44963

National Library of Canada | Bibliothèque nationale du Canada

CANADIAN THESES ON MICROFICHE

THÈSES CANADIENNES SUR MICROFICHE

NAME OF AUTHOR/*NOM DE L'AUTEUR*: Michael Scriabin

TITLE OF THESIS/*TITRE DE LA THÈSE*: Facility Layout by Analysis of Clusters

UNIVERSITY/*UNIVERSITÉ*: Simon Fraser University

DEGREE FOR WHICH THESIS WAS PRESENTED/*GRADE POUR LEQUEL CETTE THÈSE FUT PRÉSENTÉE*: Ph.D.

YEAR THIS DEGREE CONFERRED/*ANNÉE D'OBTENTION DE CE GRADE*: 1980

NAME OF SUPERVISOR/*NOM DU DIRECTEUR DE THÈSE*: Roger C. Vergin, Professor

DATED/*DATÉ*: April 2/80 SIGNED/*SIGNÉ*:

PERMANENT ADDRESS/*RÉSIDENCE FIXE*: [redacted]

NL-91 (3-74)

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us a poor photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

THIS DISSERTATION HAS BEEN MICROFILMED EXACTLY AS RECEIVED

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de mauvaise qualité.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

LA THÈSE A ÉTÉ MICROFILMÉE TELLE QUE NOUS L'AVONS REÇUE

F L A C

Facility Layout by Analysis of Clusters

by

Michael Scriabin

M.B.A., Simon Fraser University, 1974

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
in the Department
of
Economics

SIMON FRASER UNIVERSITY
January 1980

Title of Thesis/Dissertation:

Facility Layout by Analysis of Clusters

Author:

(signature)

Michael Scriabin

(name)

April 2/80

(date)

APPROVAL

Name: Michael Scriabin

Degree: Doctor of Philosophy

Title of Thesis: Facility Layout by Analysis of Clusters

Examining Committee:

Chairperson: Stephen T. Easton

Roger C. Vergin
Senior Supervisor

Pao Cheng

Gary Mauser

Robert Meier
External Examiner
Professor
College of Business and Economics
Western Washington University

Date Approved: April 2/80

ABSTRACT

The facility layout problem has been defined as the assignment of n facilities to n locations so as to minimize the cost of interfacility flows, and has been formulated as a quadratic assignment problem. Due to the combinatorial nature of the problem, and in spite of the computational capacity of today's computers, optimization remains impractical for problems involving more than about eight facilities. An examination of previous research efforts in facility layout and a discussion of the complexity of facility layout problems indicates areas of weakness of current heuristic computer approaches, and leads to the development of the computer algorithm FLAC (Facility Layout by Analysis of Clusters).

Incorporating certain features hitherto used only in visual or interactive methods, FLAC solves the problem in three stages. Stage 1 makes use of cluster analysis to develop an unconstrained configuration of the facilities similar to the one-line schematic diagram of flows used in visual methods. A unique constant is added to the flow-cost matrix prior to the application of multidimensional scaling methods to provide appropriate separation of the facilities in both dimensions. Stage 2 fits the facilities into the available layout space. By using as a criterion the minimization of

weighted distances facilities are moved from their positions at the end of stage 1, the original quadratic assignment problem is reduced to a simple assignment problem and solved using an efficient primal-dual algorithm. Final adjustment of individual facilities is made in stage 3, using an exchange algorithm similar to those used by previous researchers.

The fully automated FLAC algorithm is shown to perform consistently better than CRAFT (Computerized Relative Allocation of Facilities Technique), which is chosen as the standard of comparison in terms of solution quality, especially in large problems and in certain types of problems in which computer algorithms have previously been shown to be inferior to visual methods. In addition to providing high quality solutions under widely varying conditions, FLAC is shown to be superior in terms of computation time since the only algorithms with competitive running times on large problems produce inferior results. The rate of increase of computation time with problem size is also attractive, with computation time increasing at a rate proportional to less than the cube of problem size.

The design of FLAC allows the easy inclusion of such features as the handling of unequal areas and other practical considerations, and it should ideally be part of a larger interactive system incorporating man's visual ability and judgment as well as the computer's consistency.

ACKNOWLEDGMENTS

Many persons have directly or indirectly supported the research and development described in this dissertation. First of all I would like to thank Professor Roger C. Vergin for his expert guidance and for his faith in the eventual useful outcome of this research. Thanks also to Professor Pao Lun Cheng whose dedication to excellence in all his undertakings inspires ever greater efforts. To Professor Gary A. Mauser and fellow student Lindsay Meredith, special thanks for opening new horizons by introducing me to the field of multi-dimensional scaling. The helpful suggestions and constructive criticism of the faculty and students of the Economics and Commerce Department at Simon Fraser University who attended my workshop presentations contributed significantly to the development and improvement of the FLAC algorithm. I would also like to record my appreciation for the extra effort and patience with which Terry Ross tackled a lengthy and difficult typing job. Finally, I wish to acknowledge the moral and financial support of the Canada Council, in the form of a doctoral fellowship, which made this research possible.

This dissertation is dedicated to my wife Jannie whose unshakable optimism, continuous unselfish support, and understanding kept me going through some difficult times.

CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION AND RESEARCH OBJECTIVES

Background

Interest in a systematic approach to the facility layout problem dates back to the industrial revolution. Prior to that, the fixed-position layout, in which the product remains stationary and the tools are brought to the product, was common.[1] Increasing use of machinery and equipment led to greater popularity of the product type layout (assembly line) for repetitive processes and of the process type layout for intermittent, or job-lot, processes in which several different products follow different paths through the machine centres or facilities, allowing the products to share the facilities. Thus emphasis shifted to the relative location of facilities or machines to minimize materials handling cost.

Because of the difficulty of minimizing materials handling cost by traditional manual and visual methods, much operations research effort in the past two decades has been aimed at finding a good computer approach to solving the problem. There are many variables affecting materials handling cost, but it is generally accepted in industrial engineering that the relationship between materials handling cost and the distance

---

[1] J.M. Moore, Plant Layout and Design, New York, Macmillan Co., 1962, p. 106.

a product is moved is linear.[1]

Francis and White have classified the facility layout problem as a special type of facility location problem,[2] and most researchers developing or evaluating computer algorithms which assist in the overall design of a layout formulate the problem simply as the assignment of n facilities to n locations so as to minimize the function

$$\min_{\{k,l \in L\}} c = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} f_{ij} d_{kl}$$

where:

f_{ij}, $i<j$ = the total cost per unit distance of flow in both directions between the ith facility and the jth facility.

d_{kl} = the distance between the kth location and the lth location ($d_{kl}=0$ for k=l).

L = set of n locations to which facilities may be assigned.[3]

In practice the f_{ij} are based on a matrix V of interfacility

---

1 The reader interested in the practical aspects of facility layout, particularly the methods of data gathering, choice of materials handling methods, etc., is referred to the excellent text by R. Muther, Systematic Layout Planning, Boston, Industrial Education Institute, 1961.

2 Richard L. Francis and John A. White, Facility Layout and Location: An Analytical Approach, Englewood Cliffs, N.J., Prentice-Hall, 1974, p. 3.

3 Christopher E. Nugent, Thomas E. Vollmann, and John Ruml, "An Experimental Comparison of Techniques for the Assignment of Facilities to Locations," Operations Research, Vol. 16, No. 1, Jan. 1968, p. 151.

flow volumes (sometimes called a travel chart) and a matrix U of interfacility costs per unit volume per unit distance. These n x n matrices are usually asymmetric. Typically the flow volume from facility i to facility j does not equal that in the opposite direction, and costs may also differ. However once each element of U is multiplied by the corresponding element of V, the resulting matrix is usually added to its transpose, forming the symmetric matrix F in which the n(n-1)/2 elements f_{ij} (i<j) lie above the diagonal.[1]

In most computer approaches the problem is at least initially further simplified by considering all facilities to require an equal area. A good solution to the problem with assumed equal areas can then be used as a guide in designing the final practical layout, taking into account unequal areas using methods such as those suggested by Ritzman.[2]

Despite much research effort in the past two decades, no method has been developed which can provide an optimum solution to non-trivial layout problems, and the effectiveness of available methods for even this narrowly defined problem varies from layout to layout.

---

1 T.E. Vollmann, "An Investigation of the Bases for the Relative Location of Facilities," Unpublished doctoral dissertation, University of California, Los Angeles, 1964, pp. 25-26.

2 L.P. Ritzman, "The Efficiency of Computer Algorithms for Plant Layout," Unpublished doctoral dissertation, Michigan State University, 1968, p. 22, suggests several methods for dealing with unequal areas which need not be included in the computer algorithm.

Optimizing Algorithms

During the sixties, a number of optimum-seeking procedures were designed or suggested. Initial attempts at solution by enumeration led to formulation as a quadratic assignment problem by Gilmore and Lawler, and as a travelling salesman problem by Gavett and Plyter, and solved by branch-and-bound.[1] Koopmans and Beckmann showed that a linear programming model equivalent to the quadratic programming model for an n-facility problem would require n^4+n^2 variables and n^3+2n constraints, and without the additional limitation that n^2 variables must be 0 or 1, would result in the trivial solution that an equal fraction of each facility should be assigned to each location.[2] The quadratic assignment problem formulation of the layout problem has been shown to be "not computationally feasible for larger problems."[3]

The Gavett and Plyter travelling salesman formulation

---

[1] P.C. Gilmore, "Optimal and Sub-Optimal Algorithms for the Quadratic Assignment Problem," Journal of the Society for Industrial and Applied Mathematics, Vol. X, 1962, pp. 305-313; E.L. Lawler, "The Quadratic Assignment Problem," Management Science, Vol. 9, 1963, pp. 586-599; J.W. Gavett and N.V. Plyter, "The Optimal Assignment of Facilities to Locations by Branch and Bound," Operations Research, Vol. 14, 1966, pp. 210-232.

[2] Tjalling C. Koopmans and Martin Beckmann, "Assignment Problems and the Location of Economic Activities," Econometrica, Vol. 25, Jan. 1957, pp. 67-68.

[3] Nugent et al., "Experimental Comparison," p. 152.

transforms the n-facility layout problem into an N-city travelling salesman problem (where $N=n(n-1)/2$), which is solved by branch-and-bound. As of 1969, no one had had much success with the optimization of travelling salesman problems in excess of about 40 cities.[1] Gavett and Plyter themselves stated that the largest problem conveniently handled by branch-and-bound on the IBM 7074 was an 8 facility problem (equivalent to a 28 city travelling salesman problem.)[2]

Heuristic Algorithms

Heuristic methods for solving the facility layout problem do not guarantee an optimum solution, but rather attempt to provide a high probability of obtaining an acceptable solution with a reasonable amount of effort.

Heuristic methods for the facility layout problem have been classified into two categories:

(1) Construction procedures, in which the algorithm begins with an empty layout, assigning one facility at a time until the layout is complete, and

(2) improvement procedures, in which an initial layout, often chosen randomly, is iteratively improved upon.

---

1 M.L. Balinski and K. Spielberg, "Methods for Integer Programming," Progress in Operations Research, Vol. III, ed. Julius S. Aronofsky, New York, John Wiley & Sons, 1969, p. 213.

2 Gavett and Plyter, "Optimal Assignment," p. 228.

Although these heuristic algorithms have been described elsewhere,[1] a short description and discussion are included of the better known algorithms which will be referred to in this dissertation.

Construction procedures

CORELAP.--(COmputerized RElationship LAyout Planning)[2] chooses the facility with the highest total flow(closeness rating) between it and other facilities, and places it in the layout. It then looks for the facility having the greatest interaction with the first one, and places it in the layout adjacently. When no more facilities are available which have large interactions with those facilities already placed, the next facility is selected which has the highest total flow, and so on until all facilities are placed. In placing each facility in the layout, CORELAP attempts to maximize the length of border it has in common with facilities with which it has high interaction (the shape of each facility is however limited to be rectangular). A recent version of CORELAP is available which is interactive.[3]

---

[1] For a summary of the earlier methods, see for instance T.E. El-Rayah and R.H. Hollier, "A Review of Plant Design Techniques," The International Journal of Production Research, Vol. 8, No. 3, 1970, pp.263-279.

[2] R.C. Lee and J.M. Moore, "CORELAP--Computerized Relationship Layout Planning," Journal of Industrial Engineering, Vol. 18, No. 3, 1967, pp.195-200.

[3] Francis and White, Facility Layout, p. 114.

ALDEP.--(Automated Layout DEsign Program)[1] is similar to CORELAP, but when no facilities are available which have high interaction with previously placed facilities, the next facility is selected randomly.

ALDEP and CORELAP have been referred to as "qualitative" rather than quantitative because they design layouts on the basis of activity relationships rather than flows.[2] However, once a relationship chart has been quantified, these algorithms can be compared with the quantitative algorithms which follow.

HC-66.--(Hillier and Connors 1966 construction procedure)[3] calculates at each iteration a lower bound to the cost of assigning each unassigned facility to each unoccupied location. That facility is assigned which has the greatest opportunity loss associated with its assignment to another location.

MAT.--(Modular Allocation Technique)[4] makes use of the theorem: "The sum of pairwise products of two sequences of real numbers is minimized if one sequence is arranged in non-decreasing order and the other is arranged in non-increasing order." MAT

---

[1] J.M. Seehof and W.O. Evans, "Automated Layout Design Program," The Journal of Industrial Engineering, Vol. 18, No. 12, 1967, pp.690-695.

[2] Francis and White, Facility Layout, p. 140.

[3] F.S. Hillier and M.M. Connors, "Quadratic Assignment Problem Algorithms and the Location of Indivisible Facilities," Management Science, Vol. 13, No. 1, Sept. 1966, pp. 47-49.

[4] H.K. Edwards, B.E. Gillett, and M.E. Hale, "Modular Allocation Technique (MAT)," Management Science. Vol. 17, Nov. 1970, pp.161-169.

therefore attempts to assign the largest flow cost to the smallest distance in the layout, and so on, assigning one or two facilities to locations at each iteration, according to whether or not one of the pair involved in the flow has been assigned.

Improvement procedures

CRAFT.--(Computerized Relative Allocation of Facilities Technique)[1] is probably the best known heuristic procedure for the facility layout problem. It begins with an initial, often random, layout and evaluates the effect on cost of all possible pairwise exchanges of facilities. An exchange is performed of that pair of facilities which results in the highest cost saving, and the process is repeated until no pairwise exchange results in a cost improvement.[2] CRAFT can also handle unequal facility areas. However, when facility areas are not equal, CRAFT considers only exchanges of facilities with common boundaries. Thus CRAFT is most powerful in the situation where facility areas are equal.

H-63.--(Hillier's 1963 procedure)[3] incorporates the calculation of a move desirability table (MDT) at each iteration. The MDT

[1] G.C. Armour and E.S. Buffa, "A Heuristic Algorithm and Simulation Approach to Relative Location of Facilities," Management Science, Vol. 9, No. 2, Jan. 1963, pp.294-309.

[2] The effects of three-way exchanges are also evaluated and discussed in chapter IV of this dissertation.

[3] F.S. Hillier, "Quantitative Tools for Plant Layout Analysis," The Journal of Industrial Engineering, Vol. 14, No. 1, Jan.-Feb. 1963, pp.33-40.

contains for each facility the cost saving which would be achieved by moving that facility unilaterally in vertical or horizontal directions. H-63 chooses the facility with the highest entry in the MDT and evaluates an exchange with the adjacent facility in the direction indicated. The exchange is executed if it is profitable, otherwise the next highest MDT rating is used, and so on. When an exchange has been made, a new MDT is calculated. The algorithm terminates when no more profitable exchanges can be found.

HC63-66.--(Hilliers and Connors 1966 improvement procedure)[1] is similar to H-63 but calculates an N-step move desirability table, allowing for non-adjacent moves. N is initially set to max. $\{$no. rows - 1, no. columns - 1$\}$. Diagonal moves are considered as well as horizontal and vertical ones, but they must be in a straight line.

Biased Sampling.--Biased sampling[2] is a method devised by Nugent et al. to produce several different layouts from the same initial layout, whether chosen randomly or not. The method differs from CRAFT in that the pairwise exchange selected at each iteration is not simply that one which achieves the greatest cost reduction, but is chosen randomly from all exchanges which produce cost reductions. The choice is biased

1 Hillier and Connors, "Quadratic Assignment Problem Algorithms," pp.49-50.

2 Nugent et al., "Experimental Comparison," pp.156-157.

in favour of greater cost reductions. The degree of bias is controlled by the choice of the parameter in the calculation of the selection probability

$$P_j = (S_j)^C / \sum_{i=1}^{k} (S_i)^C$$

where:

P_j = the probability of selecting the jth pairwise exchange with positive cost reduction,

S_j = the amount of the jth pairwise exchange's cost reduction,

k = the number of pairwise exchanges with positive cost reductions, and

C = a parameter to vary the effect of differing cost reductions.

The choice of a very high value for C would lead to identical repetitions of a CRAFT run. On the other hand, setting C=0 would result in an unbiased choice of exchanges at each iteration, providing a potentially wider range of solutions. Nugent et al. chose 2.0 as the value of C in their experiments.

COL.--(Computerized model for Office Layout)[1] uses a modification of the basic CRAFT two-way exchange algorithm, designed to reduce computational effort. It achieves this reduction by limiting the number of potential exchanges evaluated at each iteration. Only interactions between facilities separated by more than α distance units are considered. The constant α is

---

[1] T.E. Vollmann, C.E. Nugent, and R.L. Zartler, "A Computerized Model for Office Layout," The Journal of Industrial Engineering, Vol. 19, July 1968, pp.321-327.

specified by the user. COL selects the two facilities having the greatest interaction with other facilities more than α distance units away, then, taking first the facility with greatest interaction, evaluates all possible exchanges with other facilities. The most profitable exchange is executed and when no further exchanges involving that facility are profitable, the second facility is processed in the same way. When no further exchanges involving the second facility are profitable, two more facilities are chosen, and so on. This portion of the algorithm terminates when two facilities are chosen neither of which can be profitably exchanged with any other. The algorithm finally performs two iterations of the CRAFT algorithm and stops.

FRAT.--(Facilities Relative Allocation Technique)[1] is similar to COL except that the distance parameter α is varied within the program, being set initially to

$$\alpha = R-L$$

where R and L are, respectively, the maximum and minimum possible distances between facilities. When no further improvements can be made, R is decreased by L and iteration resumes, and so on until α is less than L.

