm, is very similar to GRASP, except that it shares information between the static subproblems. In tests both these heuristics performed well, generating solutions within 1% of optimal.

4.1.1.2 Pair-wise Interchange Heuristics

The pair-wise interchange procedure proposed by Urban [34] is the multiperiod equivalent of CRAFT and incorporates layout rearrangement costs. This heuristic makes use of ▯forecast windows▯, *m*, to find different sets of good layout plans for the planning horizon. The *m* ranges from 1 to the number of periods in the planning horizon. The forecast window is the number of periods being considered when the pair-wise exchange is performed. Using an initial layout and pair-wise exchanges, one set of layouts is obtained for the given planning horizon for each forecast window.

For example, when the forecast window is 1, *i.e., m*=1, in each run of the pair-wise interchange procedure, only material flows from one period are considered. An assumed or existing initial layout is used to find the most appropriate layout for period 1 by considering the material flows for period 1 only. Then this newly generated layout for period 1 is used as the initial layout for period 2. Pair-wise interchange is now used to determine a good layout for period 2 by considering the material flow of period 2 only. This process is repeated for each period in the planning horizon using the newly generated layout for the previous period as the initial layout for pair-wise interchange. Thus, a layout plan for the entire planning horizon is obtained. The total cost of the plan is the sum of the material handling flows for every period in the planning horizon and the costs of rearranging the layout at the end of each period if necessary.

The next stage of the procedure involves using a forecast window of 2. This is shown in Figure 2 for a five period problem. The material flows of periods 1 and period 2 are combined to determine the layout for period 1 from an initial layout using pair-wise interchange. This final layout for period 1 serves as an initial layout for the next period□s analysis. Similarly, the flow costs for periods 2 and 3 are combined and used to determine the layout for period 2, and so on. Thus, a look ahead principle is used. Note that in period 5, only the material flow for period 5 is used as period 6 does not exist. As was the case with *m*=1, the total cost of the plan is the sum of material handling flows for every period in the planning horizon and the costs of rearranging the layout at the end of each period if necessary. In a five period problem, this process is repeated for *m = 1,2,3,4,5*. For each *m*, a layout plan and an associated cost is obtained. The plan with the lowest cost is selected as the solution.

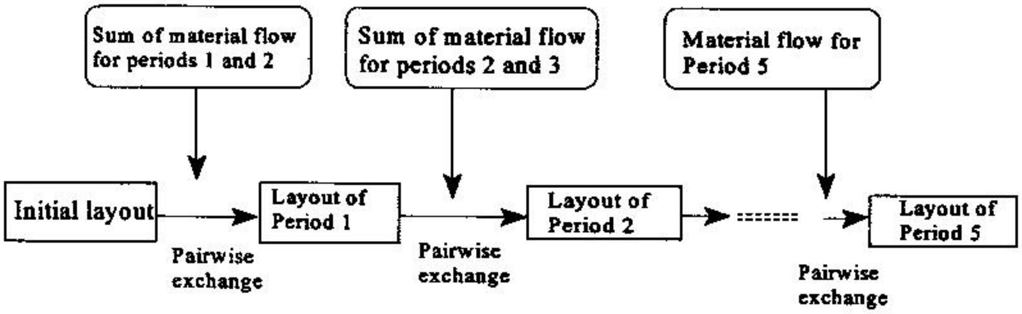

Figure 2

Pair-wise exchange when forecast window *m* = 2

Test problems ranging in size from six departments and four periods to thirty departments and twenty periods were solved. The performance of the heuristic (using two initial layouts) was compared to dynamic programming using; 1) one hundred random static layouts in each period and; 2) four best layouts from each period (Ballou method). The results indicate that the proposed heuristic performs better than DP (using random static layouts) especially for larger problems. In these problems DP is solved as a heuristic since $N << n!$. The second best method was the Ballou method. However, it could not provide solutions for problems with more than twelve departments. On the other hand the Urban heuristic was able to solve a thirty department, twenty period problem in a little more than 1000 CPU seconds on 10-MHz 386 computer. On problems where an optimal solution was available, the exchange heuristic performed only slightly worse than optimal. In addition, since different forecast windows are used, this method might be more suitable than DP under rolling planning horizons. Thus, Urban□s heuristic is an extremely practical approach. The only disadvantage is that it can handle only equal sized departments. However, the concept of the forecast window can be applied to unequal sized department heuristics such as SPIRAL, which would result in an Urban-like procedure incorporating unequal sized departments.

