Locating Input and Output Points in Facilities Design—A Comparison of Constructive, Evolutionary, and Exact Methods

Rifat Aykut Arapoglu, Bryan A. Norman, and Alice E. Smith, *Senior Member, IEEE*

***Abstract*—This paper formulates and compares four new approaches to optimally locate the input and output station for each department within a facility design such that material handling costs are minimized. This problem is an NP-hard combinatorial problem with many real-life applications of considerable economic consequence. A genetic algorithm (GA) is shown to be an effective and efficient optimization method when compared to integer programming, simulated annealing, and three versions of a greedy constructive heuristic on a suite of test problems of varying size. Seeding versus random initialization of GA populations are compared.**

***Index Terms*—Genetic algorithms, materials handling, optimization methods.**

I. INTRODUCTION TO THE PROBLEM

THE FAMILY of facility design problems involve the partitioning of a planar region into departments or work centers of given area so as to minimize the costs associated with projected interactions between departments. These costs usually reflect material handling costs among departments based on the volume of material handling, distance to be transported, and cost per unit distance. The material moves from the input–output (I/O) point of a department to the I/O point of the next department on the routing specification. Generally, the routings for each product or batch will be different, and each product may only visit a subset of the departments or work stations within a facility. Such problems occur in many organizations, including manufacturing cell design [3], hospital design [11], land-use planning [6], and construction-site management [29]. For U.S. manufacturers, between 20% and 50% of total operating expenses are spent on material handling and an appropriate facilities design can reduce these costs by at least 10% to 30% [19]. Dr. J. A. Tompkins, a seminal researcher in the field, recently wrote, "Since 1955, approximately 8% of the U.S. GNP has been spent annually on new facilities. In addition, existing facilities must be continually modified... These issues represent more than \$250 billion per year attributed to the design of facility systems, layouts, handling systems, and facilities locations" [27]. Altering facility designs due to incorrect decisions, forecasts, or assumptions usually involves considerable cost, time, and disruption of activities. On the other hand, good designs can reap economic and operational benefits for a long time period. The problem studied primarily in the literature has been "block layout," which only specifies the placement of the departments without regard for aisle structure, material handling system, machine placement within departments, or I/O station locations. Block layout is usually a precursor to these subsequent design steps, which are termed "detailed layout." Two recent survey articles on the facility design problem are Kusiak and Heragu [18] and Meller and Gau [19].

Because of the computational complexities in optimizing multiple and nonlinear objectives and constraints, only limited work has been done beyond block layout. The recent work of Benson and Foote [5] considers the placement of aisles and I/O locations after the relative location of the departments and the general aisle structure have been selected. Kim and Klein [17] consider the problem of placing I/O locations for a given automated guided vehicle network. They propose constructive heuristic methods for determining the I/O locations.

Additional related work on integrated facility design that considers machine placement within departments includes papers by Nagi and others [14], [16]. This work uses predefined departmental shapes set on a grid covering the facility space. In [14], Dijkstra's shortest path algorithm is used to calculate the rectilinear distance to and from prespecified I/O locations. In [16], I/O points are placed during the optimization and a constraint is imposed to encourage aisles that are straight. Both papers use a simulated annealing (SA) heuristic to alter departmental placement. Another related work is by Banerjee *et al.* [3], where a genetic algorithm (GA) finds a "rough" design that is then defined fully using a subordinate mathematical programming routine. The number of I/Os per department is prespecified and then they are located optimally with the department placement. The rectilinear distance is calculated between I/O points.

This paper focuses on the problem of optimal location of an I/O for each department given a specific block layout. The optimization approaches described are intended to work as subroutines for other optimization algorithms that search through block layouts, optimally locating the I/Os as a nested routine to evaluate the objective function (i.e., minimize material handling costs) of the total facility design. Because of the intended integrated nature of the optimization, computational effort versus accuracy becomes the most important point, an aspect that has

Manuscript received January 25, 1999; revised August 2, 1999, March 20, 2000, and March 21, 2000. This work was supported in part by the National Science Foundation under Grant DMI 95-02134 and Grant DMI 99-08322.

R. A. Arapoglu and B. A. Norman are with the Department of Industrial Engineering, University of Pittsburgh, Pittsburgh, PA 15261 USA (e-mail: banorman@engrng.pitt.edu).

A. E. Smith is with the Department of Industrial and Systems Engineering, Auburn University, Auburn, AL 36849 USA (e-mail: aesmith@eng.auburn.edu).

Publisher Item Identifier S 1089-778X(01)05372-3.

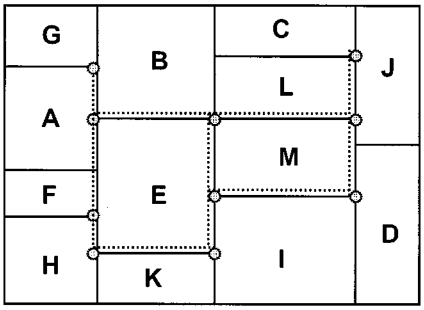


Fig. 1. Flexible bay design with 13 departments, ten distinct I/O locations, and material travel along department perimeters (dashed lines).

not been well studied by the papers cited above. Since the I/O optimization will be done many hundreds or thousands of times during facility design, computationally expensive approaches such as integer programming (IP) are prohibitive. The results of this paper show that using an improvement version of a constructive heuristic for most of the search and sending the most promising solutions periodically to the GA properly balances the goals of computational practicality with optimization precision.

As in the papers mentioned above, material flow both in and out of a given department is assumed to occur at the same physical point. However, the heuristic methods described in this paper can be generalized straightforwardly to separate the input point from the output point for specified departments. This will add constraints to the optimization problem and increase the search space; it is a subject of continuing research by the authors.

II. PROBLEM FORMULATION

In this paper, it is assumed that a block layout has been constructed, where all of the departments have rectangular shapes within a rectangular building (see Fig. 1). The two most general mechanisms in the literature for constructing such layouts are the flexible bay [23], [26] and the slicing tree [8], [10], [13], [15], [24], [25]. However, these structures are not necessary for the approaches described here. Subsets, such as the popular quadratic assignment problem (QAP) approach to block layout, can be handled in the same manner.

In this paper, *contour distance*[1] is used [7], [20]–[22]. Traditionally, rectilinear and Euclidean distance measures have been used in layout design problems. Neither of these measures is capable of depicting the real distances experienced by material movement in a facility as they assume centroid to centroid rectilinear/Euclidean distances and permit material movement through other departments (see Fig. 2). The contour distance metric, on the other hand, only permits travel along the boundaries of the departments and follows the shortest path connecting a given pair of departments from and to the I/O points, as shown in Fig. 1.

The single I/O point placement problem, i.e., one I/O per department, in facilities design can be described as a minimization problem that uses the notation below.

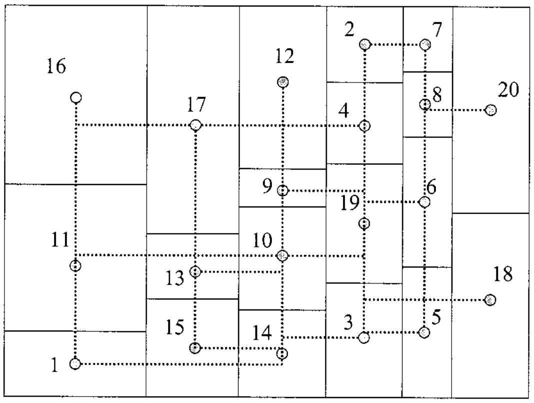


Fig. 2. Block layout design of 20 departments showing the impracticalities of using the rectilinear distance metric between departmental centroids.