---

[1] T.M. Khalil, "Facilities Relative Allocation Technique (FRAT)," International Journal of Production Research, Vol. II, No. 2, 1973, pp.183-194.

TSP--(Terminal Sampling Procedure)[1] is also very similar to COL, evaluating only exchanges involving the two facilities with the highest flow-distance product vectors. TSP also keeps track of tie solutions, and returns to perform improvements on those tied layouts. In any one run several ties can occur, with the result that several different solutions can be reached. Each solution is further improved upon, as in FRAT, by executing the basic CRAFT algorithm. TSP chooses the best of these final solutions.

Comparisons of Current Computer Algorithms

Using improvement procedures which consider only pairwise or three-way exchanges can lead to a sub-optimal final layout because profitable multiple exchanges may be missed.[2]

Similarly, building layouts using a construction procedure without rearrangement of assigned facilities as each new facility is added may lead to a sub-optimal final layout even if an intermediate layout is optimal, since each facility except the last is placed without complete information as to the locations of the rest.

---

[1] G.G. Hitchings and M. Cottam, "An Efficient Heuristic Procedure for Solving the Layout Design Problem," OMEGA, The International Journal of Management Science, Vol. 4, No. 2, 1976, pp.205-214.

[2] Any multiple exchange can be viewed as a sequence of two or more pairwise exchanges. However, even though the completion of the sequence would result in a cost reduction, an intermediate exchange in the sequence could increase cost, bringing the algorithm to a halt.

Studies have shown that the improvement procedures, especially CRAFT, are superior to the construction procedures, although combinations such as MAT followed by CRAFT provide results competitive with CRAFT.

Denholm and Brooks found CRAFT superior to CORELAP and ALDEP. They also found that CORELAP, which does not have a restriction on plant shape, generated an irregular configuration which was not a practical solution to their problem since they were considering rearranging their plant.[1]

HC66(Hillier's construction procedure) was found by Hillier and Connors to be inferior to the improvement algorithm HC63-66[2] and in any case is inefficient for large problems as computational effort is stated by Hillier and Connors to be proportional to n^4 as is that of Gilmore's n^4 procedure(his n^5 procedure is even more sensitive to problem size).[3]

Although Nugent et al. claim Biased Sampling gives better results (with additional computation effort) than CRAFT,[4] there is no theoretical reason why, on the average, these

---

[1] D.H. Denholm and G.H. Brooks, "A Comparison of Three Computer Assisted Plant Layout Techniques," Proceedings, American Institute of Industrial Engineers, 21st Annual Conference and Convention, Cleveland, 1970, pp.77-84.

[2] Hillier and Connors, "Quadratic Assignment Problem Algorithms," p. 55.

[3] P.C. Gilmore, "Optimal and Sub-optimal Algorithms for the Quadratic Assignment Problem," Journal of the Society for Industrial and Applied Mathematics, Vol. 10, No. 2, June 1962, pp. 305-313.

[4] Nugent et al., "Experimental Comparison," p. 162.

results should be better than those that would be achieved by running CRAFT from random starting layouts for an equal length of computer time. In fact it is possible that the best of say ten CRAFT runs from random initial solutions may be stochastically better than Biased Sampling solutions with sample size ten. The rationale behind this possibility is based on the assumption that the variance of the costs of any group of ten solutions evaluated in a Biased Sampling run is likely to be less than the variance of ten CRAFT solutions from random starts, especially if the parameter C is high.[1] At the same time, the means will tend to be the same, as found by Nugent _et al_. in their own experiments.[2] If this is indeed the case, then one should expect the best of groups of ten CRAFT solutions to be better than the Biased Sampling solutions with sample size ten. The superiority of the CRAFT solutions depends on the extent of the difference in variances. For instance, if the variance of the Biased Sampling layout costs from one random initial layout were zero (which would be the case given a sufficiently high value for the parameter C), then choosing the best layout would be equivalent to choosing one random CRAFT solution. This would clearly be inferior, on the average, to choosing the best of ten CRAFT solutions.

In order to test this hypothesis, two sets of ten Biased

---

1 See the description of Biased Sampling on page 9 .

2 Nugent _et al_., "Experimental Comparison," p. 163.

Sampling runs with sample size ten were performed on the twelve facility problem used by Nugent et al. in their experiments. In the first set of runs the parameter C was set equal to 2.0, the value chosen by Nugent et al. in their experiments. In the second set, C was set equal to zero, so that exchanges would be chosen randomly from those which effect positive cost reductions. The ten results from each set were compared with the best results of ten groups of ten CRAFT runs from random initial layouts for the same problem.

Table 1 shows the costs of the best layouts from each method ranked for convenient comparison in ascending order. The CRAFT costs are slightly better than both of the Biased Sampling costs. However, applying the Mann-Whitney U test to these results indicates the differences are not statistically significant. Thus it cannot be concluded on the basis of this experiment that the best layouts of groups of ten CRAFT runs are better than those of Biased Sampling runs with sample size ten, when $C \leq 2$. However, Biased Sampling appears to offer no advantage in solution quality over multiple CRAFT runs using the same sample size.

The computation time of Biased Sampling was reported by Nugent et al. to be increased over CRAFT due to the additional solutions generated. A closer look at his run times shows that in all except the twelve facility problem, the average computation time for Biased Sampling was more than ten times as long as the average CRAFT run time, so that running costs would

TABLE 1

LAYOUT COSTS OF BEST SOLUTIONS OF GROUPS OF TEN CRAFT RUNS AND OF BIASED SAMPLING RUNS WITH SAMPLE SIZE TEN

Best of ten CRAFT runs	Biased Sampling (sample size 10, C = 2)	Biased Sampling (sample size 10, C = 0)
289	293	291
293	293	293
293	295	293
293	295	293
293	295	295
295	295	295
295	295	296
296	295	297
296	298	299
297	299	300

be cheaper for multiple CRAFT runs. This is due not only to the additional complexity of Biased Sampling, but mainly to the increased number of iterations which must be performed per solution if the largest cost reduction is not chosen at each iteration. Since the number of iterations per solution in Biased Sampling runs would clearly equal that of CRAFT if the parameter C were set high enough, it should be expected that as C is reduced from infinity in order to increase solution quality versus a single CRAFT run, computation time will increase on the average beyond that of a multiple CRAFT run of the same sample size.

In the experimental set of runs with C=2, the average computation time was slightly (8%) greater for Biased Sampling than the average time required for ten CRAFT solutions. The number of iterations per solution in Biased Sampling was found to be stochastically greater than in CRAFT, at a significance level of .003 using the Mann-Whitney test. With C=0, the average computation time for Biased Sampling was 40% greater than that required for ten CRAFT runs, and the number of iterations was sufficiently great in Biased Sampling that the application of the Mann-Whitney test resulted in a U statistic more than nine standard deviations from the mean(U is approximately normally distributed). Thus there is always a penalty, either in solution quality or in computation time, associated with using Biased Sampling from random initial layouts rather than multiple CRAFT runs of the same sample size.

H-63, HC63-66, COL, FRAT, and TSP, as well as an Interchange algorithm used by Edwards et al. to improve on MAT,[1] are variations on the basic two-way exchange algorithm of CRAFT. All of these variations are basically aimed at saving computation time by limiting the search at each iteration for two-way exchanges which improve the solution. COL, FRAT, TSP, and the Interchange algorithm employ similar methods in reducing the number of potential exchanges considered at each iteration. FRAT concludes with the execution of the CRAFT algorithm, so that its results should be comparable to CRAFT. This speculation is supported by the application of the sign test to the results of FRAT and CRAFT on Nugent's problems,[2] showing no significant difference.

TSP has one feature which makes it potentially competitive with Biased Sampling or multiple CRAFT runs, in terms of solution quality. It keeps track of tie solutions, and returns to perform improvements on these tied layouts. In any one run several ties can occur, with the result that several different final solutions can be reached, from which TSP chooses the best. TSP is in effect a variation on the concept of Biased Sampling, but whereas Biased Sampling returns to the initial layout after each solution, TSP returns only to an intermediate solution at which a tie was detected, before iterating to the next solution.

---

1 Edwards et al., "Modular Allocation Technique," pp.165-166.

2 Khalil, "Facilities Relative Allocation Technique," p. 189.

However, while the user has control over the sample size in Biased Sampling, the number of solutions generated by TSP, and therefore the quality of the best solution, is dependent on the number of ties encountered. As with Biased Sampling, the additional solutions from which TSP chooses are produced at the expense of additional computation time, and there is no theoretical reason why the best of those solutions should be better in quality than the best of a similar number of CRAFT solutions, although they would be achieved with less computational effort.

Since COL concludes with the execution of only two iterations of the CRAFT algorithm, and its search is more limited than that of CRAFT, COL should not, on the average provide as good solutions. Khalil found COL to produce inferior results to FRAT.[1]

The Interchange algorithm tested by Edwards et al. provided inferior solutions to CRAFT even when using MAT input, although computation time was shorter. The combination of MAT followed by Interchange, followed by CRAFT produced results competitive with, but not better than, CRAFT alone, although computation time was shorter.[2]

Neither H-63 nor HC63-66 conclude with the complete search of all possible two-way exchanges, and Nugent et al. found both

---

[1] Ibid, p.193.

[2] Edwards et al., "Modular Allocation Technique," pp.166-167.

to produce results slightly inferior to CRAFT. Nugent's findings are supported by Ritzman and Grover.[1]

Standard of Comparison

Since, given an equal number of solutions from which to choose, and ignoring computation time, no other computer algorithm has been shown to be superior, the well known CRAFT algorithm is chosen as the standard of comparison for solution quality.[2] A discussion of computation time in chapter IV will take into account the faster algorithms as well as CRAFT.

It is important to note that because the previously documented experiments did not include multiple exchanges beyond the simple two-way exchange, unless otherwise stated, the CRAFT algorithm referred to hereinafter includes only two-way exchanges. Because of greatly increased computational effort, the effects

---

[1] Nugent et al., "Experimental Comparison," p. 165; L.P. Ritzman, "The Efficiency of Computer Algorithms for Plant Layout," Management Science, Vol. 18, No. 5, Jan. 1972, p. 247; K.L. Grover, "An Evaluation of Plant Layout Algorithms," Seattle, University of Washington, unpublished M.B.A. Research Report, 1969, p. 22.

[2] In some experimental runs of large problems, the computation time required by CRAFT was excessive, and the FRAT algorithm, which has been shown to provide comparable results, was used instead of the CRAFT algorithm.

of the addition of three-way exchanges are discussed separately in the section on computation time in chapter IV.

Visual Methods

Few studies have compared the effectiveness of the computer algorithms with that of traditional visual methods employed by industrial engineers, although indicators were available to suggest that such a comparison ought to be made. Gaunt, for instance, found a manual search procedure favoured for Travelling Salesman problems with more than forty cities.[1]

With such aids as the schematic diagram of flows shown in figure 1, and the Distance-intensity plot,[2] one might well expect good results from a visual approach.

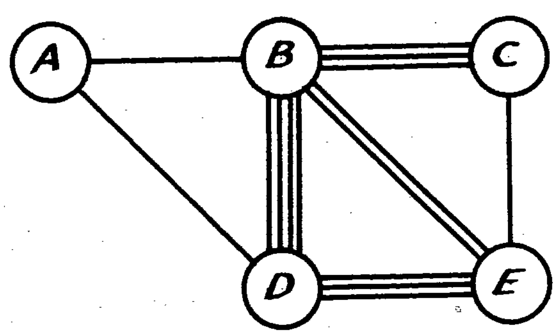

Fig. 1.--Simple one-line schematic diagram indicating intensity of flows

[1] S. Gaunt, "A Non-Computer Method Using Search for Resolving the Travelling Salesman Problem," Journal of the Canadian Operational Research Society, Vol. 6, 1968, pp.44-54.

[2] R. Muther, Systematic Materials Handling, Boston,Industrial Education Institute, 1969, p. 4-2

The reader is invited to note the simplicity of the twenty-facility layout in figure 2 when facilities are restricted to equal size, and to ponder the number of feasible solutions possible: $20! = 2.4 \times 10^{18}$. Allowing for mirror images and 180 degree rotations there are still more than 6×10^{17} feasible solutions. The reason for the difficulty in developing a good computer method for large problems can be appreciated if one considers that, even evaluating 1000 layouts per second, it would take more than nineteen million years to solve the twenty-facility problem by enumeration.

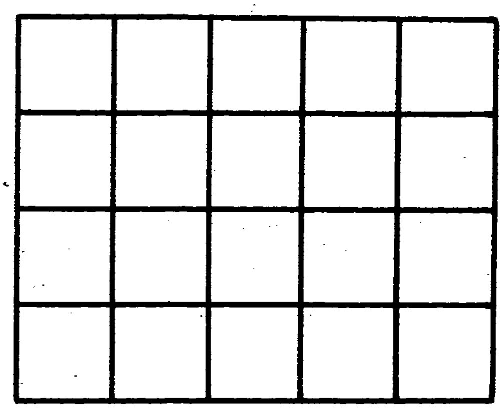

Fig. 2.--Plan showing visual simplicity of equal-area twenty-facility problem

A sixty facility problem could have more potential solutions than the estimated number of atoms in the universe(10^{80}).[1]

---

[1] J.R. Hayes, <u>Cognitive Psychology: Thinking and Creating</u>, Homewood, Ill., Dorsey Press, 1978, p. 184

Scriabin and Vergin tested three highly rated computer algorithms, CRAFT, H-63, and HC63-66, against visual methods employed by industrial engineers.[1] To make the tests comparable to those of previous studies and fair to the computer algorithms, a set of problems used by Grover[2] in a previous comparison of computer algorithms was used and no additional practical considerations were included. As in the previous study, facilities were restricted to equal areas so that the visual approach would not have an advantage.[3] To ensure proper control of the experiments, and to persuade the large group of seventy-four subjects to attempt the problems, an interactive system of computer programs was written which provided some descriptive but not prescriptive information to the subject as to his performance. The descriptive information included diagrams similar to those used in visual methods by industrial engineers.

Scriabin and Vergin found that individual solutions of the best two of the three computer algorithms (CRAFT and HC63-66)

---

[1] M. Scriabin and R.C. Vergin, "Comparison of Computer Algorithms and Visual Based Methods for Plant Layout," Management Science, Vol. 22, No. 2, Oct. 1975, pp.172-181.

[2] Grover, "Plant Layout Algorithms."

[3] No existing computer algorithm has yet been devised which handles unequal-area facilities satisfactorily. CRAFT, for instance, allows the interchange of only adjacent facilities or equal-area facilities.

were stochastically inferior to traditional visual based methods,[1] especially in large problems where man's capacity for pattern recognition seems to become an important factor (this visual capability is recognized in other areas such as chess where, in spite of years of effort and world computer chess tournaments, there is still no program available which is capable of beating a master ranked player in regular play).[2]

Using some computer algorithms' capability to produce several solutions from which the best one can be chosen, those computer algorithms can of course be shown to be superior to visual methods given a large enough sample size.[3] Care must be taken, however, in the application of this argument since it can also be used to show that the generation of random layouts is superior to the visual approach. If one takes into account the computation time, one is likely to find a strong trend favouring

---

[1] The terms "visual" and "manual" should not be confused. Not all manual methods are visual. For instance R.J. Wimmert in "A Quantitative Approach to Equipment Location in Intermittent Manufacturing," Unpublished doctoral dissertation, Purdue University, 1957, describes a manual procedure in which assignments with large costs are eliminated step by step until the problem is reduced to only one feasible solution. This is a manual heuristic procedure but not a visual one and was found in Ritzman, "Efficiency of Computer Algorithms," to be inferior to other computer algorithms when programmed.

[2] P.R. Jennings, "The Second World Computer Chess Championships," Byte, Jan. 1978, p. 118.

[3] This argument has been used by D.R. Coleman, in his note, "Plant Layout: Computers versus Humans," Management Science, Vol. 24, No. 1, Sept. 1977, pp.107-112, to defend the superiority of computer algorithms.

the use of visual methods for larger problems since multiple computer runs become impractical given a large enough problem.

Buffa objected to Scriabin and Vergin's conclusions on the basis that the problems solved in their tests were "simple" because the flows exhibited a high coefficient of variation.[1] Block tested a small group of eight subjects on another set of problems devised by Nugent et al.,[2] and found that in the presence of a low coefficient of variation of flows, the computer algorithms perform better than his eight test subjects, four of whom were lay people. His reported results do not appear to be statistically significant. Block does not indicate whether his subjects had any computational assistance to reduce the tedium of the search for a good solution (Scriabin and Vergin's subjects were provided with descriptive feedback from an interactive computer program including the calculated cost of flow in each submitted layout).

Arguments regarding the results of Scriabin and Vergin's experiments have centered on the choice between computer and visual methods. Coleman is even further sidetracked, and

---

[1] E.S. Buffa, "On a Paper by Scriabin and Vergin," Management Science, Vol. 23, No. 1, Sept. 1976, p. 104; M. Scriabin and R.C. Vergin, "Computer and Visual Methods for Plant Layout--A Rejoinder," Management Science, Vol. 23, No. 1, Sept. 1976, p. 105. This controversial topic is discussed in greater depth in chapter II.

[2] T.E. Block, "A Note on 'Comparison of Computer Algorithms and Visual Based Methods for Plant Layout' by M. Scriabin and R.C. Vergin," Management Science, Vol. 24, No. 2, Oct. 1977, pp. 235-237; Nugent et al., "Experimental Comparison."

suggests a method for generalizing the experiments in order to prove or disprove the superiority of computers to humans.[1] The important conclusion drawn by Scriabin and Vergin was that the effectiveness of computerized methods for facility layout would be increased by combining some of the capabilities of the visual approach with those of the fully automated computer algorithms.[2]

Research Objectives

In 1971 an interactive version of CORELAP had already been developed but the original program on which it is based is a construction algorithm and does not make use of the proven CRAFT algorithm. The technology already exists for the development of a better interactive computer program which combines visual aids with prescriptive feedback from the best current improvement algorithms.

The objective of this research is therefore directed instead to the possibility of developing a new fully automated computer algorithm for solving facility layout problems, the effectiveness of which is increased relative to current

---

[1] Coleman, "Computers versus Humans."