4.1.1.3 Genetic Algorithms

Conway and Venkataramanan [8] examine the suitability of genetic algorithms (GA) for the CDPLP. A genetic search uses the mechanics of natural selection and natural genetics to evolve a population of initial solutions into a near-optimal solution. This approach is suited to handle multiple and nonlinear objective functions as well as side constraints.

In their procedure, for a six department, five period problem, a string in the population would consist of $n \times t$, or 30 digits, representing every department in each period. There would also be *n!* or 720 such strings in the population. To crossbreed, two strong strings (based on fitness function) from the population are selected. Then a random splicing position is generated and the strings are split. The substrings are then swapped. Since these swaps may create infeasible solutions (eg. two occurrences of department 5 in one period), additional digit swaps are done to ensure feasibility. The string with lower cost is allowed to survive to the next generation. In tests, where cross-breeding was done for up to 100 generations using population sizes of up to 800, the algorithm performed well compared to dynamic programming for six and nine department problems. No computation times were given.

4.1.1.4 Tabu-Search

The DFLP lends itself well to tabu-search. In the method proposed by Kaku and Mazzola [14], pairwise interchange is used to evaluate candidate moves in a local neighbourhood search. A tabu-list is maintained to prevent cycling and in each iteration the “best” non-tabu move is implemented if it results in a better solution. The procedure stops when the incumbent solution does not improve after a number of moves or if an iteration limit is reached. Diversification strategies are used to ensure that different regions of the search space are explored for better solutions. Intensification strategies which allow the procedure to do more searches in a neighbourhood, by actions such as reducing the tabu list length, also proved helpful. The algorithm has two stages. Diversification allows for a number of starting solutions. The tabu-search then generates good solutions which are fed into the second stage for a more intensive search. The best solution found in the second stage is the final solution. Computational experiments showed that the tabu-search procedure provided solutions that were as good as or slightly better than the best solutions for different algorithms in Lacksonen and Enscore [20]. Problems with 30 departments and 5 periods took about 3 hours on an average to solve on a Pentium 200 Mhz PC.

4.1.1.5 Comparison of Algorithms

Lacksonen and Enscore conduct tests on various methods of solving the quadratic assignment formulation of the dynamic layout problem. The authors modified these methods for use in dynamic layout. The procedures were:

1. Exchange algorithm

Each run starts with eight random layouts. Then pairs of departments are analysed for exchanges over all consecutive blocks of time in a CRAFT based method. This converts the static pair-wise exchange into a dynamic one.

2. Cutting planes

Based on Bukard and Bonniger [7], this method combines cutting planes with an exchange routine. At each iteration, an assignment and exchange routine are implemented. The process is repeated for different starting solutions. For simplicity, the assignment routine does not consider department changes between time periods. This is done by the exchange routine.

3. Branch and bound

Based on the algorithm of Pardolos and Crouse [27], this method employs the cutting plane results as an upper bound. The procedure is run as a heuristic by storing on the best nodes and by terminating after a certain number of nodes are analysed.

4. Dynamic Programming

This is the DP method of Rosenblatt . *N* in the DP ranged from 30 to 700. It was larger for the bigger problems. The exchange algorithm is used to find the best static layouts. Hybrids between the best layouts of consecutive time periods are also used as static layouts.

5. Cut trees

This is based on the research in [22]. Each department in each time period is represented by a node. Arcs represent material flow and layout rearrangement. Cut tree arcs are then manually converted into layouts. This manual layout is then fed into the exchange algorithm to obtain better results.

Problems ranging in size from six departments and three periods to thirty departments and five periods were tested. In the larger problems optimality could not be proved. They found that the cutting plane and branch and bound algorithms performed better than the exchange algorithm.

DP and cut tree performed poorly. In addition the cut trees were not applied to the thirty department problems as its implementation became difficult. The branch and bound was not able to solve the thirty department problems within the allotted CPU time. On the thirty department problems the exchange algorithm and DP provided solutions that were respectively, 1.3% and 3.7% more costly on average than the cutting plane solution. However the cutting plane method required 27 times and 70 times respectively as much computation time on average than the exchange algorithm and DP (on an IBM 3070, the cutting plane took between 172.47 and 235.97 CPU seconds to solve a thirty department, five period problem). Further, the cutting plane is limited to equal sized layouts.