Notation:

$X_{i,k}$	Binary variable that equals 1 if department i uses I/O candidate location k, zero otherwise.
$Z_{i,k,j,l}$	Binary variable that equals 1 if department i uses I/O candidate location k and department j uses I/O candidate location l, zero otherwise. It indicates which path is used for moving material from department i to department j.
$d_{i,k,j,l}$	Distance from candidate I/O location k indepartment i to candidate I/O location l in department j.
$d\_\mathrm{cost}_{i,j}$	Distance from department i to department j using the I/O that is assigned for each department.
Loc_i	Set of candidate I/O locations for department i.
$f_{i,j}$	Flow (material quantity times cost per unit distance) between departments i and j.
N	Number of departments in the layout.

The objective of the optimization problem is to minimize the product of flow times distance for the layout. The flow amount is fixed but the flow distances traveled between the departments depend on where the I/O locations are placed in each department. The I/O location problem can be represented as the following integer program:

$$\min \sum_{i=1}^{N} \sum_{j=1}^{N} f_{ij} d\_\mathrm{cost}_{ij} \tag{OF}$$

subject to an assignment constraint that requires each department to have exactly one I/O location

$$\sum_{k \in \mathrm{Loc}_i} x_{ik} = 1 \quad \forall i = 1, \ldots, N \tag{C1}$$

and cost calculation constraints

$$\sum_{k} \sum_{l} z_{i,k,j,l} = 1 \quad \forall i = 1, \ldots, N, \quad j = 1, \ldots, N \tag{C2}$$

$$z_{i,k,j,l} \le x_{i,k} \quad \forall i = 1, \ldots, N, \quad j = 1, \ldots, N, \quad k \in \mathrm{Loc}_i, \quad l \in \mathrm{Loc}_j \tag{C3}$$

$$\sum_{k} \sum_{l} z_{i,k,j,l} d_{i,k,j,l} = d\_\mathrm{cost}_{i,j} \quad \forall i = 1, \ldots, N, \quad j = 1, \ldots, N. \tag{C4}$$

---

[1] Also called "free flow" in some papers.

Constraint (C2) ensures that a path is selected for moving material from department i to department j. Otherwise, the objective function would force $z_{i,k,j,l}$ to be zero for all k, l pairs, where $k \in \mathrm{Loc}_i$ and $l \in \mathrm{Loc}_j$, no path would be selected, and the distance would be zero. Constraint (C3) permits $z_{i,k,j,l}$ to equal one only if both $x_{i,k}$ and $x_{j,l}$ equal one (the material flow path can only go from I/O location k of department i to I/O location l of department j if k and l were chosen as the I/O locations of departments i and j, respectively). Constraint (C4) evaluates the distance of the path from department i to department j. Note that (C2) to (C4) could be replaced with the single alternative (CA) given below, but this constraint is nonlinear and integer, which would make the resulting math program extremely difficult to solve

$$\sum_k \sum_l x_{i,k} x_{j,l} d_{i,k,j,l} = d\_\mathrm{cost}_{i,j} \quad \forall i = 1, \ldots N, \quad j = 1, \ldots N. \tag{CA}$$

III. COMPLEXITY OF THE SINGLE I/O PLACEMENT PROBLEM

The following two observations from [21] make possible the design and analysis of the single I/O point location problem.

1) The total number of potential I/O points can be reduced to the finite set of departmental intersection points. Consider the example in Fig. 3. The candidate I/O locations for department B are locations 1– 3 and 8–10.[2]
2) There can be at most $2N - 2$ such points for a flexible bay block layout.

Note that more than one department may share the same I/O location as in departments C and J of Fig. 1. Since each department may use only one of its I/O points independently of the other departments, the total number of possible combinations grows exponentially, creating an NP-hard problem [12], namely,

$$\prod_{i=1}^{N} |\mathrm{Loc}_i|.$$

For the layout in Fig. 3, this quantity is 2.9×10^7. This number is on the order of 10^{13} for a facility with 20 departments.

IV. OPTIMIZATION METHODOLOGIES

It is assumed that a block layout design with all rectangular departments located within a rectangular floorplan has already been identified and the optimization task at hand is to choose a single I/O location for each department such that material handling costs are minimized using the contour distance metric. The shortest paths between all pairs of candidate I/O points, which are located at the intersection points of departments (see Fig. 3), need to be found. This is accomplished by creating a network with nonnegative arc lengths that represent the distances between the I/O points using the perimeter distance metric. The nodes of the network include all of the potential I/O locations, which are found at the intersection points of the departments, and a node for each one of the departments. The Floyd-Warshall algorithm [1] is used to find the shortest distances between all pairs. This algorithm is polynomial and runs in $O(n^3)$ time, where n is the number of network nodes.

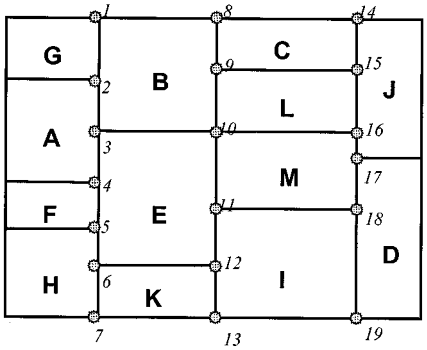


Fig. 3. Candidate I/O locations for a flexible bay layout.

To locate the optimum I/O point for each department, four approaches were developed and compared. An exact approach to solve this problem is through the use of IP. This always finds the optimum solution, but the search space grows exponentially with the number of candidate I/O locations, thereby precluding its use for medium to large sized facilities. A computationally expedient approach is a constructive heuristic that works in a greedy, or myopic, fashion. Three versions of the constructive heuristic were developed and tested. A third alternative is an SA heuristic using a neighborhood structure to evaluate moves. The fourth alternative uses GAs to improve upon a set of feasible, but randomly chosen I/O locations. A lower bound was also devised by considering the linear relaxation of the mixed integer program.

A. IP Approach

The formulation developed in Section II was coded as an IP and solved using CPLEX [9].

B. Constructive Heuristic

Another approach is a very quick constructive heuristic that operates in a greedy fashion. There are three versions of this heuristic. Version A is a deterministic constructive version, version B is the same constructive heuristic, but is not deterministic, and version C uses the solution from version A as the initial point for perturbations (i.e., version C is an improvement heuristic that is seeded with version A).

Version A is deterministic and assigns the I/O points in order of the magnitude of the material flows. The pseudocode is below.

Additional Notation:

A	Set of assigned departments.
$S = \{1, 2, 3, \ldots, N\}$	Set of all departments.
$\mathrm{tot}\, f_{i,k}$	Total material flow using I/O point k of department i.

---

[2] While the optimal location of I/O points will always be at departmental intersections, it may not be physically practical, in some cases. A perturbation along the relevant department perimeter near the optimal intersection will maintain near optimality of the cost objective while being physically tractable.

Algorithm:

```
Step 1:
  $A = \emptyset$
  Set $\mathrm{tot}\, f_{i,k} = 0$ for all $i \in S$ and $k \in \mathrm{Loc}_i$
Step 2:
  Let $\mathrm{tot}\, f_{r,s} = \mathrm{Max}\{\mathrm{tot}\, f_{i,k} : i \in S\backslash A, k \in \mathrm{Loc}_i\}$
  $A \leftarrow A \cup \{r\}$. Assign I/O point $s$ to department $r$.
  If $A = S$ then stop, all departments have been assigned an I/O point.
  Else
    $\mathrm{tot}\, f_{i,k} = 0$ for all $i \in \mathrm{S}\backslash\mathrm{A}$ and $k \in \mathrm{Loc}_i$
    Using the Floyd-Warshall algorithm, find the shortest paths between all node pairs in the network consisting of all candidate I/O locations for all departments in $S\backslash A$ and the selected I/O locations in all departments in $A$.
    For each flow, $f_{i,j}$ that uses I/O point $k$ of department $i$ and I/O point $l$ of department $j$, update $\mathrm{tot}\, f_{i,k} = \mathrm{tot}\, f_{i,k} + f_{i,j}$ and $\mathrm{tot}\, f_{j,l} = \mathrm{tot}\, f_{j,l} + f_{i,j}$
```

Version B is identical to version A except for the first item of Step 2. Rather than being a deterministic greedy selection (Max), the selection is made according to a probability. First, the maximum total flow I/O point for each department is identified. These are sorted according to flow and those below the mean are deleted. Probabilities are assigned to the remaining maximum flow I/Os and the next I/O assigned is selected based on a uniform random number and these probabilities.