[2] Scriabin and Vergin, "Computer Algorithms and Visual Based Methods," p. 181.

algorithms, especially in those areas where current computer methods are weak.[1] It is not intended that any computer algorithm developed in the course of this research be regarded as a complete problem solving package, but rather that it be considered for inclusion in an interactive facility layout system which is designed to assist the industrial engineer in the application of his own expertise to the problem.

[1] Two areas of weakness discussed so far are large problems and problems with a high coefficient of variation of flows.

Chapter II

FLOW DOMINANCE AND LINE DOMINANCE

Flow Dominance

A single product assembly line (or product type layout) can be viewed theoretically as an extreme case of a process type layout, i.e. a process type layout for a single product. At the other extreme we would have a job shop with all jobs or products requiring a different sequence of operations or machines. In the former case the single pattern of n-1 flows is easily identified in the flow matrix. In the latter, because the flow matrix may contain up to n(n-1)/2 flows, it may be difficult to identify a dominant flow pattern, especially if the variance of the flows is small.

A simple measure which has been suggested to determine the degree to which a particular problem approaches either extreme is "flow dominance," defined by Vollmann to be the coefficient of variation (100 x std. deviation/mean) of the flows between facilities.[1] An assembly line, characterized by a sparse flow matrix, would normally exhibit higher flow dominance than a job shop.

The poor performance of existing computer algorithms versus visual methods has been justified by the contention that the problems used in the comparison of visual and computer methods

[1] Vollmann, "Relative Location of Facilities," p. 134.

were simple.[1] This contention is based on the fact that these problems exhibited high flow dominance. Buffa appears to base his reasoning on the fact that if the facilities are simply the stations of an assembly line, then the coefficient of variation of flows is high, since there are many zero flows. The inference then seems to be drawn that if the coefficient of variation of flows, or flow dominance, is as high as that of an assembly line, then the problem must be as simple as that of an assembly line.

The importance of conclusions which have been drawn in the past on the basis of flow dominance suggests that further investigation is warranted.

For some unstated reason, Vollmann included the zeroes on the diagonal of the flow matrix in his calculation of flow dominance, and others have followed suit. Given some non-zero flows, the lowest flow dominance possible occurs if all flows are equal, but since we still have zeroes on the diagonal, the coefficient of variation is $100(n-1)^{-\frac{1}{2}}$ and can be seen to vary with problem size.[2]

---

[1] See above, p.25.

[2] The formula presented here is simpler than the one suggested by T.E. Block as a lower bound on flow dominance in "On the Complexity of Facilities Layout Problems," Management Science, Vol. 25, No. 3, March 1979, pp.280-285, because Block divides by n^2-1 rather than n^2 in calculating the variance of the flow matrix. This is unnecessary since we are not attempting to estimate the coefficient of variation of the population from which the flows were drawn, but simply to calculate the coefficient of variation of the flows in the particular problem.

If problems with high flow dominance are simple, then presumably problems with low flow dominance must be complex. But does low flow dominance necessarily always imply greater difficulty in solving the problem?

For illustrative purposes, let us consider a twelve department problem in which all flows are equal and non-zero except for one which is zero. Then flow dominance is 33% which is certainly low, but the problem is easy to solve by inspection, since one could concentrate on placing first the two facilities with zero flow between them. In effect this problem is as easy to solve as one in which there is only one non-zero flow, but while the variances of the flows in these two problems would be almost the same,[1] the flow dominances differ because the average flow is much greater in the former case.

This would suggest that there is an intermediate level of flow dominance at which the difficulty of solution by inspection is greatest. This could occur, for instance, when the density of the flow matrix is approximately 50%, non-zero flows are approximately equal, and flow dominance is consequently just over 100%, which would provide some support for Buffa's and Block's arguments (see above, p. 25), since the flow dominances of problems used in Block's comparison of visual and computer methods were not much in excess of 100%.

---

[1] The variances would be identical if the zeroes on the diagonal of the flow matrix were not included.

The other extreme stated by Vollmann is an assembly line. Given a single assembly line, flow dominance increases with the number of stations. For instance, in a four-facility layout problem with symmetric flow matrix, the flow dominance of an assembly line with equal flows in 129%, while the flow dominance of a similar forty-facility assembly line would be 442%.

Block recognizes this short-coming of flow dominance in measuring problem difficulty and attempts to develop a measure (complexity rating) which is unbiased by problem size.[1] Unfortunately he too uses only the simple extreme examples (all flows equal, or a single assembly line with constant flow) as upper and lower bounds on flow dominance, so that an empirical layout problem used by Vollmann[2] would have a negative complexity rating.

Vollmann, Buffa, and Block all appear to have missed the sensitivity of flow dominance to the inclusion of a few very large flows. To see that one large flow can greatly influence flow dominance, consider a forty-facility layout problem in which all flows are close to unity. Flow dominance is approximately 16%. If we replace one of the flows by a large flow, say 1000, then with a symmetric flow matrix flow dominance increases to almost 1600%, suggesting that the

---

1 Block, "Complexity of Facilities Layout Problems."

2 Vollmann, "Relative Location of Facilities," p. 135.

problem is trivally simple in view of the relatively much lower flow dominance of an assembly line, approximately 442%. Yet once it is recognized that the two facilities involved in the large flow must be located adjacent to each other, they may be treated as a single facility and the remaining 39 facility problem with flow dominance approximately 28% may again be viewed as complex.

In addition, Vollmann ignores the cost of materials flow in his calculation of flow dominance, implying that the variation in cost per unit distance of materials handling has no effect on problem complexity.

Block bases the upper bound of flow dominance in his complexity rating on an asymmetric flow matrix. Although he was aware that the problem could be simplified by the use of a symmetric flow matrix,[1] Vollmann also based his flow dominance calculations on an asymmetric flow matrix. The use of an asymmetric flow matrix implies that it is more difficult to design a layout with flows in both directions than one with unidirectional flows.

---

[1] Vollmann,"Relative Location of Facilities", p. 26. See also p. 2 of this dissertation.

Before attempting to solve any layout problem one would normally develop a symmetric flow cost matrix as described on p. 3, above, and any measure of the problem's complexity should be based on an analysis of that symmetric matrix. Unless stated otherwise, a flow matrix mentioned hereinafter is assumed to be symmetric and to include costs.

Assuming such a symmetric flow matrix, Block's upper and lower bound formulae become:[1]

$$f_{UB} = 100\ (\tfrac{1}{2}n^2/(n-1)-1)^{\frac{1}{2}}$$

$$f_{LB} = 100(n-1)^{-\frac{1}{2}}$$

Vollmann has suggested that visual based methods may be more appropriate when flow dominance is greater than 200%.[2] Buffa and Block reiterate this claim.[3] While it is clear that, given a sufficiently high flow dominance, a good solution to a problem becomes obvious, there is doubt as to the accuracy of his dividing line of 200% since it is based on flow dominances inflated by the use of asymmetric flow matrices.

---

[1] Bock, "Complexity of Facilities Layout Problems" pp. 281-282. Block's formulae are also different because he divides by n^2-1 rather than n^2 when calculating the flow variance (see above, n.2, p. 29).

[2] Vollmann, "Relative Location of Facilities," p. 139.

[3] Buffa, "On a Paper by Scriabin and Vergin"; Block, "Note on Comparison of Computer Algorithms and Visual Based Methods."

Vollmann based his conclusion on an application of the CRAFT heuristic to layouts generated by a method he describes as "inspection" in four problems with widely differing flow dominances. Vollmann determined these flow dominances to be 135%, 201%, 252%, and 519%. The problem sizes were, respectively, 12, 10, 20, and 22 facilities. Since CRAFT was able to effect a twelve percent improvement in the total layout cost of the problem with 135% flow dominance, but was much less effective in the other three problems, Vollmann concluded that problems with flow dominance in excess of 200% could be solved successfully by inspection, while those with lower flow dominances would benefit from a computer approach.

In two of the four problems,[1] Vollmann based his calculation of flow dominance on an asymmetric flow matrix, treating flows in opposite directions as two different flows. This resulted in the inclusion of many zero flows, and consequently an inflated flow dominance. Vollmann also ignored the cost per unit distance of the flows in his calculations.

If for each of Vollmann's four problems we multiply every element of the flow matrix by the corresponding cost per unit distance, and add the resulting matrix to its transpose (as

---

[1] The ten facility engineering office layout problem with flow dominance 201% and the 22 facility empirical layout problem with 519% flow dominance .

suggested above, p. 2), the resulting flow dominances are, respectively, 117%, 144%, 254%, and 397%.[1]

Had Vollmann used this method in calculating the flow dominances, he would have concluded that visual based methods are more appropriate when flow dominance is greater than 144%, and that computer algorithms are appropriate for only a limited range of problems.

To add further doubt as to the level of flow dominance at which the choice of a visual method is indicated, there is ambiguity in Vollmann's concept of a visual method. He states: "As the degree of dominance becomes less pronounced, one would expect layout by inspection to become inappropriate."[2] Fair enough. But in attempting to determine the cutoff Level he uses an approach to give a layout by inspection which is actually a heuristic construction procedure rather than a visual method. In his procedure, all flows less than some arbitrary amount are initially ignored. The facilities which have the greatest interaction with the others on the basis of this

---

1 The problem with the highest flow dominance is the only empirical one used by Vollmann. The flow dominance of this 22 facility problem is further reduced from 397% to 320% if four facilities are eliminated from the problem which have no interaction with the others. One of these four facilities was included to meet the CRAFT requirement of a rectangular layout.

2 Vollmann, "Relative Location of Facilities," p.135.

reduced flow matrix are then placed centrally. Facilities with large flows to the centrally placed facilities are then placed adjacently. The next higher flows are then examined and facilities placed accordingly, and so on, until all facilities are placed. Following this procedure, in the twelve-facility problem with lowest flow dominance, ignoring flows less than ten , facilities H and I were placed centrally and adjacent to each other even though there is no flow between them, as can be seen in the flow matrix in figure 3. It is not likely that these facilities would have been placed adjacent to each other in a true visual approach using for instance a schematic diagram of the type shown in figure 1 (above, p. 21).

Vollmann's "inspection" method implies that smaller flows (initially eliminated in his method) are confusing. Experience in areas such as chess, where man's ability to visualize complex patterns gives him an edge over sequential procedures,[1] suggests that the elimination of some of the smaller flows may unnecessarily limit the potential results of an inspection method. It is therefore neither surprising nor significant that Vollmann obtained poor results with his inspection method, especially in the low flow dominance case, where many important flows are initially ignored.

---

[1] P.G. Rushton and T.A. Marsland,"Current Chess Programs: A Summary of their Potential and Limitations," Canadian Journal of Operational Research and Information Processing, Vol. II, 1973, pp.13-20.

	A	B	C	D	E	F	G	H	I	J	K	L
A	0	5	2	4	1	0	0	6	2	1	1	1
B	5	0	3	0	2	2	2	0	4	5	0	0
C	2	3	0	0	0	0	0	5	5	2	2	2
D	4	0	0	0	5	2	2	10	0	0	5	5
E	1	2	0	5	0	10	0	0	0	5	1	1
F	0	2	0	2	10	0	5	1	1	5	4	0
G	0	2	0	2	0	5	0	10	5	2	3	3
H	6	0	5	10	0	1	10	0	0	0	5	0
I	2	4	5	0	0	1	5	0	0	0	10	10
J	1	5	2	0	5	5	2	0	0	0	5	0
K	1	0	2	5	1	4	3	5	10	5	0	2
L	1	0	2	5	1	0	3	0	10	0	2	0

Fig. 3.--Flow matrix for twelve-facility problem used by Vollmann in his experiments

Vollmann's results on this particular problem may nevertheless be representative of the performance of a visual method since Block subsequently obtained similar although not statistically significant results[1] in an experimental comparison of CRAFT and visual methods, using a set of problems which included the same twelve-facility problem used by Vollmann.[2]

Effect of Flow Dominance on Computation Time

Mojena et al. have attempted to determine the extent to which flow dominance and other measures can affect the computational time of one computer algorithm for facility layout.[3] They found that regression models based on twenty-two initial measures of the flow matrix were poor predictors of computer run time, although they did find that flow dominance

---

1 Block, "Note on Comparison of Computer Algorithms and Visual Based Methods."

2 Vollmann's twelve-facility problem is the same problem originally used by Hillier in "Quantitative Tools for Plant Layout Analysis," and upon which the set of problems used by Nugent et al. in "Experimental Comparison" is based.

3 R. Mojena, T.E. Vollmann, and Y. Okamoto, "On Predicting Computational Time of a Branch and Bound Algorithm for the Assignment of Facilities," Decision Science, Vol. 7, 1976, pp. 856-867.

and two other related variables[1] significantly discriminated among groups of problems requiring low, medium, or high amount of computer effort.

These findings cannot of course be extrapolated to other methods of solving the layout problem, but it is interesting to note that computer as well as visual methods appear to benefit from the "simplicity" of a layout problem.

Effect of Flow Dominance on Efficiency of Computer Algorithms and Visual Methods

To test the extent to which the quality of solutions varies with flow dominance it is desirable to have some measure of quality. One approach considered was the generation of the distribution of the costs of random layouts. To this end, 1000 random layouts were produced for the thirty facility problem of Nugent et al.

The best randomly chosen solution in 1000 was only 8½% better than the average random solution, and 3.2 standard deviations from the mean, while the average CRAFT solution recorded by Nugent et al. improves on the average random layout by more than 20% and is more than eight standard deviations from the mean.

---

[1] The other two variables, besides the coefficient of variation of flows, were the c. of v. of first order differences in the flows, and the c. of v. of the total flows through each facility.

Since most experimental results would involve values far from the mean, this approach was discarded in favour of one which relates a given solution to a lower bound, as well as the mean (in all but the smallest problems the optimum solution is usually unknown). Nevertheless this small test did emphasize the complexity of the layout problem and shows why the generation of random solutions is not a good approach to solving the layout problem.

Hillier has suggested an efficiency criterion based on a lower bound to the optimal solution:[1]

$$\text{efficiency} = 100(A-S)/(A-LB)$$

where:

A = average randomly chosen solution value

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{k=1}^{n-1} \sum_{l=k+1}^{n} f_{ij} d_{kl} / \left[n(n-1)/2 \right]$$

= average distance x sum of flows,

S = value of the solution for which an efficiency rating is desired, and

LB = lower bound for the optimal solution to the problem

A is easy to calculate, especially for a rectangular problem where the average rectilinear distance can be shown to be equal to (no. of rows + no. of columns)/3, but the calculation of the lower bound LB is more involved, requiring first the calculation of a lower bound on the cost of locating each facility

[1] Hillier, "Quantitative Tools for Plant Layout Analysis," p.37.

individually in each location, then using an assignment algorithm to minimize the total cost of such individual assignments.

For most non-trivial problems, the lower bound calculated in this manner is lower than the optimum, but greater than the simpler bound used by Nugent *et al*. and Ritzman in their studies.[1] Thus Hillier's efficiency criterion is better (although it requires more calculation).

Flow dominance and the efficiency of visual methods

Table 2 shows the efficiency, based on Hillier's criterion, of visual solutions to a set of problems originally devised by Nugent *et al*.[2] and used by Block[3] in his experimental comparison of visual and computer methods.

The first striking feature is that efficiency decreases as problem size increases. This result is not unexpected and occurs consistently in these experiments regardless of the method of solution or the level of flow dominance. This decrease in efficiency cannot be entirely attributed to the increased difficulty of finding the best solution to larger problems, as can be seen by the fact that the theoretical efficiency of the optimum solution also decreases with an

---

[1] Nugent *et al*., "Experimental Comparison"; Ritzman, "Efficiency of Computer Algorithms."

[2] Nugent *et al*., "Experimental Comparison."

[3] Block, "Note on Comparison of Computer Algorithms and Visual Based Methods."

TABLE 2

EFFICIENCY OF VISUAL SOLUTIONS TO PROBLEMS WITH LOW FLOW DOMINANCE

No. of facilities	Flow dominance (percent)	Best visual layout cost	Efficiency[a] (percent)
5	108	25	100
6	129	43	89
7	111	74	84
8	128	107	77
12	117	320	54
15	106	631	52
20	104	1378	48
30	112	3408	37

a The true efficiency of the solutions to the four smallest problems is 100% since the solutions are known to be optimum.

increase in problem size. Thus the decrease in efficiency of the solutions really reflects in part the inefficiency of Hillier's efficiency criterion itself, and caution is required in interpreting these results.

It is more interesting to compare the efficiency of the solutions in table 2 with those in table 3, showing the efficiency of visual solutions to Grover's set of problems[1] obtained in Scriabin and Vergin's experiments.[2] The solutions to Grover's problems with higher flow dominance are more efficient according to Hillier's criterion.

Flow dominance and the efficiency of CRAFT

Tables 4 and 5 show respectively the efficiency of the best of five CRAFT solutions to the same sets of problems with low and high flow dominance. As with the visual solutions, the efficiency of CRAFT solutions is higher in the group of problems with high flow dominance.

These two groups of problems are used here because of the availability of visual solutions recorded under controlled conditions. There may be other differences besides flow dominance affecting the quality of solutions to these problems. To provide a better test of the effect of flow dominance on the quality of CRAFT solutions, a new set of problems was generated

---

1 Grover, "Plant Layout Algorithms."

2 Scriabin and Vergin, "Computer Algorithms and Visual Based Methods."

TABLE 3

EFFICIENCY OF VISUAL SOLUTIONS TO PROBLEMS WITH HIGH FLOW DOMINANCE

No. of facilities	Flow dominance (percent)	Best visual layout cost [b]	Efficiency (percent)
5	269	64.02	94
6	263	80.32	95
7	199	163.28	85
8	213	215.86	91
8(alt) [a]	213	214.76	89
10	233	269.92	89
12	256	314.10	89
15	248	511.40	89
20	254	1109.68	83

a Grover used two different plant shapes with the same eight-facility flow information.

b True costs are half of those reported. Costs were doubled to maintain consistency with Grover's original results in which all flows were counted twice.