4.1.1.6 Bounds and Fathoming

Balakrishnan [3] proposes a fathoming procedure to reduce the number of possible static layouts in the DFLP. One advantage of this method is that this can be applied before any feasible solution is available. The disadvantage is that effectiveness of this fathoming method depends on the magnitude of the rearrangement costs. For high rearrangement costs it is not likely to be effective. Batta [6] shows that if the same layout is used in every period, then the DFLP reduces to an SFLP in which the interdepartmental flow can be determined by adding the flow data for all the periods.

Urban [33] examines the application of lower bounds to the DFLP. This is useful in eliminating solutions when using procedures such as branch and bound, and also in testing the effectiveness of heuristics when optimal solutions are not available. Rosenblatt suggests using the sum of the static optimal solutions as a bound. This ignores the rearrangement costs and so forms a lower bound. However, this bound requires the static optimal solutions, which may be difficult to obtain for large problems.

Urban therefore examines other lower bounds such as Gilmore type bounds [10] and probabilistic bounds [12]. The computations required for the Gilmore type are less than that for Rosenblatt□s bounds. A probabilistic bound is of the type μ - $k\sigma$, where μ and σ are the mean and standard deviation of the material handling flow distribution. The computational performances of these different lower bounds were compared in an experiment. In general, these bounds are more effective when the variations in work flow between departments and variations in interdepartmental distances are low.

Rosenblatt□s bound is usually the most effective but works only for small problems due to excessive computation time. The probabilistic method is computational the most efficient and can perform better than Rosenblatt□s under small k and low flow variability. Gilmore type bounds are dominated by Rosenblatt bounds but are close in some cases.

In a more recent paper, Urban [35] proposes another set of lower and upper bounds. The bounds would involve solving more QAPs than the bounds in Rosenblatt, Batta, and Balakrishnan. However, the extra effort results in more states being fathomed on some problems. In tests on problems with up to 15 departments and 8 periods, the proposed bounds dominate the Rosenblatt, and Batta bounds. It eliminates more static solutions than the Balakrishnan□s fathoming procedure for shorter planning horizons. Under longer horizons Balakrishnan□s fathoming procedure performs better.

4.1.2 Stochastic Material Flow

4.1.2.1 Robustness

All the procedures discussed so far assume a deterministic scenario as in Table 1. Clearly, this may not be true in many situations. Kouvelis et al. [16] address this problem by using the concept of □robust layouts□ [30]. The robustness of a layout is an indicator of its flexibility in handling demand changes. This can be measured by whether a designed layout is within a Δ% of the optimal solution for every possible demand scenario. Under uncertainty, it may be better to choose a layout which performs well under all possible situations rather than one that is optimal for one possible scenario (which may not occur) but does poorly for the scenario that actually occurs. They also introduce the concept of □monuments□; those departments that are prohibitively expensive to shift once they are located.

Their method involves using a branch and bound (B&B) procedure for finding a list of solutions for the static problem in each period that is within Δ% of optimality. Then families from these solutions are identified. A family is a set of multiperiod layouts in which the monuments are not shifted during the planning horizon. Then solutions from families that are common across all scenarios are identified. These form the candidate list of solutions which can then be evaluated for deviation from optimality for each scenario, given the $\Delta\%$.

Experiments indicate that for more than 15 departments, the B&B procedure would have to be terminated before optimality confirmation. The authors do solve thirty department problems, but only for the static case. The B&B procedure is limited to equal sized departments. However, in the dynamic stage, the procedure is independent of the method used to generate the $\Delta\%$ solutions for the static problems. So, conceptually, heuristics such as CRAFT or SPIRAL could be used in the static stage.

4.1.2.2 Markov Processes

Another paper that deals with uncertainty in dynamic layout is one by Kouvelis and Kiran [17]. However, the methodology described in that paper is specific to automated manufacturing systems. The basic model is a modification of the QAP which incorporates a close queueing network. This network allows the authors to model stochastic factors such as setup and material transportation times, alternate process plans, and product mixes. For the dynamic model, a transitional matrix is used to update the probabilities of the product mixes in each time period. Computational however, this leads to a Markov process resulting in large problems becoming intractable. The authors are able to solve only a special case of the problem using dynamic programming.

