A Genetic Algorithm Approach to Input/Output Station Location in Facilities Design

Rifat Aykut Arapoglu, Bryan A. Norman and Alice E. Smith
Department of Industrial Engineering
University of Pittsburgh
Pittsburgh, Pennsylvania 15261 USA
banorman@engrng.pitt.edu or aesmith@engrng.pitt.edu

Abstract: This paper presents a genetic algorithm approach to the location of a single 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. A genetic algorithm (GA) is shown to be an effective and efficient optimization method when compared to integer programming and to a greedy constructive heuristic on several test problems of varying size.

1. Introduction to the Problem

Facility design problems are a family of design problems involving 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. Such problems occur in many organizations, including manufacturing cell design, hospital design, and service center design. The problem primarily studied in the literature has been "block layout" which only specifies the placement of the departments, without regard for aisle structure and material handling system, machine placement within departments or input/output (I/O) locations. Block layout is usually a precursor to these subsequent design steps, termed "detailed layout." Two recent survey articles on the facility design problem are by Kusiak and Heragu [8] and Meller and Gau [9].

Because of the computational complexities in optimizing multiple and non-linear objectives and constraints, only limited work has been done beyond block layout. The recent work of Benson and Foote [5] in particular, considers the placement of aisles and I/O points *after* the relative location of the departments and the general aisle structure have been selected. Related work on integrated facility layout that considers machine placement includes papers by Nagi and others [6, 7]. This work uses predefined departmental shapes set on a grid covering the facility space. In [6], Dijkstra's shortest path algorithm is used to calculate the rectilinear distance to and from pre-specified I/O points. In [7], 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 heuristic to alter departmental placement. Another related work is by Banerjee et al. [3] where a genetic algorithm finds a "rough" layout that is then fully defined using a subordinate mathematical programming routine. The number of I/O's per department is pre-specified and then they are optimally located 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 of the block layout.

2. Problem Formulation

In this paper, block layout is approached by allowing departments to take rectangular shapes within a rectangular building (see Figure 1). The building is divided by first making vertical cuts, then making horizontal cuts within each vertical cut, termed a "bay." The bay is flexible because its width may vary according to the block layout chosen. Departments have specified area but not specified dimensions and are generally constrained by a minimum side length or a maximum aspect ratio. Layouts using flexible bays have been used before in the literature [12, 13]. The flexible bay construct is not necessary for the coming development but it simplifies the formation of a related network a great deal.

In this paper, a new distance metric, perimeter distance metric, is being used [10, 11]. 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 the moving pieces in a facility as they assume centroid to centroid rectilinear / Euclidean distances. The perimeter distance metric, on the other hand, uses the boundaries of the departments and follows the shortest path connecting a given pair of departments, as shown in Figure 1.

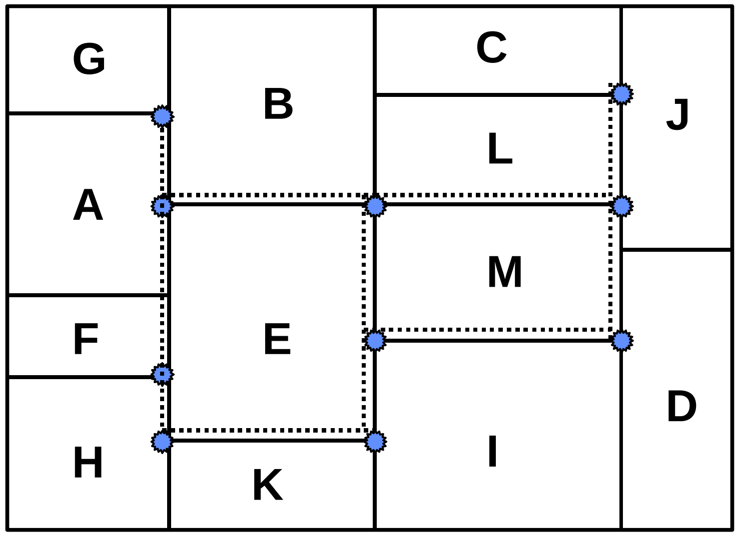

Figure 1. Flexible bay layout with 10 I/Os and material travel along department perimeters (dashed lines).