TABLE 4

EFFICIENCY OF CRAFT SOLUTIONS TO PROBLEMS WITH LOW FLOW DOMINANCE

No. of facilities	Flow dominance (percent)	Best of five CRAFT layout costs	Efficiency (percent)
5	108	25	100
6	129	43	89
7	111	74	84
8	128	107	77
12	117	289	74
15	106	583	67
20	104	1324	56
30	112	3148	51

TABLE 5

EFFICIENCY OF CRAFT SOLUTIONS TO PROBLEMS
WITH HIGH FLOW DOMINANCE

No. of facilities	Flow dominance (percent)	Best of five CRAFT layout costs[a]	Efficiency (percent)
5	269	64.02	94
6	263	80.32	95
7	199	163.28	85
8(alt.)	213	214.76	89
10	233	279.02	86
12	256	331.58	83
15	248	531.14	85
20	254	1186.66	77

a True costs are half of those reported. Grover originally counted all flows twice.

with flows selected from a Poisson distribution. Problem size was held constant at thirty facilities. The choice of Poisson flows allows the easy generation of problems with different flow dominances, since the variance of a Poisson distribution is equal to its mean, and in a problem with Poisson flows with mean λ the coefficient of variation is $100\lambda^{-\frac{1}{2}}$, which is approximately equal to the flow dominance.[1] Five problems were generated, with flow dominance varying from 96% to 282%. Appendix A contains the flow-distance matrices for these problems.

Table 6 shows that the efficiency of CRAFT again increases as flow dominance increases.

Line Dominance

An additional bias may have been inadvertently introduced by Nugent et al. when they used problems generated by randomly redistributing flows from Hillier's twelve department problem.

Flow dominance is a measure which attempts to describe the degree to which a problem contains assembly-line-like flows.[2] Even if a problem contains several assembly lines, by randomly

---

[1] Following the precedent set by Vollmann and including the diagonal elements of the flow matrix, flow dominance is $100\left[(n+\lambda)/(\lambda(n-1))\right]^{\frac{1}{2}}$.

[2] Vollmann, "Relative Location of Facilities," p. 134.

TABLE 6

EFFICIENCY OF CRAFT SOLUTIONS TO PROBLEMS[a] WITH POISSON FLOWS AND VARYING FLOW DOMINANCE

Flow dominance (percent)	Best of five CRAFT layout costs	Efficiency (percent)
96	1343	48
115	861	52
144	508	51
199	255	63
282	98	74

a All five problems were the same thirty-facility size.

rearranging the flows one is likely to break up those lines, reducing the true flow dominance (hereinafter called line dominance to distinguish it from Vollmann's measure), but of course not changing the coefficient of variation of the flows. Thus in any problem generated by randomly placing flows in the flow matrix, the coefficient of variation is likely to overstate the line dominance and, for any calculated flow dominance, real problems should tend to have a higher line dominance than problems generated by the method used by Nugent <u>et al</u>.

In practice, one would expect the flow matrix to be made up of the addition of flows along several job routes. Thus the flows would not likely be randomly distributed over the flow matrix.

Unfortunately, to measure line dominance it is necessary to get at the data behind the flow matrix. To this end, a set of problems was generated in which job routes with specified route lengths and flows, rather than simply individual flows, were randomly chosen and superimposed on the flow matrix. The flow along each route was chosen randomly from a uniform distribution but held constant over the length of the route. No attempt was made to generate realistic problems since the purpose of these problems is to investigate the effect of a particular arrangement of flows on solution quality, and realistic problems likely contain a mixture of different arrangements.

By varying the number of routes and their lengths, but keeping the product of these two numbers constant,flow matrix density and flow dominance remain approximately constant and we can study the effects of varying line dominance. Line dominance is high in a problem with only a few long job routes, while a problem with many very short job routes has low line dominance.

Five problems were generated, each with twenty facilities. The job route length was varied from 1 to 15, while the number of jobs (lines) was chosen in such a way that for each problem the product (line length) x (number of lines) equalled 120. For instance, in the only problem with no line dominance, line length was 1 and the number of lines was 120, so that the flow matrix was made up of 120 randomly placed flows. Appendix B contains the flow-distance matrices for these five problems with varying line dominance.

Five CRAFT runs were performed from randomly chosen initial layouts for each of the five problems, and the efficiency based on Hillier's criterion was calculated for the best CRAFT results on each problem. Table 7 shows that as line dominance increases the efficiency of CRAFT decreases.

Since the presence of flow dominance has been shown to increase the efficiency of CRAFT, while the presence of line dominance has the opposite effect, it can be concluded that the coefficient of variation of flows, Vollmann's measure of flow dominance, is not a good indicator of the presence of line dominance as originally intended by Vollmann.

TABLE 7

EFFICIENCY OF CRAFT SOLUTIONS TO PROBLEMS[a] WITH VARYING LINE DOMINANCE[b]

Line Length	Number of lines	Flow domin-ance (percent)	Best of five CRAFT layout costs	Efficiency (percent)
1	120	143	710	70
5	24	140	687	62
8	15	142	870	56
10	12	143	729	56
15	8	143	622	54

a All five problems were the same twenty-facility size.

b A problem in which the flows are concentrated in a few long lines or job routes has higher line dominance than a problem in which the flows are distributed among many short lines.

Need for Further Development

Although existing computer algorithms have been shown to provide good results to problems with low flow dominance, these algorithms have been shown to be inferior to visual methods when flow dominance is high, and may also be inferior to visual methods in the presence of line dominance. On the other hand, computer algorithms tend to provide more consistent results than visual methods.[1]

It would therefore seem desirable to develop a consistently good computer algorithm which can handle problems with high as well as low flow dominance, and which works well in the presence of line dominance. Such an algorithm should ideally be competitive with CRAFT in problems with low flow dominance, and competitive with visual methods in problems with high flow dominance and line dominance. In addition, since the efficiency of all current methods decreases as problem size increases, any new algorithm should be as effective as possible on larger problems.

[1] Scriabin and Vergin, "Computer Algorithms and Visual Based Methods," p. 180.

CHAPTER III

A CLUSTER ANALYTIC APPROACH

Rationale

The basic exchange heuristic derives its power to achieve good solutions in a reasonable amount of time from its simplicity, but that same simplicity often stands in the way of further improvements.

Consider for instance the suboptimal layout with high line dominance shown in figure 4. Each arrow connecting a pair of facilities represents a unit flow between those facilities. Facilities not connected by arrows have no flow between them. No further improvement is possible with only two-way exchanges, however the layout can be improved by reversing the lines

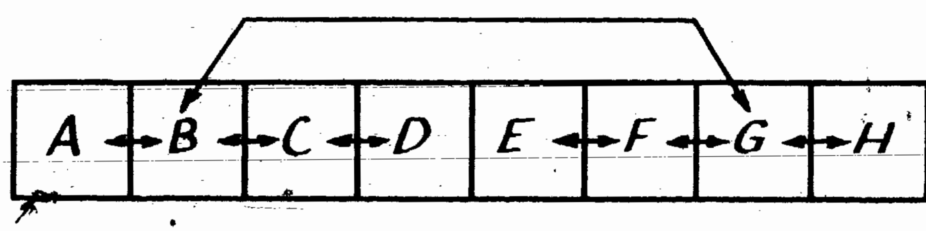

Fig. 4--Example of layout requiring movement of clusters to effect improvement

A-B-C-D and E-F-G-H, or by interchanging them, i.e. by manipulation of clusters of facilities.

Another example of a suboptimal layout which cannot be improved by a two-way exchange is shown in figure 5. An appropriate rotation of either the left-hand or right-hand group of four facilities would improve the solution.

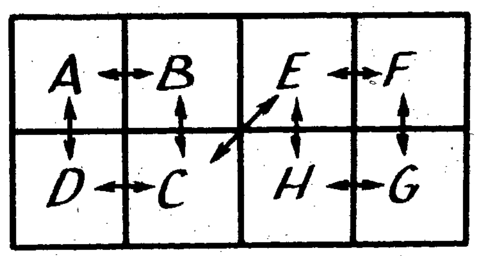

Fig. 5.--Example of layout which can be improved by rotation of cluster.

It is not difficult to see that with high line dominance such groups could be quite large, requiring multi-way exchanges or movements of larger clusters of facilities to effect further improvement. It is also possible for a problem with relatively lower line dominance to contain several smaller groups or clusters of facilities poorly located or oriented relative to each other.

A two-stage algorithm, which is initially unhampered by the requirement that one facility cannot move without displacing another (i.e. unconstrained by the boundaries of the plant or by the size of the areas occupied by the facilities), would tend to avoid such incorrect placement of whole clusters of facilities relative to each other.

Such an approach may also work well under general conditions including that of low flow dominance if a layout based on the solution of the first unconstrained stage is good enough. That is, if the first stage can provide CRAFT with an initial solution competitive with an average CRAFT solution, but arrived at by other means than by two-way exchanges, then the two-way exchange heuristic of CRAFT should have a greater chance of improving the solution further.

In fact, Edwards et al. attempted this approach, using the results of MAT as a starting solution for CRAFT and another two-way interchange algorithm, with modest (not significant) success.[1] However, the initial solution provided by MAT was not competitive with CRAFT (in all but the smallest problems, the MAT result was worse than the median CRAFT solution and, in the largest thirty-facility problem, the MAT cost was sixteen percent worse than the average CRAFT solution).

One of the reasons visual methods performed better than the computer algorithms on large problems is that all the computer algorithms essentially consider the movement of only one or two

---

1 Edwards et al., "Modular Allocation Technique."

facilities at a time,[1] while the visual approach is not so limited. Thus in a case where it might be profitable to move a cluster, the computer algorithm may be unable to do so because of the strong influence the cluster has on each member facility.

In solving the facility layout problem by traditional methods, the industrial engineer first uses the schematic line diagram to help him develop a picture of where the facilities would like to be relative to each other. He then squeezes the picture into the constrained plant floor space in a separate step. A new algorithm will now be presented which incorporates this overall strategy of the traditional visual approach.

Facility Layout by Analysis of Clusters (FLAC)

The proposed algorithm, FLAC (Facility Layout by Analysis of Clusters), consists of three distinct stages.

Stage 1 includes an essential ingredient from the traditional visual approach. It makes use of techniques from the field of cluster analysis to locate facilities on an

---

[1] Exchange involving more than two facilities require excessive computation time. A three-way exchange algorithm is discussed in chapter IV.

unconstrained map in such a way that distances between them are as far as possible inversely related to flow. (This map is equivalent to the industrial engineer's schematic diagram of flow intensity).

During stage 2 facilities are assigned to locations in the constrained plant space in such a way as to maintain as far as possible their positions attained in stage 1. It will be seen that at this stage the original unwieldy quadratic assignment problem is reduced to a simple assignment problem.

It is expected that stages 1 and 2 will give quite good results since clusters of facilities will tend to be well located relative to each other. However, some individual facilities will likely require movement in order to achieve better results. Such fine adjustment will be provided by a CRAFT-type of algorithm in stage 3.

Hypothesis

It is hypothesized that FLAC will give better results than CRAFT alone in larger problems with high flow dominance and line dominance, and will provide results competitive with CRAFT in smaller problems and those with low flow dominance.

Stage 1: Developing the Unconstrained Configuration

The first stage involves the use of the techniques of multidimensional scaling (MDS).[1]

MDS is concerned with determining the configuration of a set of points in a real space with the fewest dimensions which adequately represent the real or perceived distances or dissimilarities between the points. Early development of MDS centered on a metric approach now regarded as classical.[2] Recent emphasis has been on the development of non-metric approaches in which the objective is to provide a picture in which distances between objects (stimuli, persons, nations, etc.) are ranked in such a way that distances are monotonically increasing as similarities decrease.[3]

---

[1] For an introduction to multidimensional scaling the interested reader is referred to Shlomo Maital, "Multidimensional Scaling: Some Econometric Applications," Journal of Econometrics, Vol. 8, 1978, pp. 33-46. A more thorough treatment is available in R.N. Shepard, A.K. Romney, and S.B. Nerlove, Multidimensional Scaling: Theory and Applications in the Behavioral Sciences, Seminar Press, New York, 1972.

[2] W.S. Torgerson, "Multidimensional Scaling: I. Theory and Method," Psychometrika, Vol. 17, 1952, pp.401-419; G. Young and A.S. Householder, "Discussion of a Set of Points in Terms of their Mutual Distances," Psychometrika, Vol. 3, 1938, pp. 19-22.

[3] R.N. Shepard, "The Analysis of Proximities: Multidimensional Scaling with an Unknown Distance Function I," Psychometrika, Vol. 27, No. 2, June 1962, pp.125-140; Shepard, "The Analysis of Proximities: Multidimensional Scaling with an Unknown Distance Function II," Psychometrika, Vol. 27, No. 3, Sept., 1962, pp.219-246; J.B. Kruskal, "Multidimensional Scaling by Optimizing Goodness of Fit to a Non-metric Hypothesis," Psychometrika, Vol. 29, No. 1, March 1964, pp.1-27; Kruskal, "Non-metric Multidimensional Scaling: A Numerical Method," Psychometrika, Vol. 29, No. 2, June 1964, pp.115-129.

The approach used in stage 1 of the FLAC algorithm is roughly the same as that used by Forrest Young in his highly rated computer program TORSCA[1] used for non-metric multidimensional scaling.

The basic approach consists of the development of an initial configuration of the facilities using factor analysis of a transformation of the flow-cost data,[2] followed by an iterative routine which attempts to improve on the initial configuration. In facility layout, the objective is to construct a two-dimensional[3] picture in which the facilities are positioned so that, as much as possible, the distances between them are inversely proportional to the flows.[4]

---

[1] F.W. Young and W.S. Torgerson, "TORSCA, a FORTRAN IV Program for Shepard-Kruskal Multidimensional Scaling Analysis," <u>Behavioral Science</u>, Vol. 12, 1967, p. 498.

[2] The flow matrix is assumed already to have been multiplied by the cost per unit distance matrix and made symmetric following the procedure described above, p. 3. Further transformations are described in detail on pp. 61-80.

[3] TORSCA attempts to determine the fewest dimensions necessary to represent adequately the relationships between the points or stimuli. In FLAC, the number of dimensions is predetermined by the physical nature of the problem. FLAC is designed to handle two-dimensional layouts, however there is no theoretical obstacle to generalizing the approach to handle three-dimensional problems.

[4] It is not practical to consider minimization of layout cost as the objective at this stage since in an unconstrained layout total material handling cost would be at a minimum with all facilities at the same location. It is important to maintain some separation of even those facilities with heavy interaction so that they can be located correctly relative to each other in stage 2. The chosen intermediate objective is discussed further in the section entitled "Transformation of flows to dissimilarities," below.

In spite of the transformations which will be applied to the flow-cost data, it will be shown that the use of interval measures remains valid, permitting the use of a metric approach in FLAC.[1] Some features are included which increase the effectiveness of the algorithm when used for the specific purpose of facility layout.

Distance measurements in stage 1

Usually the industrial engineer does not initially concern himself with the overall orientation of the unconstrained diagram, since he can easily rotate the diagram during the second stage while squeezing the facilities into the available space. Similarly, in the first stage of FLAC, the polar orientation of the facilities is not considered explicitly, but will be addressed in stage 2 while fitting the facilities into the constrained layout.

In most layout models "city block" or rectilinear distances are assumed, to allow for flows along aisles or corridors. The ultimate objective of FLAC also is based on a rectilinear distance criterion, however since the first stage develops only a rough configuration, the final orientation of which may be altered in stage 2, Euclidean distances are

---

[1] TORSCA is complicated by the need to use non-metric approaches since the program is aimed at the analysis of sociological problems using ordinal measurements. In attempting to improve the rank order of the distances, Young's algorithm iteratively adjusts the initial dissimilarities as well as the distances. Such alteration of the original data is incompatible with the metric approach used in FLAC.

initially used in the first stage.[1]

Factor analysis

Torgerson has shown how interpoint distances can be used to determine projections on axes using factor analysis.[2]

Briefly, the procedure is to first construct an n x n matrix B of the scalar product of vectors from the centroid to all pairs of points. This matrix B can be shown to be equal to AA^T, where A is the n x 2 matrix of coordinates. The matrix B is then factored by obtaining its eigenvalues and eigenvectors (if the distances can be accurately represented in two dimensions, the eigenvectors associated with the largest eigenvalues will account for all the variability in the data). The results of the factor analysis are then used to develop the matrix A of coordinates, which form the basis of a map.

---

1 The reader may be tempted to object to the use of Euclidean distances on the basis that this approach is in conflict with the ultimate objectives of the model. Firstly, this approach is suggested for the practical reason mentioned above that the final orientation is not known. Secondly, there must be a high correlation between Euclidean and rectilinear distances, and thirdly, the pursuit of alternatives which explicitly attempt directly to achieve the ultimate objective of a minimum cost constrained layout, tend to lead one back to the optimization approaches which have been shown to be impractical.

2 W.S. Torgerson, Theory and Methods of Scaling, New York, Wiley, 1958, pp. 254-259.

More precisely, Torgerson has shown that each element of the B matrix[1] can be calculated directly from the inter-point distances using the formula

$$b_{jk} = (n^{-1}\sum_{j} d_{jk}^{2} + n^{-1}\sum_{k} d_{jk}^{2} - n^{-2}\sum_{j}\sum_{k} d_{jk}^{2} - d_{jk}^{2})/2.$$

In general, a problem with n points can be represented in n-1 or less dimensions. If the interpoint distances between n points were actually measured from a two-dimensional map, then the eigenvectors associated with the largest two eigenvalues would form the basis of the coordinates of the n points (each eigenvector multiplied by the square root of its associated eigenvalue is the vector of coordinates on one of the two axes). The remaining eigenvalues in such a problem would be zero. If the interpoint distances were not measured from a map, but are instead measures of "dissimilarity," then more than two dimensions may be necessary to accurately portray these dissimilarities.

The coefficient of variation of the dissimilarities is related to the number of dimensions required to accurately portray the relationships. If the coefficient of variation is small, then more dimensions may be needed, while if the coefficient of

---

[1] Torgerson refers to this as the B* matrix in loc.cit.

variation is large, then possibly two dimensions are too many and a calculated eigenvalue may be negative, indicating that the basic triangular inequality rule is violated and one or more dimensions are in imaginary space.

Transformation of flows to dissimilarities

In the facility layout problem, the dissimilarities are the inverse of the flows between facilities, scaled by a constant factor so that the average dissimilarity equals the average distance between available locations in the layout.

Since we are interested in constructing a two-dimensional plan, only the two largest positive eigenvalues and associated eigenvectors are used in the construction of that plan.