Version C takes the solution from version A and perturbs the I/O locations, department by department, for each department. The best perturbation is taken as the move. The procedure stops when there is no improving move possible. While slower than either version A or B (which are fully constructive), this improvement version is nonetheless very quick.

C. SA

As a baseline for the GA described in the next section, an SA approach was coded. Starting with a random solution, a move to a neighboring solution is considered. The move space consists of moving one I/O point from its location to another feasible location. Improving moves are always accepted while nonimproving moves are accepted with

$$\Pr(\text{accept}) = \mathbf{e}^{-\left(\frac{\Delta \mathrm{OF}}{T}\right)}$$

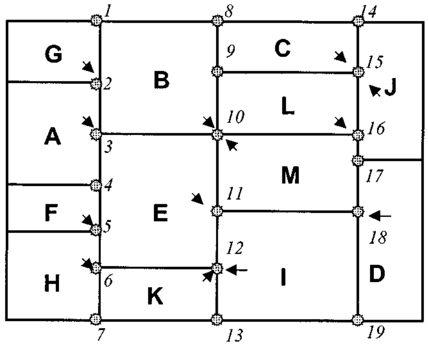


Fig. 4. I/O locations for an example GA chromosome.

where T is the temperature and $\Delta\mathrm{OF}$ is the increase in objective function value. The temperature changes with a geometric cooling schedule

$$T_{\mathrm{new}} = T_{\mathrm{old}} * 0.90$$

with an initial temperature of 100 and a final temperature of 10. The number of moves per temperature is constant during the search and ranges from 1000 to 1400, depending on problem size. The SA parameters were selected after experimenting with a wide range of initial temperatures (100 to 150), cooling schedules (0.90 to 0.99), and number of moves per temperature (300 to 1400), whilst keeping the total number of moves approximately the same. The SA was designed to require almost the same central processing unit (CPU) time as the GA so a direct comparison could be made.

D. GA

The overall flowchart of the GA is given in Appendix I. The next sections discuss the particulars of the method.

1) Encoding and Fitness: A permutation encoding was used to represent a given allocation of I/O locations to the departments. Each candidate I/O location was numbered (to a maximum of $2N - 2$) and it was possible that two neighboring departments might use the same *location* as their I/O. In that case, the same I/O number was used to represent both I/Os. The example below denotes the encoding corresponding to the layout given in Fig. 4 using the I/O numbering of Fig. 3. (See the data at the bottom of the page.) The fitness of a given solution is the cost of the material flow [i.e., (OF)] between each pair of departments using the selected I/O locations and the contour distance metric.

2) Selection, Crossover, Mutation: The GA mechanisms were chosen based on experimentation and although this set provides superior performance, many other GA parameters and evolution mechanisms will provide similar performance. An

---

Department	A	B	C	D	E	F	G	H	I	J	K	L	M
I/O Locations	3	10	15	18	11	5	2	6	12	15	12	16	10

TABLE I
COMPUTATIONAL TEST RESULTS

Problem	Structure #	Lower Bound	Integer Program	CS – Version A	CS – Version B	CS – Version C	CS – Version D	Best SA [@]	Best GA [@]	GA CV [@]
Van Camp et al. 10	FB	6214.94	7239.03	11118.31	11118.31	10007.85	10007.85	7239.03	7239.03	0.00
Bazaraa 12	FB	7116.79	8308.03	8692.03	8692.03	8308.03	8308.03	8308.03	8308.03	0.03
Bazaraa 14 –1	FB	2320.71	4223.14	5648.36	5648.36	4793.56	4793.56	4223.14	4223.14	0.00
Bazaraa 14 - 2	FB	4114.61	4991.78	5459.22	5459.22	5187.78	5187.78	4991.78	4991.78	0.02
Armour & Buffa 20 - 1	FB	223.17	344.82	388.59	388.59	357.02	358.09	352.52	345.89	0.01
Armour & Buffa 20 - 2	FB	202.19	334.94	371.20	364.43	345.68	334.94	341.81	334.94	0.00
Armour & Buffa 20 - 3	ST	391.39	461.42	545.28	545.28	497.32	497.32	463.16	461.42	0.00
Armour & Buffa 20 - 4	ST	315.47	441.87	549.56	549.56	490.45	442.76	444.67	441.87	0.00
Tam 30 -1	ST	12757.98	*	17100.97	17100.97	15314.14	15314.14	15314.14	15314.14	0.00
Tam 30 -2	ST	12846.20	*	16515.81	16515.81	15863.49	15863.49	15853.01	15853.01	0.00
Tam 30 -3	ST	2728.75	*	3819.43	3819.43	3459.11	3427.32	3354.22	3346.52	0.01
Tam 30 -4	ST	2920.04	*	4383.53	4383.53	4066.18	3976.25	3753.65	3746.06	0.01
Arapoglu et al. 40 - 1	ST	22409.94	*	24015.80	24015.80	23943.40	23943.40	23942.40	23942.40	0.00
Arapoglu et al. 40 - 2	ST	22524.86	*	25735.00	25735.00	25282.50	25282.50	25282.50	25282.50	0.00
Arapoglu et al. 50 - 1	FB	24327.50	*	41985.60	41985.60	40504.40	40504.40	38333.81	38333.81	0.01
Arapoglu et al. 50 - 2	FB	36453.30	*	40192.21	40192.21	39694.41	39694.41	39694.41	39694.41	0.00
Arapoglu et al. 60	FB	45071.36	*	51449.91	51449.91	50930.23	50930.23	50935.62	50930.23	0.00

FB (flexible bay) or ST (slicing tree).

[@] Over five runs.

CV is Coefficient of Variation.

* Cannot be solved optimally because of problem size.

elite set of ten from a population of 50 was copied directly to the list of candidates to survive to the next generation. From the old generation of 50, one parent was selected using a tournament of size two while the other parent was selected randomly from the population (excluding the parent already selected via the tournament). Uniform crossover with probability of 1.0 was applied to produce one offspring, with a preference of 0.70 for alleles from the fitter parent. Note that every crossover produced a feasible allocation of I/O locations and, thus, there was no repair needed. This continued until 50 offspring were created each generation.

The offspring were subject to mutation with a probability of 0.5. Every allele was tested for mutation individually and independently with a preselected mutation probability of 0.10. If an allele was chosen for mutation, the new allele was selected randomly from the feasible I/O locations for that department, not including the current value. Again, every mutation resulted in a feasible allocation of I/O locations. A wide variety of combinations of mutation probabilities were tried with the result that the GA was very robust from mutation probabilities ranging from 0.05 to 1.0; however, 0.5 was marginally better than the others.

The 50 mutated offspring were combined with the ten elite solutions from the old population and sorted. The top 50 solutions were kept to form the new population.

3) Termination Criterion: The GA was terminated after 500 consecutive generations without any improvement in the best objective function value.

V. TEST PROBLEMS AND RESULTS

The four approaches were applied to instances of well-known test problems with either slicing tree or flexible bay structures: a ten-department problem by Van Camp *et al.* [28], a 12-department problem by Bazaraa [4] that was encoded in the GA as a 16-department problem to handle empty space within the facility as was done by Tate and Smith [26], two instances of a 14-department problem by Bazaraa [4], four instances of a 20-department test problem of Armour and Buffa [2], and four instances of a 30-department problem from Tam [24], [25]. Three new test problems with 40, 50, and 60 departments, respectively, were developed by the authors and the associated department areas and flows are available by email.