The single input/output (I/O) point placement problem, i.e. one I/O per department, in facilities design can be described as the minimization problem given on the following page which uses the notation below:

Notation:

$d\_cost_{i,j}$ = The distance from department *i* to department *j* using the I/O that is assigned for each department.

$X_{i,k}$ = binary variable that equals 1 if department *i* uses I/O candidate location *k*, zero otherwise

$X_{i,k,j,l}$ = binary variable that represents which I/Os give the shortest path from department *i* to department *j*. It equals 1 if department *i* uses I/O candidate location *k* and department *j* uses I/O candidate location *l*, zero otherwise

$f_{i,j}$ = the flow (material times costs per unit distance) between departments *i* and *j*

$d_{i,j,k,l}$ = the distance from candidate I/O location *k* in department *i* to candidate I/O location *l* in department *j*.

Loc_i= the set of candidate I/O locations for department *i*.

N the number of departments in the layout.

$$\text{Min} \sum_{i=1}^{N}\sum_{j=1}^{N} f_{ij} d\_cost_{ij} \qquad (1)$$

Subject to:

An assignment constraint that requires each department to have exactly 1 I/O location:

$$\sum_{k \in Loc_i} x_{ik} = 1 \quad \forall\, i = 1,\ldots N$$

Cost calculation constraints:

$$x_{i,k,j,l} \leq x_{i,k} \qquad \forall\, i=1,\ldots N,\ j=1,\ldots N,\ k \in Loc_i,\ l \in Loc_j$$

$$\sum_k \sum_l x_{i,k,j,l} = 1 \quad \forall\, i = 1,\ldots N,\ j = 1,\ldots N$$

$$\sum_k \sum_l x_{i,k,j,l} d_{i,k,j,l} = d\_cost_{i,j} \quad \forall\, i = 1,\ldots N,\ j = 1,\ldots N$$

The second constraint permits $x_{i,k,j,l}$ to equal one only if both $x_{i,k}$ and $x_{j,l}$ equal one. The third constraint forces one of the x_{ik} and x_{jl} pairs (when both are equal to one due to the first two constraints) to be chosen as the shortest path from department *i* to department *j*. Constraint four insures that the shortest path will be chosen.

3. Complexity of the Single I/O Placement Problem

The following two observations from [11] make possible the design and analysis of the single I/O point location problem:

- The total number of I/O points is finite and restricted to the set of departmental intersection points.
- There can be at most 2*N*-2 such points for a flexible bay layout.

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. Namely,

$$\prod_{i=1}^{N} \text{Loc}_i$$

This number is on the order of 10^{13} for a layout with 20 departments.

4. Optimization Methodologies

It is assumed that a flexible bay layout has already been identified by a genetic algorithm or other search technique and the optimization task at hand is to locate a single I/O point for each department in the layout such that material handling costs are minimized using the perimeter distance metric. The shortest paths between all pairs of candidate I/O points, which are located at the intersection points of departments (see Figure 2), need to be found. This is accomplished by creating a network with non-negative arc lengths that represent the distances between the I/O points using the perimeter distance metric. The nodes of the network are the I/O points found at the intersection points 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.

To locate the optimum I/O point for each department, three approaches were developed and compared. An exact approach to solve this problem is through integer programming (IP). This always finds the optimum solution, but the search space grows exponentially with the number of candidate I/O locations. A computationally expedient approach is a constructive heuristic that works in a greedy fashion. The third alternative used genetic algorithms (GA) to improve upon a set of feasible, but randomly chosen, I/O points. These three approaches are now discussed.