Two difficulties are immediately encountered when we try to implement this approach. The first is that in calculating the inverse of the flows, some of those flows are zero. This can be overcome by adding an arbitrary constant to each flow, but what size should that constant be, and does the addition of a constant alter the original problem? The second difficulty is that, as mentioned above, it is possible for one of the largest eigenvalues to be negative, implying imaginary space. The power method[1] which is an efficient method of obtaining the largest eigenvalue of a large matrix, chooses the eigenvalue whose magnitude is largest but whose sign may be negative. This second difficulty can also be overcome by the addition of a constant to the flows since the effect of increasing all flows by a constant is to reduce the coefficient of variation of the dissimilarities, leading to an algebraic increase in all eigenvalues.[2]

The solution to these two difficulties lies in the choice of a suitable additive constant, but first it is necessary to establish that the addition of a constant to all flows does not alter the problem and therefore does not invalidate the metric approach, i.e. that of attempting to achieve distances inversely proportional to the altered flows.

It can be seen that the problem is not altered if we consider any pair of solutions to a given layout problem. If the first solution yields a lower total materials handling cost than

---

[1] The method used in FLAC is derived from that suggested in M.L. James, G.M. Smith, and J.C. Wolford, Applied Numerical Methods for Digital Computation with FORTRAN, Scranton, Penn., International Textbook Co., 1967, pp.252-256. The method of deflation is used in obtaining the second largest eigenvalue and associated eigenvector, as described in Charles B. Tompkins and Walter L. Wilson, Jr., Elementary Numerical Analysis, Englewood Cliffs, N.J., Prentice-Hall, 1969, pp.162-166.

[2] W.S. Torgerson originally observed that the effect of increasing all interpoint distances is to algebraically increase all the eigenvalues while the eigenvectors are minimally affected, according to Forrest W. Young, "A FORTRAN IV Program for Non-metric Multidimensional Scaling," Research Report, the L.L. Thurstone Psychometric Laboratory, University of North Carolina, Chapel Hill, N.C., March, 1968, p. 4.

the second using the original flows, then the first must still yield a lower cost than the second after adding a constant c to each flow f_{ij}, since the effect of such an addition is to increase total cost by the constant[1]

$$c \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{ij}.$$

Thus the optimum solution to transformed problem is also the optimum solution to the original problem, and a metric approach to solving the transformed problem is justified.

<u>Additive constant</u>

A negative eigenvalue indicates a violation of the triangular inequality rule. For example, suppose we try to plot the three points whose interpoint distances, or dissimilarities,

---

[1] The total cost of any layout in the transformed problem is

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (f_{ij} + c) d_{ij}$$

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} f_{ij} d_{ij} + c \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{ij}.$$

are given by the inverse of the flows in the matrix.[1]

$$F=\begin{bmatrix} 0 & .8 & .3 \\ & 0 & .1\dot{3} \\ & & 0 \end{bmatrix}.$$

The dissimilarity matrix (inverse of flows) is then

$$\begin{bmatrix} 0 & 1.25 & 3.\dot{3} \\ & 0 & 7.5 \\ & & 0 \end{bmatrix}$$

If the matrix of distances between locations in the layout is

$$\begin{bmatrix} 0 & 1 & 1 \\ & 0 & 2 \\ & & 0 \end{bmatrix}$$

then the dissimilarity matrix, adjusted so that the scale is similar to the layout plan, i.e. so that the average dissimilarity equals the average distance between locations in the layout, is

$$D=\begin{bmatrix} 0 & .4138 & 1.1034 \\ & 0 & 2.4828 \\ & & 0 \end{bmatrix}.$$

---

[1] For convenience, only the portion of the symmetric matrix above the diagonal is shown.

Using Torgerson's formula (above, p. 62), we first calculate the symmetric B matrix

$$B = \begin{bmatrix} -.3763 & .3625 & .0138 \\ & 1.2726 & -1.6352 \\ & & 1.6214 \end{bmatrix}.$$

The largest eigenvalue β_1 and associated eigenvector X_1 are

$$\beta_1 = 3.107, \qquad X_1 = \begin{bmatrix} -.0669 \\ -.6712 \\ .7382 \end{bmatrix},$$

and the coordinates in the first dimension are

$$A_1 = \sqrt{\beta_1} X_1 = \begin{bmatrix} -.118 \\ -1.183 \\ 1.301 \end{bmatrix}.$$

The next largest eigenvalue of the matrix B is

$$\beta_2 = -.589$$

which is negative, and the three interpoint distances cannot be accurately represented physically in two dimensions.

An appealing solution is to add the smallest possible constant to the flows so as to avoid the violation of the triangular inequality rule.

In this example, the addition of 0.2 to each flow would have the desired effect of eliminating the violation of the triangular inequality rule as can be seen from the resulting unadjusted dissimilarity matrix:

$$F=\begin{bmatrix} 0 & 1 & .5 \\ & 0 & .\dot{3} \\ & & 0 \end{bmatrix}, \qquad D=\begin{bmatrix} 0 & 1 & 2 \\ & 0 & 3 \\ & & 0 \end{bmatrix}.$$

Adjusted so that the average dissimilarity equals the average distance in the layout,

$$D=\begin{bmatrix} 0 & .\dot{6} & 1.\dot{3} \\ & 0 & 2 \\ & & 0 \end{bmatrix}.$$

The B matrix, from Torgerson's formula (above, p. 62), is

$$B=\begin{bmatrix} .0494 & .1975 & -.2469 \\ & .7901 & -.9877 \\ & & 1.2346 \end{bmatrix},$$

and the largest eigenvalue and associated eigenvector are

$$\beta_1 = 2.074, \qquad X_1 = \begin{bmatrix} -.1543 \\ -.6172 \\ .7715 \end{bmatrix}.$$

The coordinates in the first dimension are

$$A_1 = \sqrt{\beta_1}\, X_1 = \begin{bmatrix} -.\dot{2} \\ -.\dot{8} \\ 1.\dot{1} \end{bmatrix}.$$

The second eigenvalue of the B matrix is zero, and the three facilities plot along the first axis as shown in figure 6.

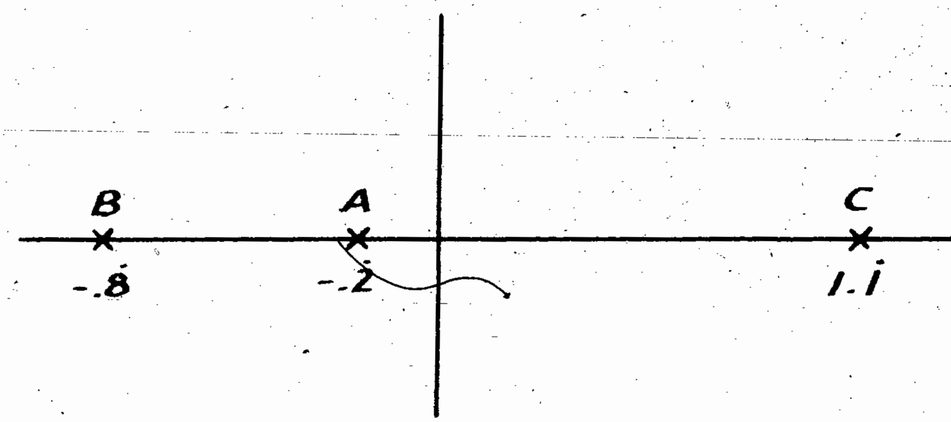

Fig. 6.--Configuration resulting from factor analysis if the smallest constant is added to the flows, which avoids violation of the triangular inequality rule.

The disadvantage of this solution in the facility layout problem is that in adding the smallest possible constant to the flows, we tend to make the facilities plot along one axis, and we do not obtain sufficient separation along the second axis to assist in positioning the facilities appropriately relative to each other in both dimensions.

Bearing in mind that in stage 2 of FLAC we wish to minimize the distance each facility is moved from its location in unconstrained space to an available location in the constrained layout, it is intuitively appealing to develop an initial configuration in which the standard deviation of the interfacility distances, as well as the average distance, is close to that in the final layout.[1]

To achieve appropriate separation in both dimensions, FLAC calculates a constant which, when added to the flows, generates dissimilarities which have the same coefficient of variation as that of the distances between available locations in the final layout. The coefficient of variation, rather than the standard deviation, is used because it allows the independent determination of a second scaling constant which maintains the average dissimilarity equal to the average distance in the constrained layout.

More precisely, two constants, c_1 and c_2, are calculated such that

$$\frac{\left\{(n(n-1)/2)^{-1}\left[\sum_{ij}\sum c_2(f_{ij}+c_1)^{-1}-(n(n-1)/2)^{-1}\sum_{kl}\sum c_2(f_{kl}+c_1)^{-1}\right]^2\right\}^{\frac{1}{2}}}{(n(n-1)/2)^{-1}\sum_{ij}\sum c_2(f_{ij}+c_1)^{-1}}=CV \quad (1)$$

and $$\sum_{ij}\sum c_2(f_{ij}+c_1)^{-1} = \sum_{ij}\sum d_{ij} \quad (2)$$

---

[1] The industrial engineer taking a visual approach makes a similar intuitive adjustment to the distances between facilities in his initial schematic diagram showing intensity of flows. It requires only a brief glance at several such schematic diagrams to see that he does not attempt to achieve distances exactly inversely proportional to flows, but rather maintains reasonable distances probably influenced by his knowledge that the facilities must eventually be located in the constrained layout.

where:

c_1 = the additive constant, the partial purpose of which is to reduce violations of the triangular inequality rule by altering the coefficient of variation of the dissimilarities,

c_2 = a scaling constant used to ensure that the average dissimilarity equals the average distance between available locations,

CV = the coefficient of variation of the distances between available locations in the layout, and

$\sum\sum_{ij}$ is shorthand notation for $\sum_{i=i}^{n-1} \sum_{j=i+1}^{n}$.

Equation (1) can be simplified to

$$\left[\sum\sum_{ij}(f_{ij}+c_1)^{-1}\right]^{-2} \sum\sum_{ij}(f_{ij}+c_1)^{-2} = (CV^2+1)(n(n-1)/2)^{-1} \qquad (3)$$

where the right-hand side is a constant for any given problem. The second scaling constant can be calculated subsequently from

$$c_2 = \sum\sum_{ij} d_{ij} \left[\sum\sum_{ij}(f_{ij}+c_1)^{-1}\right]^{-1}. \qquad (4)$$

Using the first derivative of the left-hand side of (3) with respect to c_1,

$$\frac{2\left\{\left[\sum\sum_{ij}(f_{ij}+c_1)^{-2}\right]^2 - \sum\sum_{ij}(f_{ij}+c_1)^{-3} \sum\sum_{ij}(f_{ij}+c_1)^{-1}\right\}}{\left[\sum\sum(f_{ij}+c_1)^{-1}\right]^3}$$

and Newton's method, FLAC iterates to a solution for c_1. Using an initial guess for c_1 equal to one half of the average flow, a satisfactory value is calculated even for problems as large as forty facilities in three or less iterations.

If this calculation leads to a negative constant, c_1 is set to a very small positive value and iteration continues. If a negative constant is generated twice, the final constant is arbitrarily fixed at that small value.[1] This occurs when the coefficient of variation of the dissimilarities is less than that of the distances in the layout, as for instance in a large problem with a very sparse matrix (since most flows are zero, most of the dissimilarities are very large but equal to each other, and the average dissimilarity is large). This causes no difficulty since the resulting configuration will tend to have good separation between facilities.

In the example problem, c_1 is calculated to be .294 in two iterations, resulting in the flow matrix

$$F = \begin{bmatrix} 0 & 1.094 & .594 \\ & 0 & .427 \\ & & 0 \end{bmatrix}.$$

The adjusted dissimilarity matrix is

[1] In the few cases where this occurred in these experiments, the value .0001 worked well, however, for greater generality this value should be a function of, say, the average or smallest flow.

$$D = \begin{bmatrix} 0 & .740 & 1.363 \\ & 0 & 1.896 \\ & & 0 \end{bmatrix}$$

The average 1.333 and coefficient of variation 35.4% of this dissimilarity matrix match the corresponding parameters of the matrix of distances between locations in the layout to three significant figures (see above, p. 66).

The B matrix is

$$B = \begin{bmatrix} .135 & .151 & -.286 \\ & .714 & -.865 \\ & & 1.151 \end{bmatrix}$$

The largest two eigenvalues and associated eigenvectors are

$$\beta = \begin{bmatrix} 1.883 \\ .117 \end{bmatrix}, \quad X = \begin{bmatrix} -.179 & -.797 \\ -.600 & .553 \\ .780 & .244 \end{bmatrix}$$

and the matrix of coordinates

$$A = \begin{bmatrix} -.246 & -.273 \\ -.824 & .189 \\ 1.070 & .084 \end{bmatrix},$$

which is plotted in figure 7, shows better separation along both dimensions.

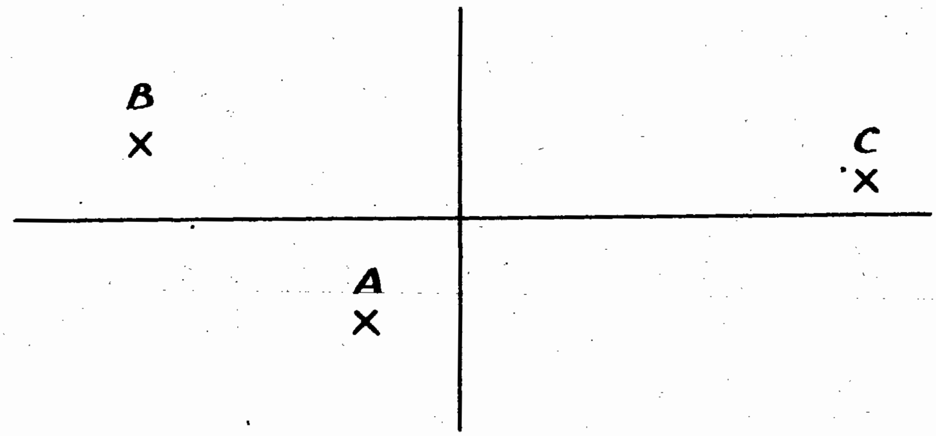

Fig. 7.--Configuration resulting from factor analysis after addition of constant to flows

Standby additive constant

Of course the addition of the constant c_1 does not theoretically guarantee that two positive eigenvalues will be generated by the power method, nor that satisfactory separation will be achieved in both dimensions. Thus a standby routine has been included in FLAC, which adds a second constant, this time directly to the dissimilarities, if a negative eigenvalue is generated in spite of the addition of c_1, or if the square root of the ratio of the two eigenvalues differs substantially from the ratio of the overall dimensions of the layout. If $\sqrt{\beta_2/\beta_1}$ is very small, then the facilities will lie close to the first axis. FLAC adds the second constant if the ratio $\beta_2/\beta_1 < .25S^2$, where S is the ratio of the shorter side of the layout to the longer. That is, FLAC adds the second constant if, in the unconstrained configuration, the facilities are spread out more than twice as much as they should be along the longer first dimension relative to the second, or if one of the eigenvalues is negative.

The objective of this standby additive constant c_s is to ensure that the facilities are well spread out along both dimensions and that the general shape of the configuration approximates that of the plant shape.

Messick and Abelson have shown that, in the presence of an additive constant such as c_s, the m^{th} eigenvalue of the B matrix can be expressed as

$$\beta m = X_m^T B X_m = X_m^T UX_m + c_s X_m^T WX_m + \frac{1}{2} c_s^2 X_m^T VX_m \quad (5)$$ [1]

where the dissimilarities d_{ij} which form the basis of the B matrix are expressed as $h_{ij} + c_s$, and

U is the B matrix which would result from $c_s = 0$,

W is the matrix of elements

$w_{ij} = (\frac{1}{n} \sum_j h_{ij} + \frac{1}{n} \sum_i h_{ij} - \frac{1}{n^2} \sum\sum_{ij} h_{ij} - h_{ij})$,

V consists of off-diagonal elements $-\frac{1}{n}$, and diagonal elements $(1-\frac{1}{n})$.

Since the sum of elements of X_m is zero by virtue of the fact that the origin is the centroid, and because of the structure of V,

$$X_m^T V = X_m^T .$$

Also, $X_m^T X_m = 1$ because the eigenvector was normalized, so that the last term in (5) reduces to $\frac{1}{2} c_s^2$.

---

[1] See Torgerson, Theory and Methods of Scaling, p. 275.

A satisfactory ratio of the first two eigenvalues occurs if

$$\sqrt{\beta_2/\beta_1} \approx S \qquad (6)$$

where S is the ratio of the shorter to the longer side of the layout space, assuming it is roughly rectangular. In FLAC, which assumes equal area facilities, S=(no. of rows - 1)/(no. of cols. - 1). If S > 1, S is inverted. If the number of rows equals the number of columns, then one unit is arbitrarily added to the denominator of the ratio S to ensure that FLAC does not attempt to calculate an additive constant which would equate the largest two eigenvalues.

Combining (5) and (6), it is now possible to solve for c_s in the equation

$$(X_2^T U X_2 + c_s X_2^T W X_2 + \tfrac{1}{2} c_s^2)(X_1^T U X_1 + c_s X_1^T W X_1 + \tfrac{1}{2} c_s^2)^{-1} = S^2.$$

Solving a quadratic equation, we obtain directly

$$c_s = \left\{ -S^2 X_1^T W X_1 + X_2^T W X_2 \pm \left[(S^2 X_1^T W X_1 - X_2^T W X_2)^2 - 2(S^2-1)(S^2 X_1^T U X_1 - X_2^T U X_2) \right]^{\frac{1}{2}} \right\} / (S^2-1). \qquad (7)$$

This method depends on an initial estimate of X_1 and X_2, the eigenvectors associated with the largest two eigenvalues. If, after the addition of the first constant, both eigenvalues are positive, then we already have good estimates for X_1 and X_2 since the eigenvectors are only minimally affected by the addition of a constant (see above, p. 64, f.2). However, if either of them is negative, then to ensure that the correct eigenvectors are used, the B matrix must be initially factored after the addition of a suitably large constant to the dissimilarities. This constant, initially set equal to the sum of the dissimilarities, ensures that the power method obtains the eigenvectors X_1 and X_2 associated with the algebraically largest eigenvalues. These initial eigenvectors are close enough to the final eigenvectors that an iterative approach is unnecessary.

It should be noted that since the standby additive constant c_s is added to the dissimilarities, rather than to the flows, this second transformation of the dissimilarities does alter the original problem. However this approach is justified on the basis that it is strictly an emergency recovery procedure for a potential but unlikely program hangup, and that the constant is dropped in the final stage of FLAC. Fortunately, the primary additive constant c_1 which, as we have seen, does not alter the original problem (above, p. 64), provides good enough results that the calculation of the standby constant was never required in all the test runs performed for this dissertation.