The relaxed mixed integer program (MIP) lower bound, the objective function results of the IP, the constructive heuristic versions, and a summary of five runs of the SA and the GA without seeding are compared in Table I. It can also be seen that versions A and B of the constructive heuristic were equivalent, except for problem Armour and Buffa-2. Version C improved on versions A and B every time with improvements ranging from 0.3% to 18% and an average improvement of 7%. The GA was very consistent with at least one seed finding the optimal solution for each problem instance with a known optimal solution except for the Armour and Buffa-1. The GA was also consistent across different random number seeds with almost no variation across all of the test problems. Although dominated by the GA, the SA performed well; in nine of the test problems, the best result over five runs was identical while the GA was better than the SA in eight cases. Using a nonparametric sign test, the GA was statistically significantly better than the SA.

The best solution of the GA equals or betters the constructive heuristics in every case, although the amount of improvement does not correlate with problem size. Compared to versions A and B, the GA obtained improvements of 0.3% to 53% with an average improvement of 13%. Compared to version C, the GA obtained improvements of 0% to 38% with an average improvement of 6%. The best I/O locations for each problem are given in Appendix II, as are the implied aisle structure, i.e., flowpaths.

Another relevant comparison is to equalize the computational effort of the GA with the constructive heuristics. This was done by running version B of the constructive multiple times (enough to equalize the CPU time of the GA) and comparing the result with the best solution found by using version A of the constructive and retaining the best solution. This solution was improved using the perturbation improvement method and the results are

TABLE II
COMPUTATIONAL TIME RESULTS (AVERAGE CPU SECONDS)

Problem	Structure #	CS – Version A	CS – Version B	CS – Version C	GA (or SA) 1 Run
Van Camp et al. 10	FB	0.01	0.01	0.02	2.78
Bazaraa 12	FB	0.04	0.05	0.08	5.38
Bazaraa 14 –1	FB	0.03	0.02	0.06	4.46
Bazaraa 14 - 2	FB	0.02	0.03	0.04	4.34
Armour & Buffa 20 - 1	FB	0.07	0.07	0.14	9.61
Armour & Buffa 20 - 2	FB	0.09	0.08	0.17	7.95
Armour & Buffa 20 - 3	ST	0.10	0.10	0.20	8.20
Armour & Buffa 20 - 4	ST	0.09	0.09	0.27	9.42
Tam 30 -1	ST	0.27	0.31	0.98	17.59
Tam 30 -2	ST	0.28	0.30	0.83	17.36
Tam 30 -3	ST	0.27	0.30	0.74	18.62
Tam 30 -4	ST	0.27	0.28	1.03	18.33
Arapoglu et al. 40 - 1	ST	0.67	0.66	1.25	32.32
Arapoglu et al. 40 - 2	ST	0.64	0.67	2.12	29.17
Arapoglu et al. 50 - 1	FB	0.93	0.96	2.56	54.95
Arapoglu et al. 50 - 2	FB	0.98	0.99	2.53	51.65
Arapoglu et al. 60	FB	1.83	1.90	4.94	84.44

FB (flexible bay) or ST (slicing tree)

presented in Table I as version D. When comparing this strategy to the GA, the GA found solutions that ranged from 0% to 38% better with an average improvement of 5%.

A GA version that used one copy of the solution generated by the constructive heuristic, version A, as partial seeding for the initial generation was explored. The seeding strategy did not produce any conclusive results. For most runs, the best solutions generated by the seeded and nonseeded GAs were identical. In a few cases, the seeded solution was slightly worse and in a few cases, the seeded solution was slightly better. For this problem, using this version of seeding did not seem to impact the search process.

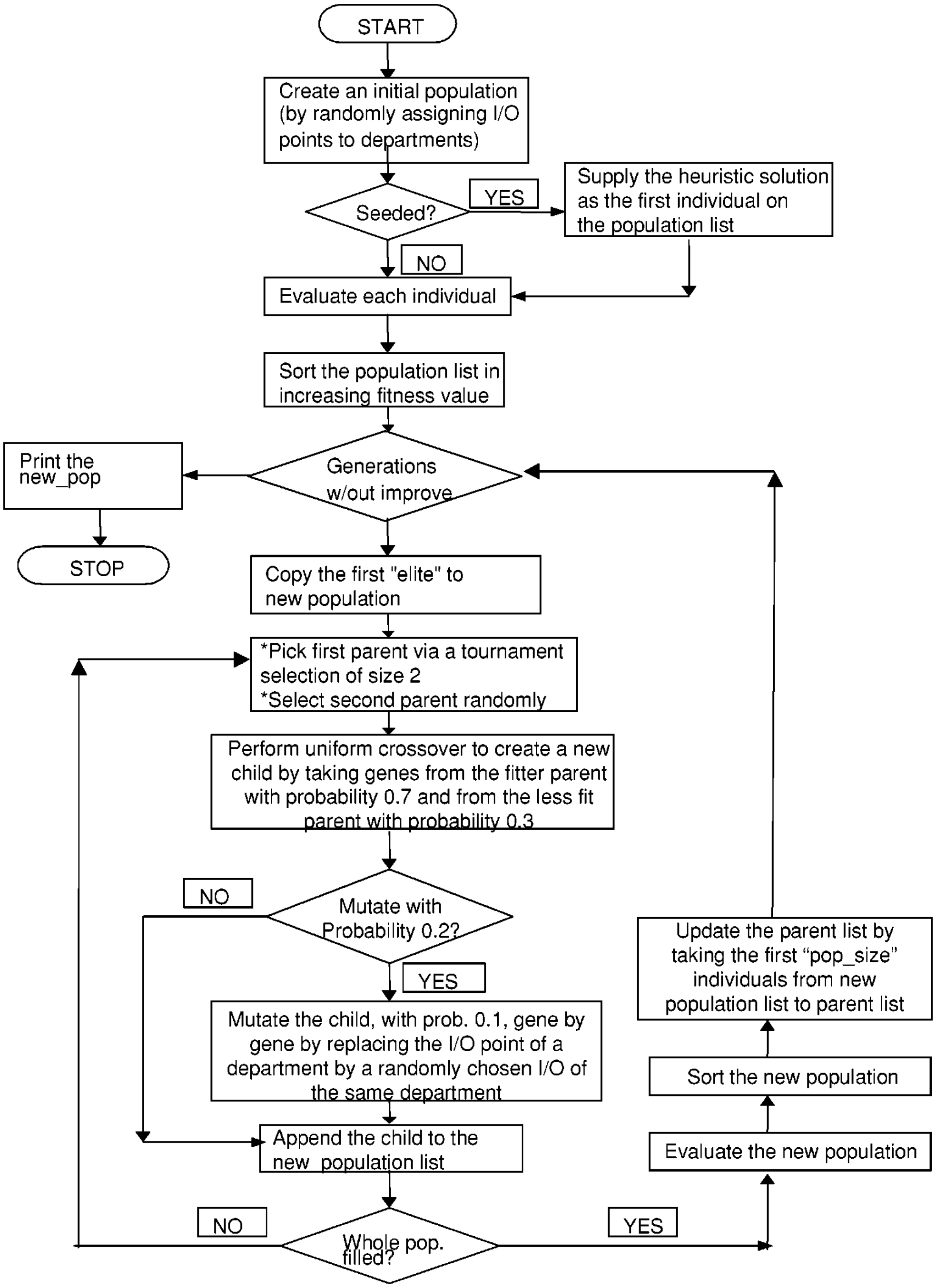
A comparison of computational effort is germane. In Table II, the CPU times in seconds on a Sun Ultra Enterprise-2 workstation with dual 200-MHz Sparc CPUs are presented for the constructive heuristic and one run of the GA (equivalently, the SA). The IP, of course, took considerably more time for each problem instance and could not finish for the larger (30 and more departments) problems. Versions A and B of the constructive heuristic were very quick. Version C was about two to five times longer than versions A or B; however, there was a marked improvement in solution quality. For the larger problems, the GA required about 20 times longer than version C of the constructive heuristic. A strategy balancing computational effort with solution quality would use version C of the constructive heuristic as an optimization subroutine for most of the block layout design optimization and use the GA as the optimization subroutine for the best few solutions of the population in the later generations.