4.1.2.3 Simulation

In one of the few discrete event simulation approaches to dynamic layout, Afentakis et al. [1] model a flexible manufacturing system. The main objective of the research is to compare a strategy of rearranging a layout every *n* periods (periodic policies) with a strategy of rearranging it when the product volume, product mix or product routing changes by a threshold percentage value (threshold policies). Factors incorporated include the number and routing of parts and the number of machines. The volume of parts and stability of this volume along with the frequency of change in the part mix were also considered. Each factor had two levels. In this research, the authors do not use the sum of material handling and rearrangement costs as the measure of performance. Each is considered a separate measure of performance. In analysing the results, they identify some dominant situations when considering both measures. They also focus on the tradeoff between the two. The optimal static layouts were determined by complete enumeration. The results showed that given a dynamic situation, the material handling performance deteriorated as *n* increased in the periodic policy and as the threshold percentage increased, while the number of rearrangements decreased. Also when the part mix changes, the threshold policies work better than the periodic policies. Overall, the results show that a poor layout can add as much as 36% to the material handling requirements. Therefore, given that FMSs are installed to perform better than job shops or dedicated lines under conditions of uncertainty, it is important to monitor them continuously on a layout basis to ensure its efficiency.

4.2 Unequal Size Departments

Montreuil and Venkatadri [23] deal with a situation where the departments in a facility show growth or decline over time. This growth or decline is divided into phases or periods. They employ strategic interpolative design [9] in which the designer develops a goal layout for the facility at the end of the growth stage. This layout is an input to the model and the intermediate layouts are generated from this layout. It is also assumed that once a department is located, it is never moved. The size of each department may grow at different rates but their relative positions do not change over time. They model this dynamic situation using linear programming.

The variables are coordinates along the X and Y axes and are bounded by the dimensions of the building also along the X and Y axes. The objective function minimizes a weighted average of the flow costs where the weight assigned to a phase depends on the duration of the phase and the time value of money. Though the initial formulation considers only rigid facilities, they also discuss modifications required to incorporate expansion and phasing out of the building and, phased construction. Since the formulation is linear, the model can solve problems of realistic size. A twelve department problem required 504 variables and 648 constraints and was solved within 10 minutes on a IBM AT computer. This research addresses the inability of the QAP based models to address the case where department sizes may be unequal. But it does so only by restricting the dynamic layout plan to a predefined skeleton. This formulation handles unequal department sizes. Thus, it addresses the deficiency in the formulation of Balakrishnan et al.

However, this formulation has a disadvantage of its own. To maintain the tractability of the formulation, an initial design skeleton that fixes the relative positions of the departments has to be specified. In later periods, only the sizes and shapes of the departments can be changed. The relative positions remain the same. This can be seen in the formulation where there is no term for the layout rearrangement cost as departments cannot be shifted relative to each other.

Montreuil and Laforge [24] extend the work by Montreuil and Venkatadri by presenting a more general procedure to analyse the design of layouts under conditions of uncertainty. The authors consider a set of probabilistic future scenarios to design the layout. The formulation is shown below. Let:

Indices:

c,i, j: A cell.

e: A facility, referring to a cell or the building

f: A future state in the scenario tree.

m,n,s: An Input/Output (I/O) station of a cell.

p: A pair of I/O stations, referring to station *m* of cell *i* and station *n* of cell *j*.

p(f): Future preceding future *f* in the scenario tree.

Parameters:

I_{pf}: Positive interaction between the I/O stations of pair *p* during future *f*.

$L\underline{U}_{ef}, L\bar{U}_{ef}$: Minimum and maximum allowed length for the *X*-axes of facility *e* in future *f*.

$L\underline{S}_{ef}, L\bar{S}_{ef}$: Minimum and maximum allowed length for the *Y*-axes of facility *e* in future *f*.

P_f: Probability of occurrence of future f according to the scenario tree.

$\underline{O}_{ef}, \bar{O}_{ef}$: Minimum and maximum allowed perimeter for facility *e* in future *f*.

$WDU_{ef}, WD\underline{S_{ef}}$: Marginal cost associated to a unit-distance displacement of the centroid of facility *e* along the *X*- and *Y*-axes, from its location in future *p(f)* to its location in future *f*.

$WD\underline{U}_{ef}, WD\bar{U}_{\underline{ef}}$,
$WD\underline{S}_{ef}, WD\bar{S}_{ef}$: Marginal cost associated to a unit-distance displacement of the west, east,

south and north sides of facility *e* from their location in future *p(f)* to their location in future *f*.

W_f: Weight associated to future *f* considering its duration, start time and associated discounted value of money, etc.