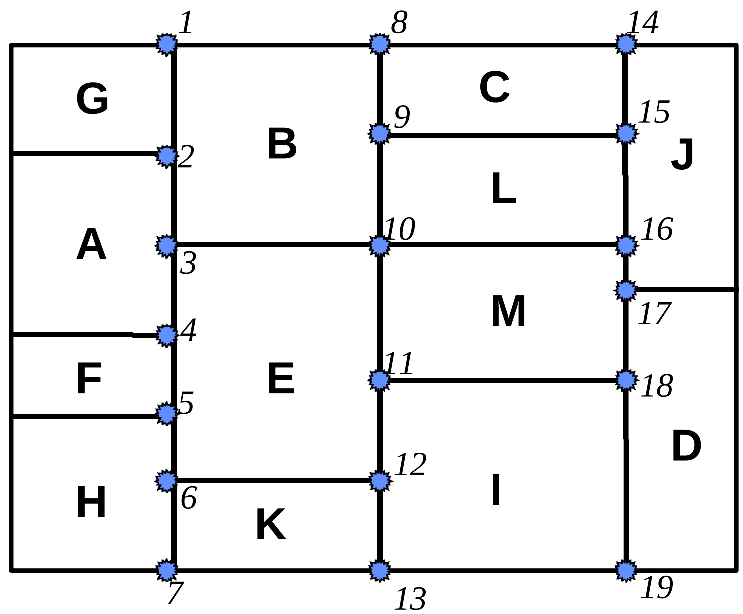

Figure 2. Candidate I/O locations for a flexible bay layout.

4.1 The Integer Programming Approach

The formulation developed in section 2 was coded as an IP and solved using CPLEX.

4.2 The Constructive Heuristic

Another approach is a very quick constructive heuristic that operates in a greedy fashion. This heuristic is deterministic and assigns the I/O points in order of the magnitude of the flows. The pseudocode is below.

Additional Notation:

A	set of assigned departments.
$S = \{1, 2, 3, \ldots, N\}$	set of all departments.
$totf_{i,j}$	total flow that uses I/O point j of department i.

Algorithm:

Step 1:

$S = \{1, 2 \ldots N\}$

$A = \varnothing$

Set $totf_{i,j} = 0$ for all $i \in S$ and $j \in$ Loc$_i$

Step 2:

If $A = S$ then stop: all departments have been assigned an I/O point.

Else

Find the shortest paths between every pair of I/O points by using the Floyd-Warshall algorithm. For each flow, $f_{i,k}$ that uses I/O point j of department i and I/O point l of department k, update $totf_{i,j} = totf_{i,j} + f_{i,k}$ and $totf_{k,l} = totf_{k,l} + f_{i,k}$

Step 3 :

Let $totf_{r,s}$ = Max $\{totf_{i,j} : i \in S\text{-}A, j \in \text{Loc}_i\}$

$A \leftarrow A \cup \{r\}$. Assign I/O point s to department r.

Go to step 2.

4.3. Genetic Algorithm

Encoding and Fitness

A permutation encoding is used to represent a given allocation of I/O points to the departments. Each candidate I/O point is numbered (to a maximum of 2N-2) and it is possible that two neighboring departments may use the same *location* as their I/O point. In that case the same I/O number is used to represent both I/O points. The following example denotes the encoding corresponding to the layout given in Figure 1 using the I/O numbering of Figure 2.

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

The fitness of a given solution is the cost of the material flow (i.e., equation 1) between each pair of departments using the selected I/O points and the perimeter distance metric.

Selection, Crossover, Mutation

One parent is selected using a tournament of size 2 while the other parent is selected randomly from the population. Uniform crossover is applied to produce one offspring, with a preference of 0.70 for alleles from the more fit parent. Note that every crossover produces a feasible allocation of I/O points and thus, there is no repair needed. Offspring replace parents except for the best solution in the population.

The new population, except for the best member, is subject to mutation. Every allele is tested for mutation individually and independently with a preselected mutation probability. If an allele is chosen for mutation, the new allele is selected randomly from the feasible I/O points for that department, not including the current value. Again, every mutation results in a feasible allocation of I/O points.

Termination Criterion

The GA is terminated after 100 consecutive generations without any improvement in the best objective function value.