VI. CONCLUSION

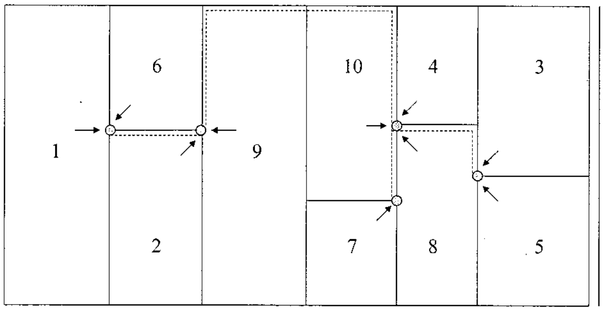
The problem of locating I/O points for a block layout also implies the paths along which material will flow in the facility. Thus, it is the most important step of detailed facility design. To improve the optimal design of facilities, it is necessary to truly integrate the block layout of departments with choosing the I/O location for each department. The GA heuristic and the constructive heuristics devised in this paper make it possible to optimize the I/O locations (and thus the flowpaths) as a subroutine to block layout optimization. It is imperative that the optimization of the I/Os be computationally expedient so that this subroutine can be done for the many thousands of block layout candidates identified during the design phase. Because of the great computational effort of the IP method, it is only practical for very small problems and the lower bound is too loose to be of much value during optimization. The constructive heuristics perform well and are very fast. The GA requires more computational effort, but is able to match or improve upon their performance in every case with improvements ranging up to 53%. For the problems where the optimal solution could be determined using the IP, the GA found solutions that were nearly always optimal. Where an optimal solution was not found, the GA identified solutions that were *at most* 8.4% worse than the optimal on any given run. While SA is an alternative to GA, its performance was worse in about half the cases while requiring the same computational effort.

Even though seeding the GA with the constructive heuristic solution is computationally trivial, it did not significantly affect the performance. For a complete optimization methodology, where both block and detailed layout factors are considered simultaneously, it makes sense to use version C of the constructive heuristic for most of the GA search of the total design space and to use the I/O GA developed in this paper for only the better population members and/or at the end of the optimization procedure. The very best few solutions may be sent to the IP for the exact I/O optimization if the design problem is not too large. There are several extensions to this work. The first, as mentioned earlier, is to consider the input location separate from the output location for specified departments. The second is to consider intradepartmental material movement that may dictate a certain departmental shape (handled by an aspect ratio or minimum side length constraint) or preferred I/O location. A preferred I/O location (for example, the I/O should be on the longer side) could be handled as a restriction (hard constraint), with a penalty (soft constraint) or even within the objective function (secondary objective). Both extensions can be handled within the same GA framework proposed although the constructive heuristic would need to be modified.

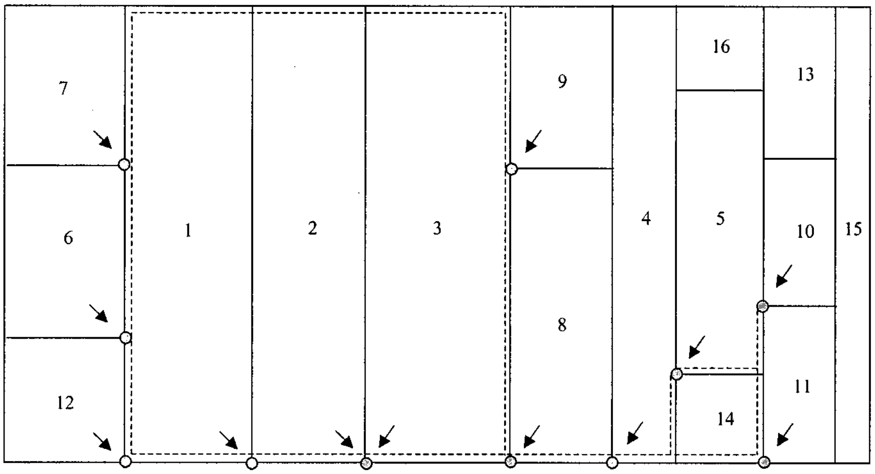
APPENDIX I
GENETIC ALGORIDHM FLOWCHART



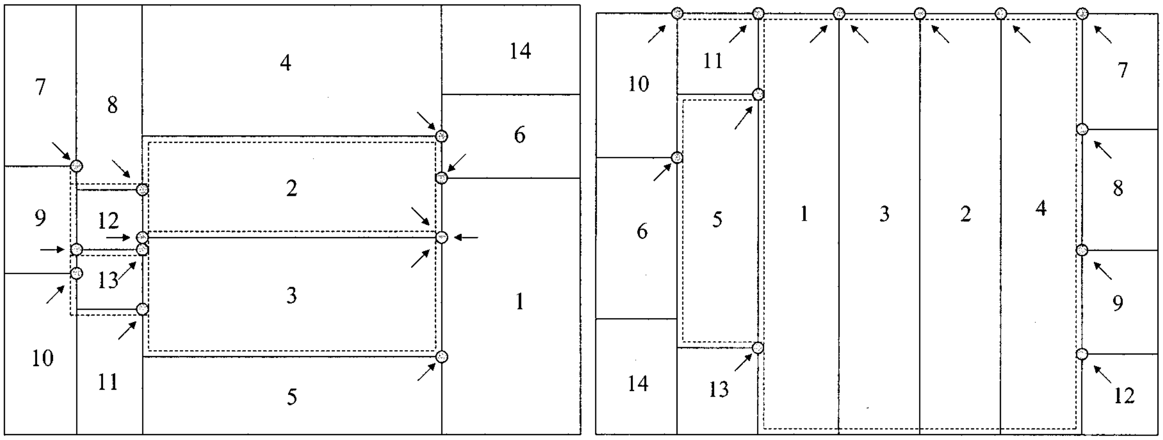
APPENDIX II
BLOCK LAYOUT, BEST I/O LOCATIONS, AND IMPLIED FLOWPATHS OF TEST PROBLEMS



Van Camp *et al.* 10.