$(Z\bar{U}_{cf}, Z\bar{U}_{cf})$,
$(Z\bar{S}_{cf}, Z\bar{S}_{cf})$ Extreme lower and upper coordinates of a rectangular zone within which cell *c* is imposed to be laid out in future *f*.

Sets:

B: The building.

C: Set of cells *c*.

I_f: Set of pairs *p* of I/O stations such that there is a positive flow between the two I/O stations during future *f*.

$N_{i\backslash if}$: Set of cells *i* and *j* such that the design skeleton for future *f* states that, in future *f*, cells *i* and *j* are to be neighbours, with cell *i* west of cell *j*.

$N_{i/if}$: Set of cells *i* and *j* such that the design skeleton for future *f* states that, in future *f*, cells *i* and *j* are to be neighbors, with cell *i* north of cell *j*.

S_c: Set of I/O stations *s* of cell *c*.

Variables:

$D\underline{U}^{+}_{ef}, D\underline{U}^{-}_{ef}, D\bar{U}^{+}_{ef}, D\bar{U}^{-}_{cf}$: Positive and negative components of the displacement of the west and east sides of facility *e*, from their location in future *p(f)* to their location in future *f*.

$DU^{+}_{ef}, DU^{-}_{ef}, DS^{+}_{ef}, DS^{-}_{ef}$: Positive and negative components of the displacement of the centroid of facility *e* from its location in future *f*, along the *X*-axis and the *Y*-axis.

$D\underline{S}^{+}_{ef}$, $D\underline{S}^{-}_{ef}, D\bar{S}^{+}_{ef}, D\bar{S}_{ef}$: Positive and negative components of the displacement of the south and north sides of facility *e*, from their location in future *p(f)* to their location in future *f*.

(x_{csf}, y_{csf}): Coordinates of I/O station *s* of cell *c* in future *f*.

$(\underline{U}_{\underline{e}f}, \underline{S}_{ef})$, $(\bar{U}_{ef}, \bar{S}_{ef})$: Extreme lower and upper coordinates of rectangular facility *e* in future *f*, along the *X*-axis and the *Y*-axis.

$x^{+}_{pf}, x^{-}_{pf}, y^{+}_{pf,} y^{-}_{pf}$: Positive and negative components of distance between the Input/Output stations *im* and *jn* of pair *p*, along the *X*-axis and the *Y*-axis, as the stations are location in future *f*.

Formulation:

$$\text{Minimize} \quad \sum_{\forall f} P_f W_f \quad \Big[\sum_{\forall_p \varepsilon I_f} I_{pf} \, (x^+_{pf} + x^-_{pf} + y^+_{pf} + y^-_{pf})$$

$$+ \sum_{\forall e \varepsilon CUB} (WD\underline{U}_{ef} \, (D\underline{U}^+_{ef} + D\underline{U}^-_{ef}) + WD\bar{U}_{ef} \, (D\bar{U}^+_{ef} + D\bar{U}^-_{ef})$$

$$+ \ (WD\underline{S}_{ef} \, (D\underline{S}^+_{ef} + D\underline{S}^-_{ef}) + WD\bar{S}_{ef} \, (D\bar{S}^+_{ef} + D\bar{S}^-_{ef}) \qquad (1)$$

$$+ \ WDS_{ef} \, (DS^+_{ef} + DS^-_{ef}) + WDS_{ef} \, (DS^+_{ef} + DS^-_{ef}) \Big]$$

Subject to

$$L\underline{U}_{ef} \leq \bar{U}_{ef} - \underline{U}_{ef} \leq L\bar{U}_{ef} \qquad \forall \, e, f \qquad (2)$$

$$L\underline{S}_{ef} \leq \bar{S}_{ef} - \underline{S}_{ef} \leq L\bar{S}_{ef} \qquad \forall \, e, f \qquad (3)$$

$$\underline{O}_{ef} \leq 2 \, (\bar{U}_{ef} - \underline{U}_{ef} + \bar{S}_{ef} - \underline{S}_{ef}) \leq \bar{O}_{ef}) \qquad \forall \, e, f \qquad (4)$$

$$\underline{U}_{ef} \leq x_{csf} \leq \bar{U}_{cf} \qquad \forall \, c, f \qquad \forall s \varepsilon S_c \qquad (5)$$

$$\underline{S}_{ef} \leq y_{csf} \leq \bar{S}_{cf} \qquad \forall \, c, f \qquad \forall s \varepsilon S_c \qquad (6)$$