5. Test Problems and Results

The three approaches were applied to three instances of well known test problems. A 14 department problem by Bazaraa [4] and two instances of the well known 20 department test problem of Armour and Buffa [2] were used. The objective function results of the IP and the constructive heuristic along with a summary of 5 runs of the GA are compared in Table 1.

Table 1. Computational Test Results

Problem	Constructive Heuristic	GA			Mathematical Program
		Min	Average	Max	
Bazaraa 14 dept.	5459.2	4991.8	5109.4	5187.8	4991.8
Armour & Buffa 20 dept. 1	388.6	344.8	348.2	350.8	†
Armour & Buffa 20 dept. 2	371.2	334.9	334.9	334.9	†

† These problems could not be solved to optimality by CPLEX after 60 hours on a SunSparc20

Both the Constructive Heuristic and the GA required less than one second of CPU time to solve all three problems. The IP solved the 14 department problem in 40 minutes but could not solve the two larger problems in 60 hours on a SunSparc 20.

6. Conclusions

The GA was shown to be an effective heuristic method for the single I/O placement problem. Because of the great computational effort of the integer programming method, it is only practical for very small problems. The constructive heuristic performs well for this problem and is very fast. The GA requires more computational effort than the constructive heuristic but is able to improve upon its performance in every case with improvements ranging from 5.0 % to 11.2%. For the one problem where the optimal solution could be determined using the IP the GA found solutions that were at most 4% above the optimal value but on average were only 2.4% above the optimal value. For a complete optimization methodology, it may make sense to use the constructive heuristic for most of the GA search of the layout total 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.

References

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

[2] Armour, G. C., Buffa, E. S.: A Heuristic Algorithm and Simulation Approach to Relative Allocation of Facilities. *Management Science* (9) (1963) 294-309

[3] Banerjee, P., Zhou, Y., Montreuil, B.: Genetically Assisted Optimization of Cell Layout and Material Flow Path Skeleton. *IIE Transactions* (29) (1997) 277-291

[4] Bazaraa, M. S.: Computerized Layout Design: A Branch and Bound Approach. *AIIE Transactions* (7) (1975) 432-438

[5] Benson, B., Foote, B. L.: Door FAST: A Constructive Procedure to Optimally Layout a Facility Including Aisles and Door Locations Based on an Aisle Flow Distance Metric. *International Journal of Production Research* (35) (1997) 1825-1842.

[6] Harhalakis, G., Lu, T., Minis, I., Nagi, R.: A Practical Method for Design of Hybrid-type Production Facilities. *International Journal of Production Research* (34) (1996) 897-918

[7] Kane, M. C., Nagi, R.: Integrated Material Flow and Layout in Facilities Design. In *Proceedings of the Sixth Industrial Engineering Research Conference*, Miami FL (May 1997) 919-924

[8] Kusiak, A., Heragu, S. S.: The Facility Layout Problem. *European Journal of Operational Research* (29) (1987) 229-251

[9] Meller, R. D., Gau, K.-Y.: The Facility Layout Problem: Recent and Emerging Trends and Perspectives. *Journal of Manufacturing Systems* (15) (1996) 351-366

[10] Norman, B. A., Smith, A. E., Arapoglu, R. A.: Integrated Facilities Layout Using a Perimeter Distance Metric. *Parallel Problem Solving From Nature*, Springer-Verlag, (1998) in print.

[11] Norman, B. A., Smith, A. E., Arapoglu, R. A.: Integrated Facilities Layout Using a Perimeter Distance Measure. *IIE Transactions*, in review

[12] Tate, D. M., Smith, A. E.: Genetic Algorithm Optimization Applied to Variations of the Unequal Area Facilities Layout Problem. In *Proceedings of the Second Industrial Engineering Research Conference*, Los Angeles (May 1993). IIE, Atlanta (1993) 335-339

[13] Tate, D. M., Smith, A. E.: Unequal-area Facility Layout by Genetic Search. *IIE Transactions* (27) (1995) 465-472