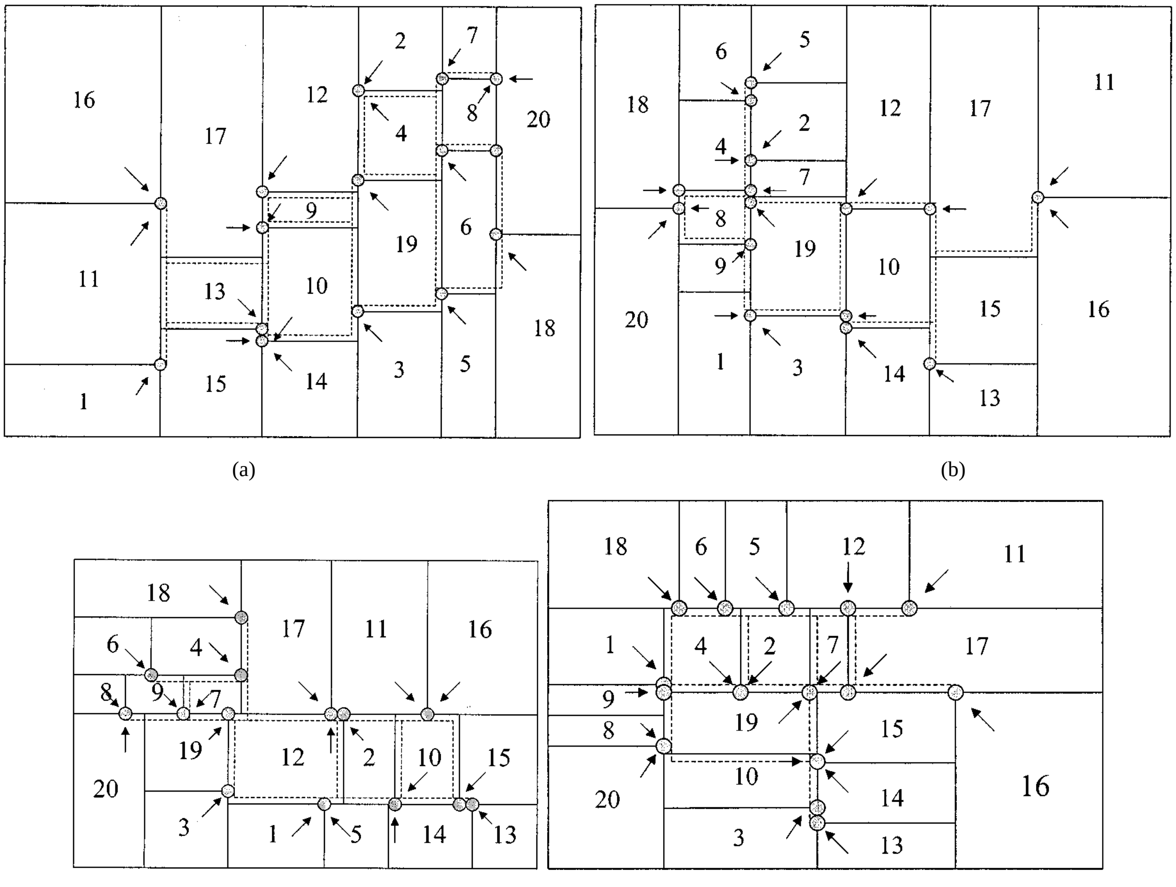


Bazaraa 12.



(a) (b)

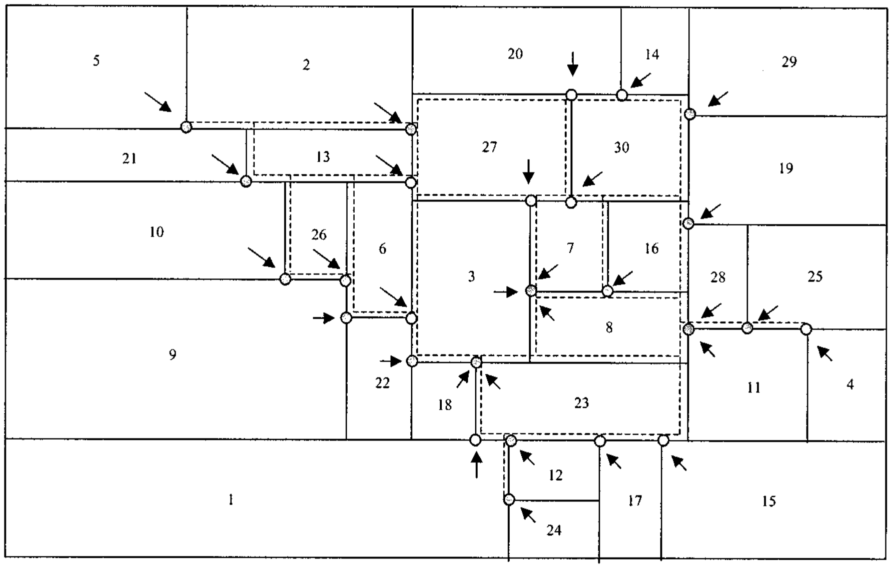
Bazaraa (a) 14–1 and (b) 14-2.



(a) (b)

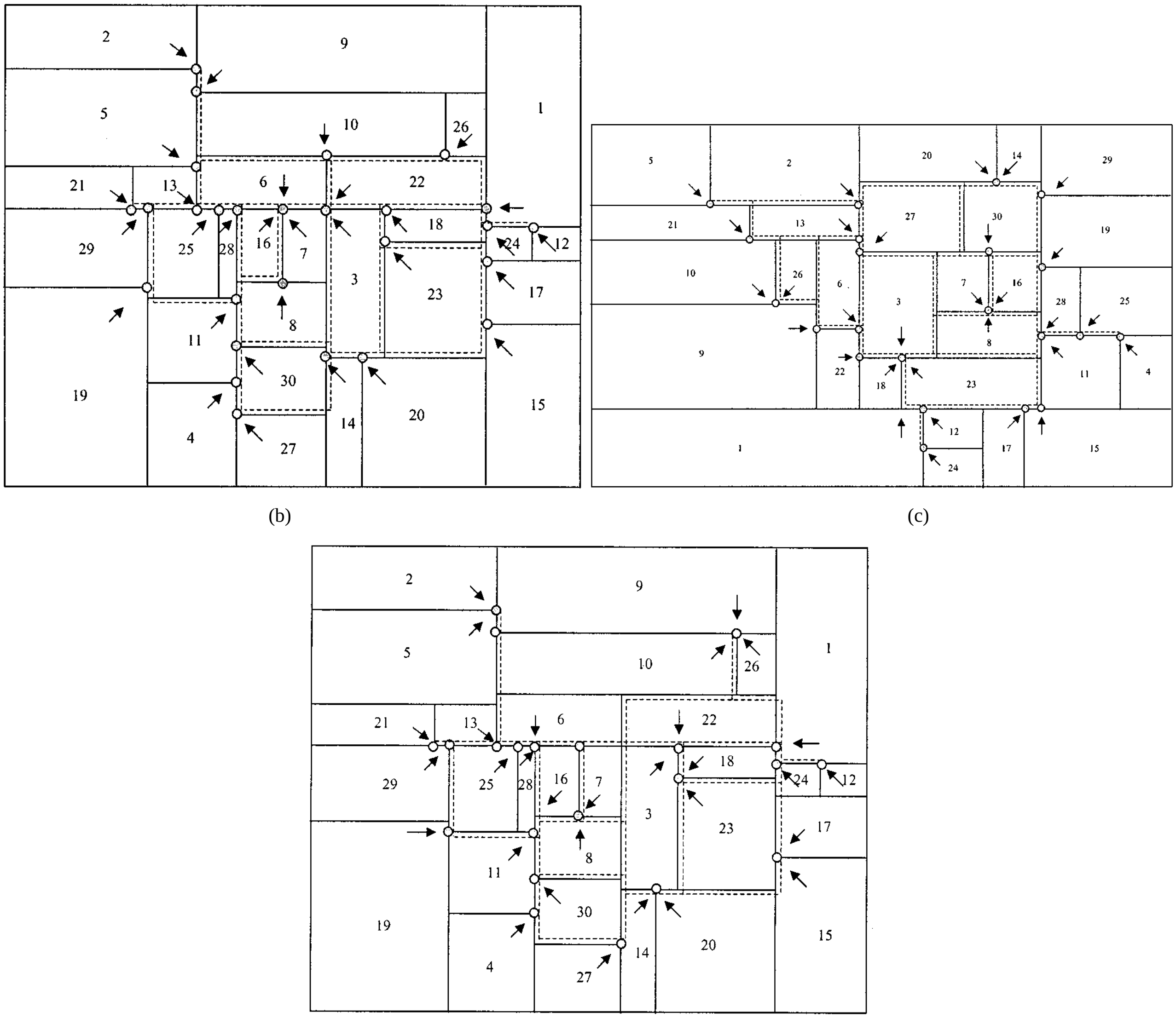
(c) (d)

Armour and Buffa (a) 20-1, (b) 20-2, (c) 20-3, and (d) 20-4.