$$\underline{U}_{Bf} \leq Z\underline{U}_{cf} \leq \bar{U}_{cf} \leq Z\bar{U}_{cf} \leq \bar{U}_{Bf} \qquad \forall \, c, f \qquad (7)$$

$$\underline{S}_{Bf} \leq Z\underline{S}_{cf} \leq \bar{S}_{cf} \leq Z\bar{S}_{cf} \leq \bar{S}_{Bf} \qquad \forall \, c, f \qquad (8)$$

$$x_{imf} - x_{jnf} = x^+_{pf} - x^-_{pf} \qquad \forall f; \ \forall p = (im, jn) \, \varepsilon I_f \qquad (9)$$

$$y_{imf} - y_{jnf} = y^+_{p\,f} - y^-_{pf} \qquad \forall f; \ \forall p = (im, jn) \, \varepsilon I_f \qquad (10)$$

$$1/2 \, ((\underline{U}_{ep(f)} + \bar{U}_{ep(f)}) - (\underline{U}_{ef} + \bar{U}_{ef})) = DU^+_{ef} - DU^-_{ef} \qquad \forall \, e, f \qquad (11)$$

$$1/2 \, ((\underline{S}_{ep(f)} + \bar{S}_{ep(f)}) - (\underline{S}_{ef} + \bar{S}_{ef})) = DS^+_{ef} - DS^-_{ef} \qquad \forall \, e, f \qquad (12)$$

$$\underline{U}_{ep(f)} - \underline{U}_{ef} = D\underline{U}^+_{ef} - D\underline{U}^-_{ef} \qquad \forall \, e, f \qquad (13)$$

$$\bar{U}_{ep(f)} - \bar{U}_{ef} = D\bar{U}_{ef} - D\bar{U}^{-}_{ef} \qquad \forall\, e, f \qquad (14)$$

$$\underline{S}_{ep(f)} - \underline{S}_{ef} = D\underline{S}^{+}_{ef} - D\underline{S}^{-}_{ef} \qquad \forall\, e, f \qquad (15)$$

$$\underline{\bar{S}}_{ep(f)} - \bar{S}_{ef} = D\bar{S}^{+}_{ef} - D\bar{S}^{-}_{ef} \qquad \forall\, e, f \qquad (16)$$

$$\bar{U}_{if} \leq \underline{U}_{jf} \qquad \forall\, (i,j)\ \varepsilon\ N_{ij\backslash f} \ :\ \forall\, f \qquad (17)$$

$$\bar{S}_{jf} \leq \underline{S}_{if} \qquad \forall\, (i,j)\ \varepsilon\ N_{ij/f} \ :\ \forall\, f \qquad (18)$$

The objective function minimizes the weighted sum of material flow and layout rearrangement over some future scenarios. Constraints 2 through 4 define cell shapes. 5 and 6 contain I/O constraints and cell-within-zone-within- buildings constraints are in 7 and 8. Constraints 9 and 10 define interstation distance computation. Constraints 11 through 16 relate to interfuture cell displacement constraints. Finally, constraints 17 and 18 are design skeleton based relative positioning constraints. The future scenarios have probabilities associated with them and are part of a scenario tree [28]. The weights are a combination of these probabilities and factors such as the duration of the future scenario and discounted value of money. Since the formulation does not guarantee the noninterference of cells, initial skeletons are proposed. So, the relative positions of the departments do not change. Only the shape and sizes of the departments change. As well the size of the facility may also change. While these initial skeletons may appear to restrict the model, the authors suggest an interactive approach in which different skeletons for the different futures can be proposed. This is possible as the model is linear and thus large problems can be solved. In their experiments, solving a twelve department, seven future case required a maximum of 252 CPU seconds on a SUN SPARC workstation. The model gives the resulting optimal layout for each future scenario in the scenario tree. As in decision trees, the authors suggest testing the robustness by changing factors such as the probabilities and the structure of the tree. They also suggest that this model is best used as a tool within other generalized layout design procedures.