Our example problem is so small that the calculation of c_s is not typical. This is because there are very few facilities involved and the number of rows and columns in the layout is very small.

Nevertheless, suppose the final layout has two rows and two columns. Then

$$S = (\text{rows} - 1)/(\text{cols.} - 1) = 1$$

and because the number of rows equals the number of columns, the denominator is arbitrarily increased by one unit, so that $S=\frac{1}{2}$(in a larger problem, S would not be so sensitive to the unit addition to the denominator).

Continuing,

$$\beta_2/\beta_1 = .117/1.883$$

(see above, p.73), which is very slightly less than $.25S^2$, and FLAC would call for the calculation of the standby constant c_s[1].

Since β_1 and β_2 were both positive, we can use the initial values for X_1 and X_2 to calculate c_s directly from (7).

The U matrix is the same as the previously calculated B matrix, so that

$$U = \begin{bmatrix} .135 & .151 & -.286 \\ & .714 & -.865 \\ & & 1.151 \end{bmatrix},$$

$$W = \begin{bmatrix} .513 & -.049 & -.464 \\ & .869 & -.820 \\ & & 1.284 \end{bmatrix},$$

leading to

$$X_1^TUX_1 = 1.884, \quad X_2^TUX_2 = .117,$$

$$X_1^TWX_1 = 1.997, \quad X_2^TWX_2 = .671,$$

and $c_s = .880$.

---

[1] Since this did not occur in any actual runs of FLAC(the smallest problem involved five facilities), this unusual occurrence is attributed to the triviality of the example problem. Nevertheless it is convenient for illustrative purposes to show what would occur if the standby additive constant were required.

Adding c_s to the dissimilarity matrix and scaling the dissimilarities so that the average dissimilarity equals the average distance in the layout results in

$$D = \begin{bmatrix} 0 & .976 & 1.351 \\ & 0 & 1.673 \\ & & 0 \end{bmatrix}.$$

The new B matrix is then

$$B = \begin{bmatrix} .307 & -.008 & -.299 \\ & .631 & -.623 \\ & & .921 \end{bmatrix}.$$

The eigenvalues and eigenvectors are

$$\beta = \begin{bmatrix} 1.462 \\ .396 \end{bmatrix}, \qquad X = \begin{bmatrix} -.199 & -.792 \\ -.586 & .568 \\ .785 & .224 \end{bmatrix},$$

and the matrix of coordinates

$$A = \begin{bmatrix} -.241 & -.499 \\ -.709 & .358 \\ .950 & .141 \end{bmatrix}$$

is plotted in figure 8.

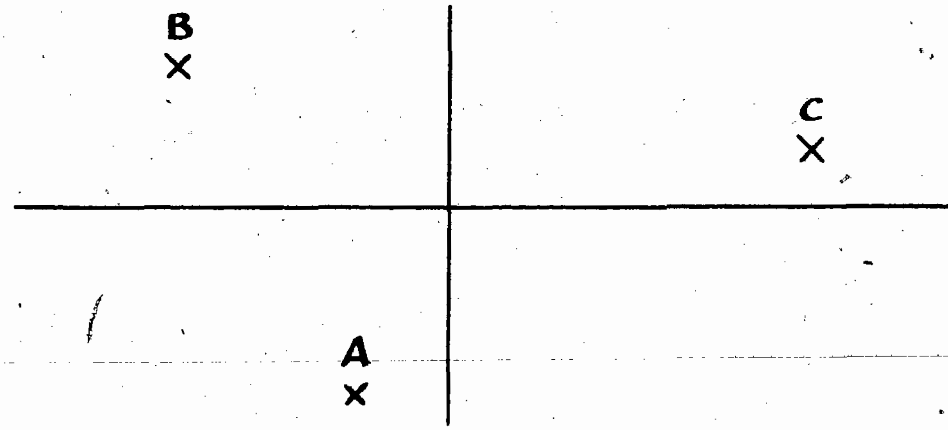

Fig. 8--Configuration resulting from factor analysis following adjustment by standby additive constant.

Note that the eigenvectors (matrix X) have not changed substantially from their previous values (see above, p. 73), but the ratio of the eigenvalues is now close to the desired ratio:

$$\sqrt{\beta_2/\beta_1} = \sqrt{.396/1.462} = .52 \approx S.$$

Need for further improvement of the unconstrained[1] configuration

While factor analysis will achieve a good configuration in two-dimensional space, especially when, as in our small example, only three facilities are to be located, some minor improvement can be made on the initial configuration when larger numbers of facilities are involved and the information contained in the third and subsequent eigenvalues and eigenvectors is ignored.

---

[1] The term "unconstrained" is used loosely to convey the idea of a configuration drawn in free space. As has been seen above, this configuration is somewhat constrained by manipulation of the dissimilarities to approximate the shape of the final layout.

It is convenient to demonstrate this by choosing an example in which we wish to draw a configuration along one dimension only. Suppose we have the adjusted dissimilarity matrix

$$D = \begin{bmatrix} 0 & 3 & 9 \\ & 0 & 10 \\ & & 0 \end{bmatrix}.$$

If we use factor analysis of the associated B matrix to plot the facilities in two-dimensional space, we obtain the configuration shown in figure 9.

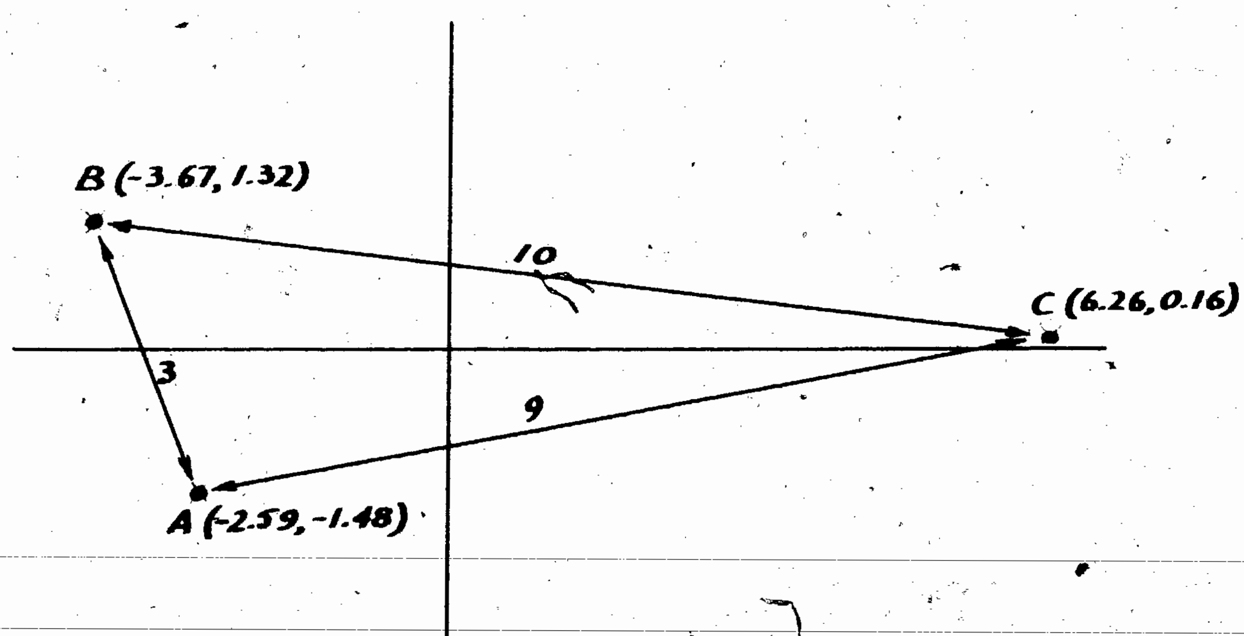

Fig. 9.--Two-dimensional configuration of a three-facility problem.

Figure 9 illustrates that the factor analytic approach draws the first axis through the points in such a way that it explains more of the variance between the points than the second axis, and so on. Since we have plotted three points in two dimensions, and the triangular inequality rule is not violated by the dissimilarity data, the distances between the points, or facilities, can be represented accurately.

Now suppose we wish to plot these points in one dimension only (this is analogous to plotting a more complex problem in only two dimensions). Ignoring the second eigenvalue and associated eigenvector, we obtain the configuration shown in figure 10.

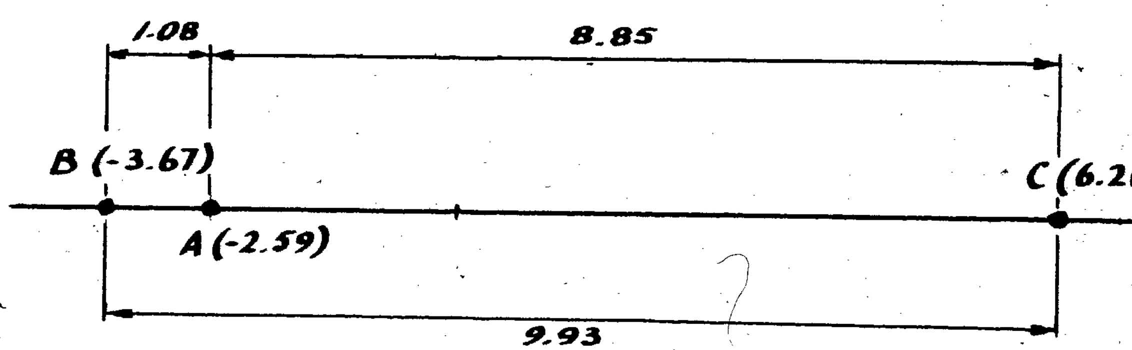

Fig.10--One-dimensional configuration of three facility problem

As might be expected, those distances which in the two-dimensional configuration were almost parallel to the first axis are much more accurately represented than the distance between A and B, which was almost at right angles to the first axis.

Thus, if the objective is to achieve distances which as far as possible are proportional to the dissimilarities (inversely proportional to the flows), it appears there is room for improvement in the configuration.

Stage 1: Improving the Unconstrained Configuration

Metric improvement algorithm

In his program for multidimensional scaling of sociometric data, TORSCA,[1] Young uses an iterative adjustment algorithm which moves each point to a new position at each iteration. The extent of the move depends on the discrepancies between the distances d_{ij} to other points and the target distances e_{ij} which Young calls disparities. In TORSCA, the disparities are also changed from time to time in an effort to achieve better ranking of the distances (measures are ordinal).

In facility layout, an argument can similarly be made for the use of a non-metric algorithm. The rationale behind such an argument would be based on the same theorem used to justify the

1 F.W. Young, "A FORTRAN IV Program."

MAT algorithm[1]: "The sum of pairwise products of two sequences of real numbers is minimized if one sequence is arranged in non-increasing order and the other is arranged in non-decreasing order." However, if one is dealing with interval measurements, then much information is lost by using a non-metric approach. For instance if, as in most problems, it is impossible to achieve this perfect ranking of distances, then using the non-metric approach one might increase the distance between two facilities with very heavy interaction in order to correctly rank the distances between a few very lightly loaded facilities.

The adjustment algorithm in FLAC is similar to that used by Young with two important exceptions:

(1) In the FLAC algorithm each adjustment is based on the weighted (by flow) discrepancies between the distances and their targets, and

(2) the target distances are inversely proportional to the associated flows.

For each coordinate of each facility, a correction is calculated based on the average discrepancy between the current distances to other facilities and the target distances:

$$c_{ia} = \frac{\alpha}{(n-1)\sum\sum_{ij} f_{ij}} \sum_{j\neq i} \left[\frac{X_{ja} - X_{ia}}{d_{ij}} \left(f_{ij} d_{ij} - \frac{\sum\sum_{ij} f_{ij} d_{ij}}{n(n-1)/2} \right) \right] \quad (8)$$

where:

---

[1] Edwards et al., "Modular Allocation Technique".

c_{ia} is the correction to be added to the coordinate X_{ia} of facility i in the a^{th} dimension,

α is a factor used to control the rate of improvement,

d_{ij} is the current distance between facilities i and j.

In this iterative algorithm, f_{ij} includes the initial additive constant c_1 calculated before performing the factor analysis earlier in stage 1. This is done to avoid the tendency for distances involving zero flow to become indefinitely larger, and is justified on the basis that the original problem is not altered by the addition of c_1 as discussed above, p. 64 .

Equation (8) is developed from the following individual correction formula for each distance in the configuration:

$$c_{ij} = \frac{\alpha f_{ij}}{\sum\sum_{ij} f_{ij}} (d_{ij} - e_{ij}) \tag{9}$$

where e_{ij} is the target distance between facilities i and j.

The weighting factor $f_{ij}/\sum\sum_{ij} f_{ij}$ is not theoretically necessary but is included so that the facilities which interact heavily with others approach their target distances with those facilities more rapidly than do others. Thus residual error is likely to involve lightly loaded facilities. The quotient $\sum_i \sum_j f_{ij}$ of the weighting factor is included so that the factor cannot exceed unity.

To achieve the inverse relationship between flows and distances and to maintain the average distance constant, the target distances e_{ij} in (9) should be chosen so that

$$e_{ij}f_{ij} = e_{kl}f_{kl} = R \quad \text{for all } i,j,k,l \tag{10}$$

and

$$\sum_i \sum_j (d_{ij} + c_{ij}) = \sum_i \sum_j d_{ij}. \tag{11}$$

Equation (11) implies that

$$\sum_i \sum_j c_{ij} = 0 \tag{12}$$

and substituting from (9) and (10) in (12) we obtain

$$\sum_i \sum_j f_{ij}d_{ij} = \sum_i \sum_j f_{ij}e_{ij} = NR \tag{13}$$

where $N = n(n-1)/2$.

Thus

$$R = \sum_i \sum_j f_{ij}d_{ij}/N \tag{14}$$

and

$$e_{ij} = \frac{\sum_i \sum_j f_{ij}d_{ij}}{N\ f_{ij}}. \tag{15}$$

Substituting in (9) from (15), we obtain the correction to be applied to d_{ij} in terms of the distances and flows:

$$c_{ij} = \frac{\alpha}{\sum_i \sum_j f_{ij}} \cdot \left[f_{ij}d_{ij} - \frac{\sum_i \sum_j f_{ij}d_{ij}}{n(n-1)/2} \right] \tag{16}$$

The appropriate correction along the a axis is then

$$c_{ija} = \frac{x_{ja} - x_{ia}}{d_{ij}} \quad c_{ij} \tag{17}$$

and averaging (17) over the n-1 distances from facility i to the other facilities j, we obtain (8).

From (17) it is clear that a positive c_{ij} leads to a reduction in the distance between facilities i and j.

To see why the adjustment tends to lead to a configuration in which distances are inversely proportional to flows, consider the effect of applying it in two stages. Then (16) could be written

$$c_{ij} = \frac{\alpha f_{ij} d_{ij}}{\sum_i \sum_j f_{ij}} - \frac{\alpha \sum_i \sum_j f_{ij} d_{ij}}{\sum_i \sum_j f_{ij} \; n(n-1)/2} \tag{18}$$

in which the right-hand term is a constant.

Thus a correction can be viewed as first reducing each distance by a factor proportional to the flow along that distance, and then increasing every distance by a constant amount. If all distances were already inversely proportional to their associated flows, each distance would be decreased and then increased by the same constant amount, resulting in no change.

Choice of convergence factor

The optimum size of the factor α is not easy to determine. Initial experiments suggested that values of α as large as 3 and as small as 0.1 could achieve visible improvements to a configuration. If α is chosen too small, then convergence is slow. On the other hand if α is too large then overcorrection occurs and the configuration is not improved. Initial experiments also suggested that the sensitivity of the configuration to the adjustment decreases as the configuration approaches the ideal, so that larger values of α can be tolerated. It was therefore decided to build into FLAC a routine which begins with a maximum α factor, successively reducing it by halving it when no improvement results from an adjustment, until a minimum value of α is reached. In addition, if an improvement occurs in six successive iterations, the factor is doubled.

The maximum and minimum α used in all runs, including those upon which computer running time estimates are based, were 50 and .01, respectively. However, examination of the results reveals that maximum and minimum values of 4 and 0.5 would have been adequate for all runs, reducing the computation time slightly.

Maintaining scale of configuration

To ensure that the scale of the configuration is not gradually eroded by truncation errors, the coordinates are scaled at each iteration in such a way that the average distance from the centroid to each facility remains constant and equal to that in the constrained layout.[1]

Stopping criterion

Since the objective of stage 1 is to achieve as much as possible a configuration in which distances are inversely

---

[1] Some computer running time could be saved by performing this scaling less frequently.

proportional to flows, a reasonable choice for a stopping criterion is the variance of the $d_{ij}f_{ij}$.

To facilitate comparison of solution quality from problem to problem, the coefficient of variation of the $d_{ij}f_{ij}$, which can also serve as an index of fit, was chosen as the stopping criterion for this improvement algorithm in the first stage of FLAC. The coefficient of variation also has the advantage that it is unaffected by scaling of the configuration, and is easily calculated from

$$CV = \left[N \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} f_{ij}^{2} d_{ij}^{2} \left(\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} f_{ij} d_{ij} \right)^{-2} - 1 \right]^{\frac{1}{2}}$$

where $N = n(n-1)/2$.

Figures 11 and 12 show respectively the configuration produced by the factor analysis and iterative adjustment routines of stage 1 of FLAC in the eight facility problem of Nugent et al. For convenience, flows have been represented schematically as follows:

Flow	No. of lines
0	0
1-3	1
4-6	2
10	3

The complete flow matrix for this problem is shown in figure 13.