(a)

Tam (a) 30–1.

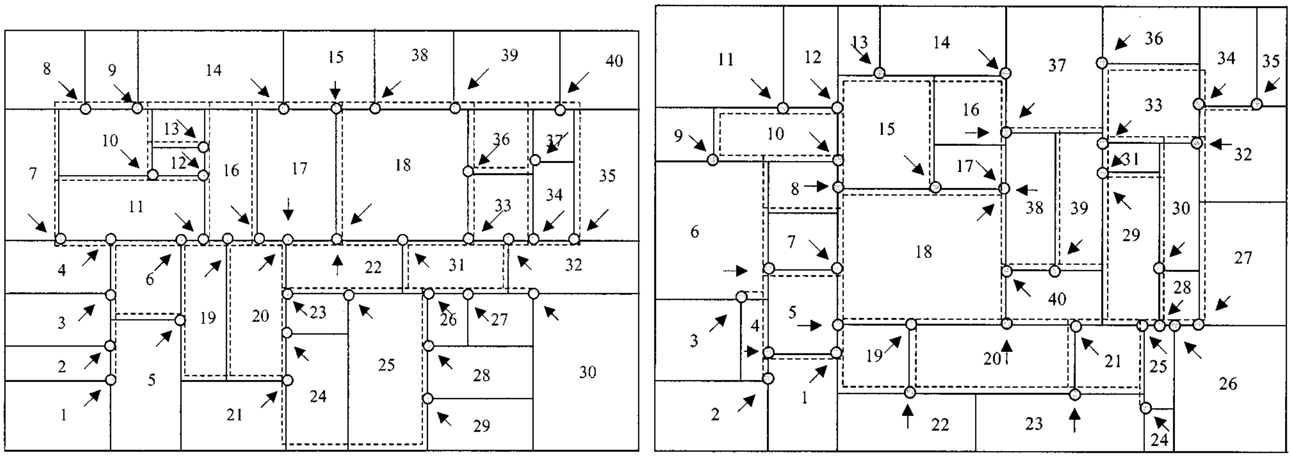


(b)

(c)

(d)

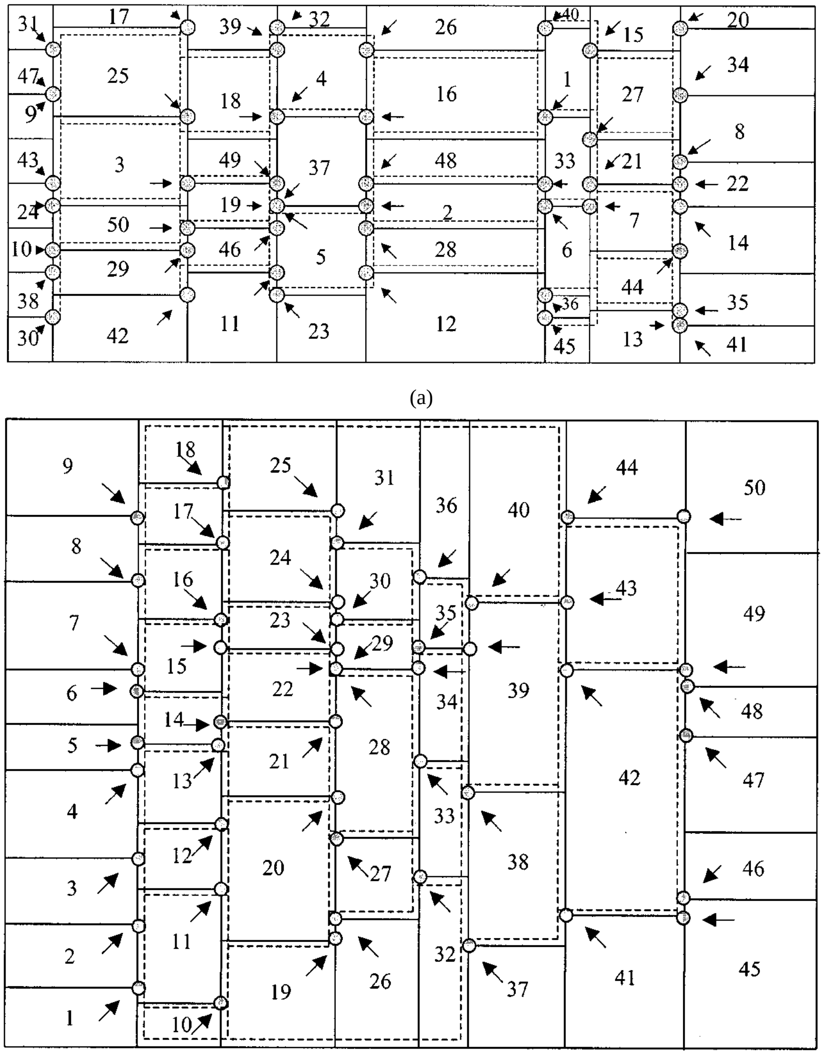
Tam (b) 30–2, (c) 30–3, and (d) 30–4.



(a)

(b)

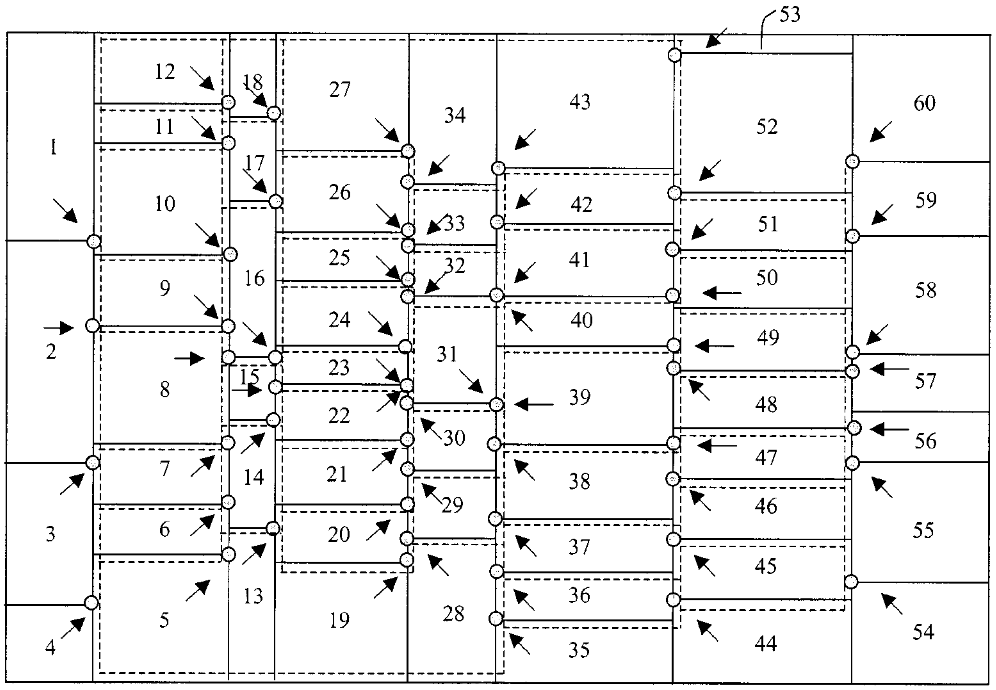
Arapoglu *et al.* (a) 40-1 and (b) 40-2.



(a)

(b)

Arapoglu *et al.* (a) 50-1 and (b) 50-2.



Arapoglu *et al.* 60.

REFERENCES

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[2] G. C. Armour and E. S. Buffa, "A heuristic algorithm and simulation approach to relative allocation of facilities," *Manage. Sci.*, vol. 9, no. 2, pp. 294–309, Jan. 1963.