Most of the algorithms in dynamic layout either assume equal department sizes, or if they handle different department sizes, require the decision maker to provide skeleton layouts. Lacksonen [19] proposes a two-stage algorithm that incorporates the advantages of both the formulations discussed above. Stage 1 of his procedure involves solving the equal sized department formulation. A cutting plane and exchange routine is used to determine the relative locations of layouts in each period. The exchange routine reduces cost by swapping pairs of departments for blocks of consecutive time periods by considering material flows as well as layout rearrangement costs. Departments that are stationary in this stage are required to remain stationary in the second stage also. By setting the relative positions of the layouts, the rearrangement costs are completely defined in this stage. In Stage 2, these departments are modified to give the various sizes and shapes as required for each period's data. A modified version of the formulation of Montreuil and Venkatadri is used. This involves using a mixed integer linear programming model to minimize flow costs. The integer variables are used to define the non-overlapping constraints. Other constraints define area requirements and stationary departments. Improved linearizations for non-linear departmental area constraints and improved departmental overlapping constraints over previous formulations are provided. Most of the tests were performed on the static version of the model as existing problems deal mainly in that area. Tests on twelve department problems proved the ability of the proposed two-stage algorithm to provide good solutions. Time constraints prevented the algorithm from finding solutions to problems with more than twelve departments. Since no other similar algorithm is available to solve the dynamic layout problem, no comparisons were possible. Again, due to time constraints, the largest problem solved had only twelve departments and three time periods.

Thus, the model by Lacksonen addresses the deficiencies of the formulations in [4] and [24]. Departments of unequal sizes can be accommodated and there is no need to specify an initial skeleton layout. However, it is still restricted as the relative positions determined in the first stage cannot be changed in the second stage, as well as problem size. So the formulations discussed in this section illustrate the difficulty in obtaining optimal solutions for realistic problems in dynamic layout. Therefore, heuristic (suboptimal) algorithms play an important role in dynamic layout.

5. Conclusion

In the past decade some issues regarding the DFLP have been addressed by various researchers. This paper attempted to categorize them. The advantages, disadvantages and importance of each research were also discussed in this paper. A summary is given in Table 2. Different basic models for equal and unequal department sizes were presented. The survey shows that given the intractability of the problem, heuristics are important in solving realistic problems. Further, research incorporating uncertainty in material handling flow data was also considered. In these situations, one can use the □robustness□ approach or the □scenario of futures□ approach. It is important to incorporate uncertainty since many dynamic layout situations are likely to be uncertain. Opportunities also lie in simulating these uncertain environments under a general manufacturing or service organization framework. The only simulation paper discussed dealt with FMSs. Finally as suggested while discussing the different research, opportunities exist for creating hybrids of the different models. This will allow researchers to combine the advantages of the different models.

Acknowledgements

The authors are thankful to the Natural Sciences and Engineering Research Council (NSERC) of Canada for financial support. The comments of the referees were also very helpful.

Category		**Algorithm**	**Authors**
Equal Size Departments	Deterministic Material Flow	Dynamic Programming	Rosenblatt (1986) Balakrishnan et al. (1992) Urban (forthcoming)
		Pairwise-Interchange	Urban (1993)
		Genetic	Conway and Venkataramanan (1994)
		Tabu-search	Kaku and Mazzola (forthcoming)
		Comparison, Branch and Bound	Lacksonen and Enscore (1993)
		Bounds and Fathoming	Sweeney and Tatham (1976) Batta (1987) Urban (1992) Balakrishnan (1993) Urban (forthcoming)
	Stochastic Material Flow	Branch and Bound, Robustness	Kouvelis et al. (1992)
		Markov Processes	Kouvelis and Kiran (1991)
		Simulation	Afentakis et al. (1990)
Unequal Size Departments		Linear Programming	Montreuil and Venkatadri (1991) Montreuil and LaForge (1992)
		Mixed Integer Programming	Lacksonen (1994)

Table 2

Summary of the Different Algorithms in the Dynamic Facility Layout Problem

References

1. Afentakis P, Millen RA and Solomon MM (1990) Dynamic layout strategies for flexible manufacturing systems, *International Journal of Production Research*, 28, 2, 311-323.

2. Armour GC and Buffa ES (1963) A heuristic algorithm and simulation approach to relative allocation of facilities. *Management Science*, 9, 2, 294-300.

3. Balakrishnan J (1993) The dynamics of plant layout. *Management Science*, 39, 5, 654-655.

4. Balakrishnan J, Jacobs FR and Venkataramanan MA (1992) Solutions for the constrained dynamic facility layout problem. *European Journal of Operations Research*, 57, 2, 280-286.

5. Ballou RH (1968) Dynamic warehouse location analysis. *Journal of Marketing Research*, 5, 271-276.

6. Batta R (1987) The dynamics of plant layout. *Management Science*, 33, 8, 1065.

7. Bukard, RE and Bonniger T (1983) A heuristic for quadratic Boolean program with applications to quadratic assignment problems. *European Journal of Operational Research*, 13, 374-386.