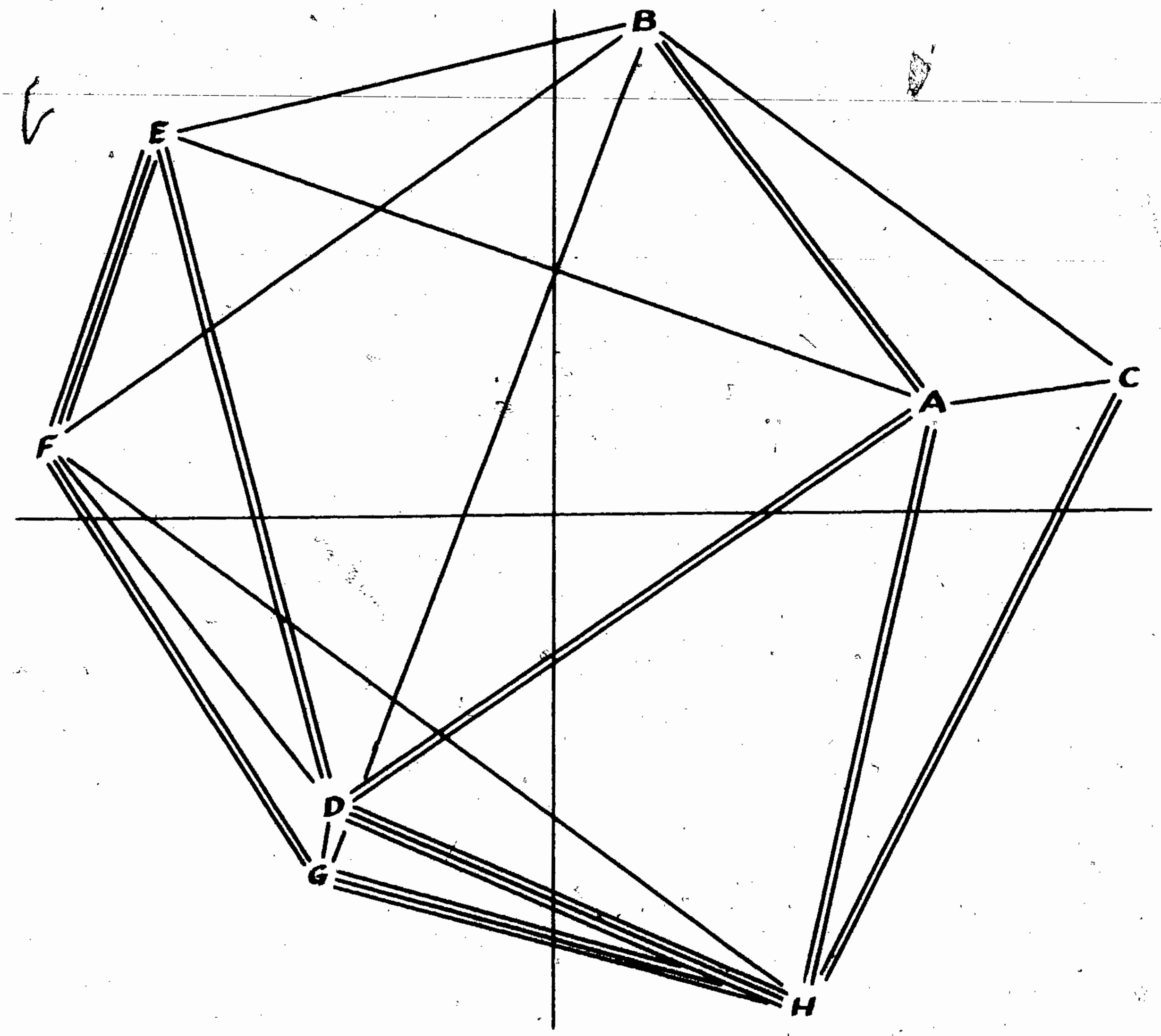

Fig. 11.--Configuration resulting from initial factor analysis in stage 1 of FLAC, using flow data from the eight facility problem of Nugent et al.

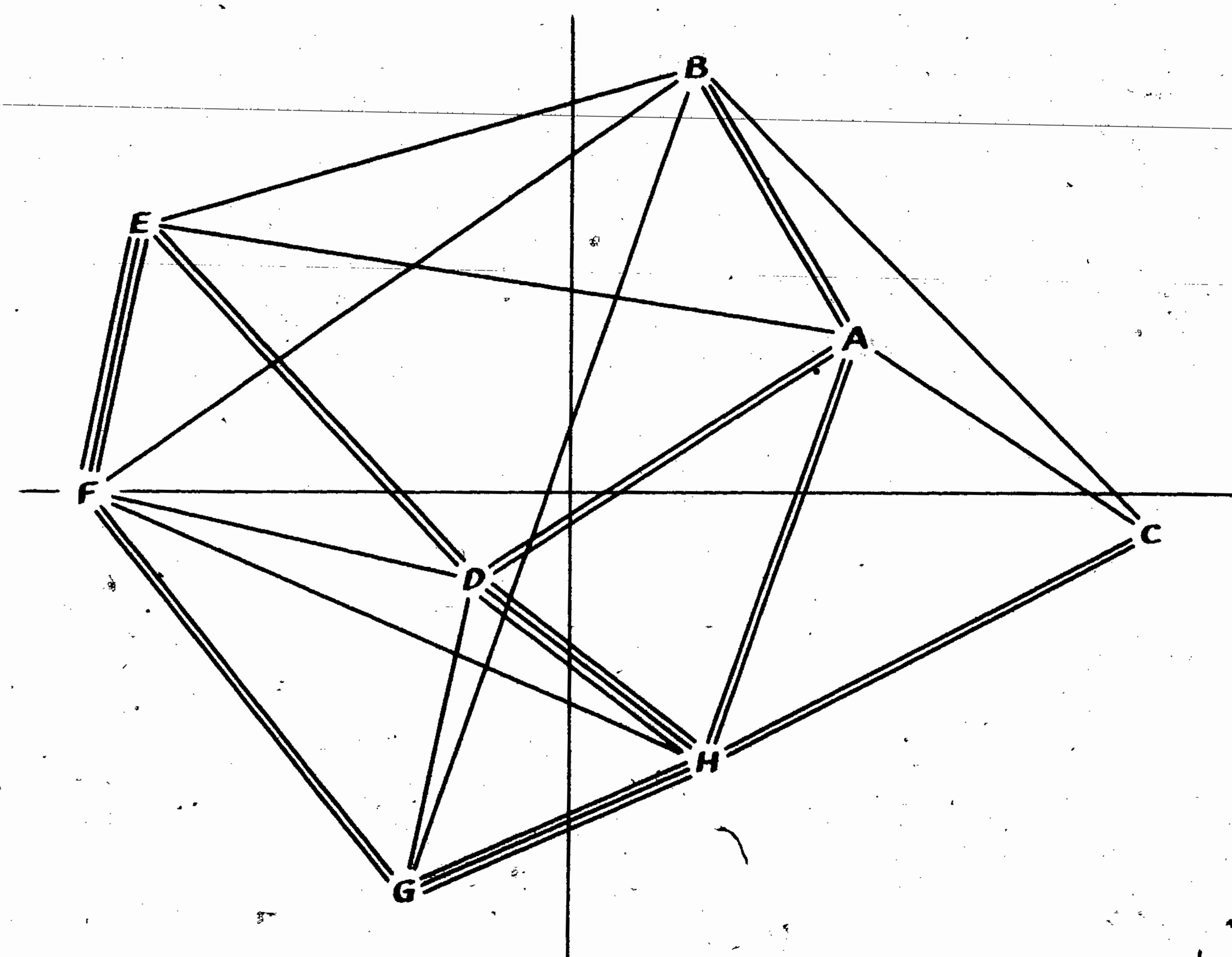

Fig. 12.--Configuration of the eight facilities after iterative adjustment by the metric improvement algorithm.

	A	B	C	D	E	F	G	H
A	0	5	2	4	1	0	0	6
B	5	0	3	0	2	2	2	0
C	2	3	0	0	0	0	0	5
D	4	0	0	0	5	2	2	10
E	1	2	0	5	0	10	0	0
F	0	2	0	2	10	0	5	1
G	0	2	0	2	0	5	0	10
H	6	0	5	10	0	1	10	0

Fig. 13.--Flow matrix for eight-facility problem generated by Nugent _et al_.

Figure 12 shows a visible improvement resulting from the iterative adjustment routine, both in the relative reduction of distances with heavy flows, and in the better separation between facilities such as D and G which have light interaction. The coefficient of variation of the $f_{ij}d_{ij}$ is reduced from 48% after factor analysis to 36% after iterative adjustment.

Stage 2: Fitting the Facilities into the Constrained Layout Space

If, in fitting the facilities into the constrained plant shape, we attempt to use materials handling cost directly as a criterion, we have to consider all the interactions between the facilities as they are placed in the layout, and the problem is as complex as the original. However, if we use as a criterion only a function of the distance each facility has to move from its position on the unconstrained map to its final location in the superimposed layout, then the original quadratic assignment problem is reduced to a simple assignment problem:

$$\text{minimize} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\text{subject to} \quad \sum_{i=1}^{n} x_{ij} = 1, \qquad j = 1,2,\ldots,n$$

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad i = 1,2,\ldots,n$$

where x_{ij} = 1 if facility i is assigned to location j, 0 otherwise;

c_{ij} is a function of the distance facility i must be moved to place it at location j.

Since it is not yet possible to use layout cost as a criterion, other criteria than linear functions of distance can be entertained, such as for instance the sum of squared deviations from the unconstrained configuration. Such a criterion would not however yield good results since it would place greater penalties on large movements and therefore exaggerate the importance of correctly locating outlying facilities which are likely to have only light interaction with the other facilities. In general, the final cost of a layout is clearly more sensitive to the location of heavily loaded facilities than lightly loaded ones. Thus the criterion chosen for stage 2 in FLAC is the distance each facility must be moved from its position in the unconstrained configuration, weighted by the sum of all flows passing through that facility. The choice of this criterion could occasionally lead to a lightly loaded outlying facility being located between two heavily loaded central facilities. To counteract any tendency for this occurrence in FLAC, the scale of the unconstrained configuration may be reduced by a factor specified by the user before solving the assignment problem.[1]

[1] A reduction factor of .9 was chosen arbitrarily and used in all runs performed for this dissertation.

Rotation

Just as statisticians have to rotate the axes after performing factor analysis on a correlation matrix, so is it necessary to consider rotations of the configuration provided by stage 1 of FLAC, relative to the layout space. This is particularly the case since Euclidean distance measures, insensitive to rotation, were used in stage 1, whereas the final layout cost is based on rectilinear distances between facilities.

In FLAC, the configuration resulting from stage 1 is rotated a number of times through an angle specified by the user. The assignment algorithm is applied to the result of each rotation and the best assignment is retained. At this stage, for the first time, materials handling cost $\sum_i \sum_j f_{ij} d_{ij}$ can be used as the criterion, where the d_{ij} are rectilinear

Initial experiments indicated that six rotations of 30 degrees (180 degrees total) provided good results, even with an irregular layout shape, and these numbers were used in all tests carried out for this dissertation.

Primal-dual algorithm

Hatch found primal-dual algorithms were superior to primal-simplex methods for transportation problems, especially in the

the presence of degeneracy.[1] Thus a subprogram was written for FLAC based on Ford and Fulkerson's adaptation of the primal-dual algorithm for transportation problems, as described by Hadley.[2]

Applying this algorithm to the results of stage 1 in the Nugent et al. eight facility problem leads to the assignment shown in figure 14. Total cost of this assignment is 113,

E	D	B	A
F	G	H	C

Fig. 14.--Layout resulting from application of assignment algorithm to stage 1 results of Nugent et al. eight facility problem

which is competitive with the CRAFT results reported by Nugent et al. In this case further assignments using rotations of the stage 1 configuration led to no improvement over the initial assignment.

---

[1] Richard S. Hatch, "Bench Marks Comparing Transportation Codes based on Primal-Simplex and Primal-Dual Algorithms," Operations Research, Vol. 23, No. 6, Nov. - Dec. 1975, pp. 1167-1172.

[2] G. Hadley, Linear Programming, Reading, Mass., Addison-Wesley, 1962, pp. 351-367.

Stage 3: CRAFT-type Exchange

While the first two stages of FLAC may provide good solutions to the equal area facility layout problem, it must be remembered that the purpose of those stages is primarily to avoid poor placement of lines or clusters of facilities relative to each other. The combination of algorithms used is more likely to succeed in this respect than in the optimum placement of each individual facility relative to others, since, even if all facilities are well placed in the unconstrained configuration, it is likely that some individual placements will be upset during the stage 2 assignment. Also, since real total cost cannot be considered until after an assignment has been made, it is likely that minor adjustments to the layout could result in an improvement in total cost.

Thus stage 3 of the FLAC algorithm incorporates the well known two-way exchange heuristic of CRAFT.

It is hypothesized that the layout resulting from an application of the CRAFT-type heuristic improvement to the layout produced in stage 2 will be statistically better than layouts produced by CRAFT from random initial layouts, especially on larger problems and problems with line dominance, where lines or clusters of facilities may be poorly located relative to each other.

The basic two-way exchange algorithm of CRAFT has been shown to be an effective improvement algorithm and is therefore a logical choice for this final stage of FLAC. It also allows comparison with previously recorded results in some problems. Since the exchange heuristic is to be applied to good layouts resulting from stage 2 of FLAC, it is expected that the running time of the CRAFT portion would be significantly reduced from that of CRAFT runs from random initial layouts. Nevertheless there exist several variations on CRAFT, one of which, FRAT, provides results competitive with CRAFT in a shorter time.[1] For this reason, the FRAT algorithm suggested by Khalil[2] is used in stage 3 of FLAC to save computation time.

In any case, the final equal area layout produced by either stage 2 or stage 3 of FLAC must be subjected to further manipulation to take into account unequal areas. Such manipulation is compatible with any two-way exchange algorithm and has been dealt with by Buffa _et al_. in CRAFT[3] and by Ritzman,[4] who suggests several methods which need not be included in the computer algorithm. The handling of unequal areas, which also clouds comparisons with previously recorded results is not

---

[1] As stated in chapter I (above, p. 18) there is no theoretical reason why FRAT should provide better or worse solutions than the basic CRAFT algorithm, since it performs only two-way exchanges and terminates with CRAFT.

[2] Khalil, "Facilities Relative Allocation Technique."

[3] E.S. Buffa, G.C. Armour, and T.E. Vollmann, "Allocating Facilities with CRAFT," Harvard Business Review, Vol. 42, No.2, 1964, pp. 136-159.

[4] L.P. Ritzman, "The Efficiency of Computer Algorithms for Plant Layout," Unpublished doctoral dissertation, Michigan State University, 1968, p. 22.

included in this dissertation but can subsequently be appended to stage 3 of FLAC.

In the eight facility problem of Nugent <u>et al</u>. , one exchange was performed by the FRAT algorithm of stage 3, yielding the final layout shown in figure 15. Total cost of this layout is 107, which is optimal.

E	D	A	B
F	G	H	C

Fig. 15.--Final layout resulting from application of stage 3 exchange algorithm to the stage 2 layout in the Nugent <u>et al</u>. eight facility problem

CHAPTER IV

EXPERIMENTAL RESULTS AND CONCLUSIONS

Solution Quality

As stated previously, it is the hypothesis that the three stages of FLAC will provide results either competitive with the basic CRAFT heuristic in small problems and problems with low flow dominance, or better than CRAFT alone under conditions of high line dominance and flow dominance in larger problems.

FLAC results

Although it is not intended that FLAC ever be used except in its entirety, the intermediate results of stage 2 will be reported since they are of theoretical interest. These intermediate results will hereinafter be reported under the heading $FLAC_i$. The final results of all three stages are reported as FLAC results.

CRAFT/FRAT results

It was the initial intention in these experiments to use the CRAFT two-way exchange algorithm as the standard of comparison for solution quality for reasons stated earlier in chapter I. However it soon became apparent that the computation time required to obtain five CRAFT solutions to each of the experimental problems would be excessive. Thus the standard for some problems, especially the larger ones, is based on five solutions from random initial layouts using the faster FRAT algorithm. This should cause no theoretical difficulty since, as discussed previously in the section comparing current computer algorithms (see above, p. 12), there is no reason to believe FRAT results should be different in quality from CRAFT results. Thus FRAT results are reported along with CRAFT results under the heading CRAFT/FRAT.

High line dominance

To test the hypothesis that FLAC performs better than CRAFT/FRAT on problems with high line dominance, a set of seven problems, ranging in size from twelve to forty facilities, was generated using the interactive LINEMIX function, written in the APL language and listed in appendix C. The length of the lines, or job routes, and the number of different lines in any one problem were chosen in such a way as to maintain flow-dominance at an intermediate level of approximately 140 percent. Line length was maintained at approximately n/2, so that these are rather extreme cases of line dominance. However, there is a sufficient number of lines superimposed upon each other in each problem that this line dominance is not easily determined from an examination of the flow matrices, which are reproduced in appendix D. The number of lines superimposed in each problem was varied by trial and error to achieve a flow dominance in the region of 140 percent, and ranged from n/2 to 2n/3.

The layout costs achieved by FLAC and five CRAFT/FRAT runs from random initial solutions are shown in table 8. In all seven problems, FLAC produced a better layout than the median CRAFT/FRAT layout. The probability that this would occur by chance alone given algorithms of equal capability is less than .01, so that these results are highly significant.

FLAC even appears to be superior to the best of five CRAFT/FRAT runs, having produced better layouts in all but the smallest problem, but the results are statistically significant only at $\alpha = .0625$.

The improvement in efficiency of the FLAC solutions over the best of five CRAFT/FRAT solutions ranged from minus three percentage points in the smallest problem to plus six percentage points and, when compared with the median CRAFT/FRAT results, the improvement was four to seven percentage points.

Table 8 also shows that there is no significant difference between the intermediate results obtained in stage 2 of FLAC ($FLAC_i$) and the best of five CRAFT/FRAT results.

Varying line dominance

In the second experiment, line dominance is varied while holding other factors (problem size, flow dominance) virtually constant. All five problems are of the intermediate size of

TABLE 8

PERFORMANCE OF FLAC AND CRAFT/FRAT IN PROBLEMS
WITH HIGH LINE DOMINANCE AND INTERMEDIATE FLOW DOMINANCE

No. of facilities	Line dominance			Layout cost (efficiency)			
	Route length	No. of routes	Flow dom. (percent)	$FLAC_i$	FLAC	CRAFT/ FRAT median	CRAFT/ FRAT best of 5
12	6	7	147	205+ (72)	203+ (73)	212 (67)	199 (76)
15	7	10	146	383 (66)	369++ (71)	379 (67)	378 (67)
20	10	10	141	593++ (63)	593++ (63)	617 (58)	601 (61)
20	10	12	143	711++ (59)	708++ (60)	739 (54)	729 (56)
30	15	20	137	2365+ (51)	2331++ (58)	2409 (48)	2352 (51)
30	15	20	138	2420+ (47)	2315++ (53)	2426 (46)	2415 (47)
40	20	25	134	5095++ (43)	5045++ (45)	5152 (42)	5113 (43)

\+ indicates layout better than median CRAFT/FRAT layout
++ indicates layout better than best of five CRAFT/FRAT layouts

twenty facilities.[1] Flow dominance was again maintained at approximately 140 percent by holding the product of line length and number of lines constant at 120. One problem with no line dominance was generated by holding line length at unity and superimposing 120 lines. This is equivalent to choosing 120 random flows. The remaining four problems had line lengths ranging from five to fifteen legs, as shown in table 9.