[3] P. Banerjee, Y. Zhou, and B. Montreuil, "Genetically assisted optimization of cell layout and material flow path skeleton," *IIE Trans.*, vol. 29, no. 4, pp. 277–291, Apr. 1997.

[4] M. S. Bazaraa, "Computerized layout design: A branch and bound approach," *AIIE Trans.*, vol. 7, no. 4, pp. 432–438, Dec. 1975.

[5] B. Benson and B. L. Foote, "Door FAST: A constructive procedure to optimally layout a facility including aisles and door locations based on an aisle flow distance metric," *Int. J. Prod. Res.*, vol. 35, no. 7, pp. 1825–1842, July 1997.

[6] J. Bos, "Zoning in forest management: A quadratic assignment problem solved by simulated annealing," *J. Environ. Manage.*, vol. 37, no. 2, pp. 127–145, Feb. 1993.

[7] D. Chhajed, B. Montreuil, and T. J. Lowe, "Flow network design for manufacturing systems layout," *Eur. J. Oper. Res.*, vol. 57, no. 2, pp. 145–161, Mar. 1992.

[8] R. Cheng, M. Gen, and T. Tozawa, "Genetic search for facility layout design under interflows uncertainty," *Jpn. J. Fuzzy Theory Syst.*, vol. 8, pp. 267–281, 1996.

[9] CPLEX 6.0 User Documentation, CPLEX Optimization Inc.. (1999). [Online]. Available: www.cplex.com

[10] J. P. Cohoon and W. D. Paris, "Genetic placement," *IEEE Trans. Computer-Aided Design*, vol. CAD-6, pp. 956–964, Nov. 1987.

[11] A. N. Elshafei, "Hospital layout as a quadratic assignment problem," *Oper. Res. Quart.*, vol. 28, no. 1, ii, pp. 167–179, 1977.

[12] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman, 1979.

[13] J. Garces-Peres, D. A. Schoenfeld, and R. L. Wainwright, "Solving facility layout problems using genetic programming," in *Genetic Programming 1996 (Proceedings of the 1st Annual Conference—GP 96)*. Cambridge, MA: MIT Press, 1996, pp. 182–190.

[14] 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, Apr. 1996.

[15] K. Kado, P. Ross, and D. Corne, "A study of genetic algorithm hybrids for facility layout problems," in *Proc. 6th Int. Conf. Genetic Algorithms*, Pittsburgh, PA, July 1995, pp. 498–505.

[16] M. C. Kane and R. Nagi, "Integrated material flow and layout in facilities design," in *Proc. 6th Industrial Engineering Res. Conf.*, Miami, FL, May 1997, pp. 919–924.

[17] J. Kim and C. M. Klein, "Location of departmental pickup and delivery points for an AGV system," *Int. J. Prod. Res.*, vol. 34, no. 2, pp. 407–420, Feb. 1996.

[18] A. Kusiak and S. S. Heragu, "The facility layout problem," *Eur. J. Oper. Res.*, vol. 29, no. 3, pp. 229–251, May 1987.

[19] R. D. Meller and K.-Y. Gau, "The facility layout problem: Recent and emerging trends and perspectives," *J. Manuf. Syst.*, vol. 15, no. 5, pp. 351–366, 1996.

[20] B. A. Norman, A. E. Smith, and R. A. Arapoglu, "Integrated facilities layout using a perimeter distance metric," in *Parallel Problem Solving from Nature (PPSN V)*, A. E. Eiben, T. Baeck, M. Schoenauer, and H.-P. Schwefel, Eds, Berlin, Germany: Springer-Verlag, 1998, pp. 937–946. Lecture Notes in Computer Science 1498.

[21] ——, "Integrated facilities layout using a contour distance measure," *IIE Trans.*, vol. 33, no. 4, pp. 337–344, 2001.

[22] D. Sinriech, "Network design models for discrete material flow systems: A literature review," *Int. J. Adv. Manuf. Technol.*, vol. 10, pp. 277–291, MONTH? 1995.

[23] A. E. Smith and D. M. Tate, "Genetic optimization using a penalty function," in *Proc. 5th Int. Conf. Genetic Algorithms*, July 1993, pp. 499–505.

[24] K. Y. Tam, "Genetic algorithms, function optimization, and facility layout design," *Eur. J. Oper. Res.*, vol. 63, no. 2, pp. 322–346, Dec. 1992.

[25] ——, "A simulated annealing algorithm for allocating space to manufacturing cells," *Int. J. Production Res.*, vol. 30, no. 1, pp. 63–87, Jan. 1992.

[26] D. M. Tate and A. E. Smith, "Unequal-area facility layout by genetic search," *IIE Trans.*, vol. 27, pp. 465–472, 1995.

[27] J. A. Tompkins, "Facilities planning: A vision for the 21st century," *IIE Solutions*, vol. 29, no. 8, pp. 18–19, Aug. 1997.

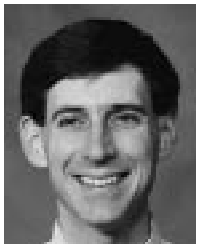
[28] D. J. van Camp, M. W. Carter, and A. Vannelli, "A nonlinear optimization approach for solving facility layout problems," *Eur. J. Oper. Res.*, vol. 57, pp. 174–189, 1991.

[29] I.-C. Yeh, "Construction-site layout using annealed neural network," *J. Comput. Civil Eng.*, vol. 9, no. 3, pp. 201–208, July 1995.

Rifat Aykut Arapoglu received the B.S. and M.S. degrees in industrial engineering from Middle East Technical University, Ankara, Turkey, in 1990 and 1993 and the M.S. degree in management science from Case Western Reserve University, Cleveland, OH, in 1996. He is currently working toward the Ph.D. degree in industrial engineering at the University of Pittsburgh, Pittsburgh, PA.

His current research interests include operations research.

Mr. Arapoglu is a member of INFORMS and the OR society of Turkey (YAD).



Bryan A. Norman received the Ph.D. degree in industrial and operations engineering from the University of Michigan, Ann Arbor, in 1995 and the B.S.I.E. and M.S.I.E. degrees from the University of Oklahoma, Norman, in 1988 and 1991, respectively.

He is currently an Assistant Professor of Industrial Engineering at the University of Pittsburgh, Pittsbrugh, PA. His research interests primarily focus on the modeling of complex problems in manufacturing systems. His areas of application include scheduling, sequencing, job rotation, assembly line balancing, and facility layout and material handling system design. He has published his research in *IIE Transactions*, *Naval Research Logistics*, *Engineering Design and Automation*, and *Computers and Industrial Engineering*.

Dr. Norman is a member of the IIE and INFORMS. He was National Science Foundation Fellowship holder while at the University of Michigan.

Alice E. Smith (M'99–SM'96) received degrees in engineering and business from Rice University, Houston, TX, Saint Louis University, Saint Louis, MO, and the University of Missouri at Rolla.

She is currently Professor and Chair of Industrial and Systems at Auburn University, Auburn, AL. She previously was on the Faculty of the Department of Industrial Engineering at the University of Pittsburgh, Pittsburgh, PA, which she joined in 1991 after ten years of industrial experience with Southwestern Bell Corporation. She has authored or coauthored over 100 publications in books, refereed proceedings, and journals, including articles in *IIE Transactions, INFORMS Journal on Computing, International Journal of Production Research, The Engineering Economist*, IEEE TRANSACTIONS ON RELIABILITY, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, and IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.

Dr. Smith is a Senior Member of IIE and SWE, a Member of Tau Beta Pi, INFORMS and ASEE, and a Registered Professional Engineer in Industrial Engineering in Alabama and Pennsylvania. She received the National Science Foundation CAREER Award in 1995. She is an Editor of *INFORMS Journal on Computing*, *IIE Transactions*, and IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.