8. Conway DG and Venkataramanan MA (1994) Genetic search and the dynamic facility layout problem. *Computers & Operations Research*, 21, 8, 955-960.

9. Davis SM *(1987) Future Perfect*, Addison-Wesley, Reading, MA.

10. Gilmore PC (1962) Optimal and suboptimal algorithms for the quadratic assignment problem. *Journal of the Society for Industrial and Applied Mathematics*, 10, 305-313.

11 Goetschalckx M (1992) An interactive layout heuristic based on hexagonal adjacency graphs. *European Journal of Operations Research*, 63, 2, 304-321.

12. Graves GW and Whinston AB (1970) An algorithm for the quadratic assignment problem. *Management Science*, 17, 7, 453-471.

13. Hirano H (1989) *JIT Factory Revolution*, Productivity Press, Cambridge MA.

14. Kaku BK and Mazzola JB A tabu-search heuristic for the dynamic plant layout problem. *INFORMS Journal on Computing*. (forthcoming)

15. Koopmans TC and Beckman M (1957) Assignment problems and location of economic activities. *Econometrika*, 25, 53-76.

16. Kouvelis P, Kurawarwala AA and Gutierrez GJ (1992) Algorithms for robust single and multiple period layout planning for manufacturing systems, *European Journal of Operations Research*, 63, 287-303.

17. Kouvelis P and Kiran AS (1991) Single and multiple period layout models for automated manufacturing systems, *European Journal of Operations Research*, 52, 300-314.

18. Kusiak A and Heragu SS (1987) The facility layout problem. *European Journal of Operational Research*, 29, 229-251.

19. Lacksonen TA (1994) Static and dynamic layout problems with varying areas. *Journal of the Operational Research Society*, 45, 1, 59-69.

20. Lacksonen TA and Enscore EE (1993) Quadratic assignment algorithms for the dynamic layout problem. *International Journal of Production Research*, 31, 3, 503-517.

21. Meller RD and Gau KY (1996) The facility layout problem: recent and emerging trends and perspectives, *Journal of Manufacturing Systems*, 15, 5, 351-366.

22. Montreuil B and Ratliff HD (1989) Utilizing cut trees as design skeletons for facilities layout. *IIE Transactions*, 21, 2, 136-143.

23. Montreuil B and Venkatadri U (1991) Strategic interpolative design of dynamic manufacturing systems layouts. *Management Science*, 37, 6, 682-694.

24. Montreuil B and LaForge A (1992) Dynamic layout design given a scenario tree of probable futures. *European Journal of Operations Research*, 63, 271-286.

25. Mote J, Murthy I, and Olson DL (1988) On solving singly constrained shortest path problems by generating pareto-optimal paths parametrically. Working Paper, Louisiana State University, Baton Rouge, Louisiana.

26. Page AL (1991) PDMA new product development survey: performance, and best practices. Paper presented at PDMA Conference, Chicago, Nov. 13.

27. Pardolos PH and Crouse JV (1989) A parallel algorithm for the QAP. *Proceedings of the 1989 Supercomputer Conference*, ACM Press, New York, 351-360.

28. Raiffa H (1968) *Decision Analysis*, Addison-Wesley, Reading MA.

29. Rosenblatt MJ (1986) The dynamics of plant layout. *Management Science*, 32, 1, 76-86.

30. Rosenblatt MJ and Lee HL (1987) A robustness approach to facilities design, *International Journal of Production Research*, 25, 4, 479-486.

31. Sahni S and Gonzales T (1976) P-complete approximation problem *Journal of the ACM*, 23, 3, 555-565.

32. Sweeney DS and Tatham RL (1976) An improved long run model for multiple warehouse location. *Management Science*, 22, 7, 758-758.

33. Urban TL (1992) Computational performance and efficiency of lower-bound procedures for the dynamic facility layout problem. *European Journal of Operations Research*, 57, 2, 271-279.

34. Urban TL (1993) A heuristic for the dynamic facility layout problem. *IIE Transactions*, 25, 4, 57-63.

35. Urban TL Solution procedures for the dynamic facility layout problem. *Annals of Operations Research*. (forthcoming)

36. Wesolowsky GO (1973) Dynamic facility location, *Management Science*, 19, 11, 1241-1248.