In these problems both FLAC and $FLAC_i$ performed well, producing a better layout than the best of five CRAFT/FRAT layouts in all except the problem with no line dominance. All of the FLAC and $FLAC_i$ results were better than the median CRAFT/FRAT results.

The increase in efficiency of the FLAC solutions measured against the best of five CRAFT/FRAT solutions ranged from minus one percentage point in the problem with no line dominance to four or five percentage points in all the problems with line dominance. Measured against the median CRAFT/FRAT solutions, improvements ranged from six to eight percentage points.

These results appear to suggest that FLAC's advantage over CRAFT/FRAT increases with line dominance. It should be noted, however, that the efficiency of the best CRAFT/FRAT layout for the problem with no line dominance is so much better than that of the median CRAFT/FRAT solution (an increase of nine percentage

---

[1] These are the same problems used to test the effect of line dominance on the efficiency of CRAFT in chapter II. The flow matrices are reproduced in appendix B.

TABLE 9

THE EFFECT OF VARYING LINE DOMINANCE ON THE RELATIVE PERFORMANCES OF FLAC AND CRAFT/FRAT IN PROBLEMS OF INTERMEDIATE SIZE[a] AND INTERMEDIATE FLOW DOMINANCE

Line dominance			Layout cost (efficiency)			
Route length	No. of routes	Flow dom. (percent)	$FLAC_i$	FLAC	CRAFT/ FRAT median	CRAFT/ FRAT best of 5
1	120	143	739+ (64)	715+ (69)	761 (61)	710 (70)
5	24	140	682++ (64)	667++ (67)	697 (60)	687 (62)
8	15	142	852++ (59)	842++ (61)	881 (55)	870 (56)
10	12	143	711++ (59)	708++ (60)	739 (54)	729 (56)
15	8	143	607++ (58)	600++ (59)	635 (51)	622 (54)

\+ indicates layout better than CRAFT/FRAT median layout.
++ indicates layout better than best of five CRAFT/FRAT layouts.
a all problems are of the same twenty-facility size.

points) as to suggest that it may be an outlier, statistically speaking.[1]

No line dominance

Since the results of the preceding tests do not show conclusively that the superiority of FLAC depends on the presence of line dominance, it is of interest to perform further comparisons of FLAC and CRAFT/FRAT on problems with no line dominance.

Five more problems were generated by choosing flows randomly from a Poisson distribution as was done in the case of the problems used in chapter II to test the effect of flow dominance on solution efficiency. The parameter λ was chosen in such a way as to hold flow dominance at the same intermediate level of approximately 140 percent. Problem size was varied from twelve to thirty facilities.[2] The flow matrices are reproduced in appendix E.

---

[1] Generally, the experiments performed in the course of this dissertation indicate that the spread in efficiency between median and best CRAFT/FRAT solutions decreases with problem size and increases with flow dominance. The possibility that the unusually large spread in efficiency in this particular problem is caused by a lack of line dominance is not supported by results on other problems reported in this dissertation. In nine other twenty-facility problems, three of which lacked line dominance, the spread in efficiency was never greater than three percentage points.

[2] One of these five problems, the thirty facility problem, had already been generated and used in chapter II to test the effect of flow dominance on solution efficiency.

Unexpectedly, FLAC performed better than ever, producing the best layout in all five problems as shown in table 10, and exceeding the efficiency of the best CRAFT/FRAT solution by from three to nine percentage points. Even the intermediate layouts of FLAC surpassed the best of five CRAFT/FRAT layouts in all the problems. Based on this experiement alone, one would conclude that FLAC and $FLAC_i$ produce significantly better (α = .03125) layouts than the best of five CRAFT/FRAT runs in problems with no line dominance, flows randomly chosen from a Poisson distribution, flow dominance at an intermediate level.

Considering all the experiments undertaken up to this point, with flow dominance held constant at an intermediate level, FLAC has obtained better results than the best of five CRAFT/FRAT runs in fourteen out of sixteen[1] problems with and without line dominance. The probability of this occurrence by chance alone if there is no difference between FLAC and the best of five CRAFT/FRAT runs is .00209. Similarly, FLAC produced intermediate layouts ($FLAC_i$) better than the median CRAFT/FRAT layout in fifteen of the sixteen problems tested to this point, and better than the best of five CRAFT/FRAT layouts in eleven of the sixteen problems, so that the intermediate $FLAC_i$ results appear to be better than individual CRAFT/FRAT

[1] Tables 8, 9 and 10 display 17 sets of results, but one twenty-facility problem fitted the requirements for two experiments and the results appear in both tables 8 and 9.

TABLE 10

PERFORMANCE OF FLAC AND CRAFT/FRAT
IN PROBLEMS WITH NO LINE DOMINANCE[a]
AND INTERMEDIATE FLOW DOMINANCE

No. of facilities	Flow dom. (percent)	$FLAC_i$	Layout cost (efficiency) FLAC	CRAFT/FRAT median	CRAFT/FRAT best of 5
12	137	52++ (80)	51++ (83)	54 (74)	54 (74)
15	137	108++ (64)	104++ (70)	113 (56)	110 (61)
20	141	193++ (64)	185++ (70)	197 (61)	196 (62)
20	136	197++ (63)	194++ (65)	201 (60)	198 (62)
30	144	494++ (55)	488++ (57)	517 (49)	508 (51)

++ indicates layout better than best of five CRAFT/FRAT
a Flows randomly selected from Poisson distribution

results at a significance level of .00026, and competitive with the best of five CRAFT/FRAT runs.

Varying flow dominance

It is now of interest to test the effect of flow dominance on the relative performance of FLAC and CRAFT/FRAT. To do this, the five thirty-facility problems originally referred to in chapter II were used. These problems, generated by drawing flows randomly from a Poisson distribution, have flow dominances varying from 95 to 282 percent and include the thirty-facility problem used in the preceding test.

Again, as shown in table 11, FLAC performed well, achieving the best layout in four of the five problems. FLAC's only poor performance was in the problem with lowest flow dominance, suggesting that FLAC may not be effective under conditions of low flow dominance.

Low flow dominance

To test this possibility further, a FLAC run was made on each of the problems previously used by Nugent *et al*.[1] These problems all have a very low flow dominance, ranging from 104 to 129 percent.

Table 12 shows that FLAC produced layouts equal to or better than the best of five CRAFT results reported by Nugent *et al*. in all except the fifteen-facility problem where the

---

1 Nugent *et al*., "Experimental Comparison."

TABLE 11

EFFECT OF VARYING FLOW DOMINANCE ON THE RELATIVE PERFORMANCES OF FLAC AND CRAFT/FRAT IN 30-FACILITY PROBLEMS[a]

	Layout cost (efficiency)			
Flow dom. (percent)	$FLAC_i$	FLAC	CRAFT/FRAT median	CRAFT/FRAT best of 5
96	1359 (46)	1349 (47)	1348 (48)	1343 (48)
115	869+ (50)	849++ (55)	880 (48)	861 (52)
144	494++ (55)	488++ (57)	517 (49)	508 (51)
199	252++ (64)	249++ (66)	268 (58)	255 (63)
282	93++ (77)	89++ (80)	106 (68)	98 (74)

+ indicates layout better than CRAFT/FRAT median.
++ indicates layout better than CRAFT/FRAT best of five
a Flows randomly selected from Poisson distribution

TABLE 12

RELATIVE PERFORMANCE OF FLAC AND CRAFT IN EIGHT PROBLEMS WITH LOW FLOW DOMINANCE PREVIOUSLY USED BY NUGENT ET AL.[a]

No. of facilities	Flow dom. (percent)	Layout cost (efficiency)			
		$FLAC_i$	FLAC	CRAFT median	CRAFT best of 5
5	108	25+= (100)	25+= (100)	29 (61)	25 (100)
6	129	43== (89)	43== (89)	43 (89)	43 (89)
7	111	74+= (84)	74+= (84)	79 (69)	74 (84)
8	128	113 (67)	107+= (77)	110 (72)	107 (77)
12	117	300 (67)	289+= (74)	295 (70)	289 (74)
15	106	588+ (66)	585+ (67)	591 (65)	583 (67)
20	104	1310++ (58)	1303++ (59)	1334 (55)	1324 (56)
30	112	3154+ (51)	3122++ (53)	3169 (50)	3148 (51)

a Nugent et al., "Experimental Comparison."

+= a plus sign, or equals symbol, immediately following a FLAC or $FLAC_i$ cost indicates a layout better than, or equal to, the median CRAFT layout. A second symbol similarly rates the layout relative to the best of five CRAFT layouts.

FLAC result is slightly (0.3 percent) inferior, so that FLAC is clearly competitive with the best of five CRAFT runs in problems with low flow dominance. However the results of FLAC are not spectacular in these problems, bettering the CRAFT best of five in only the two largest problems, and FRAT has recorded a slightly better solution for the twenty-facility problem.[1] When compared with individual CRAFT runs, FLAC fares better, recording a better solution than the median CRAFT solution in all but one problem in which it tied the median CRAFT result. The statistical significance of this performance, after dropping the tie, is .00781.

Thus the results recorded on these problems of Nugent et al. indicate that in the presence of low flow dominance FLAC layouts are better than individual CRAFT layouts and competitive with the

---

[1] Khalil, "Facilities Relative Allocation Technique." As pointed out in chapter I, there is no significant difference between the quality of FRAT and CRAFT solutions, nor is there any theoretical basis for a difference. Inclusion of Khalil's FRAT results here in addition to Nugent's CRAFT results would merely tend to lead to the conclusion that FLAC is competitive with the best of ten CRAFT/FRAT solutions, rather than only the best of five. Better results have also been recorded on the largest problem by Nugent et al., "Experimental Comparison," using Biased Sampling, and by Hitchings and Cottam, "Efficient Heuristic Procedure," using Terminal Sampling Procedure (TSP), but these results are achieved at the expense of a larger sample as discussed in chapter I. In the case of Biased Sampling, the sample size was ten in each of five runs per problem. Hitchings and Cottam did not state the size of their sample which varies with the number of intermediate tie solutions encountered, but the TSP run time suggests several ties were encountered.

best of five CRAFT layouts. An examination of the intermediate $FLAC_i$ results indicates that they are competitive with CRAFT.

Table 12 also seems to support the expectation that FLAC would be less competitive on small problems than large ones, since FLAC was unable to outperform the best of five CRAFT runs on any of the problems with less than twelve facilities.

Additional test on small problems

An independent test using four very small problems generated by drawing random Poisson flows adds further support to this conclusion. In that test, as shown in table 13, FLAC results were not significantly better than the median CRAFT/FRAT solutions even though the intermediate $FLAC_i$ results were competitive with the median CRAFT/FRAT solutions. Flow dominance in these problems was at an intermediate level.

High flow dominance (Grover's problems)

Comparing FLAC with Grover's results of CRAFT runs on his set of eight problems[1] with high flow dominance, FLAC provided better results than the best of five CRAFT runs on only the problems with twelve or more facilities, further supporting the conclusion that FLAC provides better solutions than CRAFT in large problems and competitive solutions in small problems (see table 14).

---

[1] Grover, "Plant Layout Algorithms."

TABLE 13

RELATIVE PERFORMANCE OF FLAC AND CRAFT/FRAT ON VERY SMALL PROBLEMS[a]

No. of facilities	Flow dom. (percent)	Layout cost (efficiency)			
		$FLAC_i$	FLAC	CRAFT/ FRAT median	CRAFT/ FRAT best of 5
5	146	5= (58)	5= (58)	5 (58)	4 (100)
6	152	8== (79)	8== (79)	8 (79)	8 (79)
7	139	14= (63)	14= (63)	14 (63)	13 (75)
8	138	32= (71)	32= (71)	32 (71)	30 (82)

= indicates layout equal to median CRAFT/FRAT layout.
== indicates layout equal to best of five CRAFT/FRAT layouts.
a Flows randomly selected from Poisson distribution, intermediate flow dominance.

TABLE 14

RELATIVE PERFORMANCE OF FLAC AND CRAFT IN EIGHT PROBLEMS WITH HIGH FLOW DOMINANCE PREVIOUSLY USED BY GROVER[a]

No. of facilities	Flow dom. (%)	Layout cost (efficiency)			
		$FLAC_i$	FLAC	CRAFT median	CRAFT best of 5
5	269	64.02== (94)	64.02== (94)	64.02 (94)	64.02 (94)
6	263	80.32== (95)	80.32== (95)	80.32 (95)	80.32 (95)
7	199	163.28+= (85)	163.28+= (85)	165.16[b] (83)	163.28[b] (85)
8	213	235.96 (76)	220.86 (85)	215.86 (88)	214.76 (89)
10	233	298.74 (78)	296.58 (79)	283.34 (84)	279.02 (86)
12	256	314.10++ (89)	314.10++ (89)	340.60 (80)	331.58 (83)
15	248	557.60+ (81)	511.40++ (89)	560.78 (80)	531.14 (85)
20	254	1154.78++ (79)	1127.40++ (81)	1225.70 (74)	1186.66 (77)

a Grover, "Plant Layout Algorithms."

b Based on additional CRAFT runs of seven-facility problem, substituted for Grover's results which contained errors.

+= a plus sign, or equals symbol, immediately following a FLAC or $FLAC_i$ cost indicates a layout better than, or equal to, the median CRAFT layout. A second symbol similarly rates the layout relative to the best of five CRAFT layouts.

Overall performance of FLAC

In summary, these experiments have shown that FLAC outperforms the best of five CRAFT/FRAT runs in large problems with intermediate or high flow dominance both in the presence and absence of line dominance, and is competitive with the best of five CRAFT/FRAT runs in smaller problems and problems with low flow dominance. Considering only problems with twelve or more facilities, and ignoring problems with flow dominance less than 120 percent, FLAC outperformed the best of five CRAFT/FRAT runs in 19 out of 21 problems for a statistical significance of .00011, while the intermediate stage 2 results of $FLAC_i$ outperformed the median CRAFT/FRAT result 20 out of 21 times for a statistical significance of .00001. Even including the problems with low flow dominance, the conclusions remain unaltered, with FLAC outperforming the best of five CRAFT/FRAT runs 22 out of 26 times[1] (α=.00027) and $FLAC_i$ outperforming the median CRAFT/FRAT run 24 out of 27 times (α=.000025).

FLAC versus visual methods

The idea for the basic three-stage strategy used in FLAC came from an investigation of visual methods. For this reason, and because there appears to be doubt as to whether computer or

1 The only tie is not included.

visual methods should be used under certain conditions (see discussion of visual methods in chapter I above, p. 25), it is of interest to compare the results of FLAC with visual results where they are available.

Block[1] concluded from his experiments that visual methods are not competitive with CRAFT in solving Nugent's problems with low flow dominance. In his experiment, CRAFT equalled the best visual result on the four smaller problems and provided better results on the four problems with twelve or more facilities. Comparing his best visual solutions with the results of FLAC (see table 15) shows that FLAC performs similarly, showing superiority in the problems with twelve or more facilities and equalling the visual results on the smaller problems. The intermediate results of $FLAC_i$ perform almost as well.

Comparing the results of FLAC with the visual results to Grover's problems with high flow dominance obtained in Scriabin and Vergin's experiments,[2] table 16 shows that the best visual layouts are still better than or equal to the computer solutions. However, the FLAC computer algorithm provided very competitive results on the three largest problems, where visual methods had shown marked superiority, equalling the best visual result in the twelve and fifteen-facility problems, and providing a better solution to the largest

---

1 Block, "Note on Comparison of Computer Algorithms and Visual Based Methods."

2 Scriabin and Vergin, "Computer Algorithms and Visual Based Methods."

TABLE 15

COMPARISON OF FLAC AND VISUAL METHODS
IN PROBLEMS WITH LOW FLOW DOMINANCE

No. of facilities	Flow dom. (%)	Layout cost (efficiency)		
		$FLAC_i$	FLAC	Best visual Layout[a]
5	108	25= (100)	25= (100)	25 (100)
6	129	43= (89)	43= (89)	43 (89)
7	111	74= (84)	74= (84)	74 (84)
8	128	113 (67)	107= (77)	107 (77)
12	117	300+ (67)	289+ (74)	320 (54)
15	106	588+ (66)	585+ (67)	631 (52)
20	104	1310+ (58)	1303+ (59)	1378 (48)
30	112	3154+ (51)	3122+ (53)	3408 (37)

\+ indicates layout better than best visual layout.
= indicates layout equal to best visual layout.
a Block, "Note on Comparison of Computer Algorithms and Visual Based Methods."

TABLE 16

COMPARISON OF FLAC AND VISUAL METHODS IN PROBLEMS WITH HIGH FLOW DOMINANCE

No. of facilities	Flow dom. (%)	Layout cost(efficiency)			
		$FLAC_i$	FLAC	Visual layouts[a]	
				Median	Best
5	269	64.02== (94)	64.02== (94)	64.02 (94)	64.02 (94)
6	263	80.32== (95)	80.32== (95)	80.32 (95)	80.32 (95)
7	199	163.28== (85)	163.28== (85)	163.28 (85)	163.28 (85)
8	213	235.96 (76)	220.86 (85)	214.76 (89)	214.76 (89)
10	233	298.74 (78)	296.58 (79)	283.78 (84)	269.92 (89)
12	256	314.10+= (89)	314.10+= (89)	333.02 (83)	314.10 (89)
15	248	557.60+ (81)	511.40+= (89)	560.22 (80)	511.40 (89)
20	254	1154.78+ (79)	1127.40+ (81)	1158.56 (79)	1109.68 (83)

\+ indicates layout better than median visual layout.
== indicates layout equal to median and best visual layouts.
+= indicates layout better than median visual layout and equal to best visual layout.
a Scriabin and Vergin, "Computer Algorithms and Visual Based Methods," p. 177.

twenty-facility problems than the median solution of the trained group of Technical Management students from B.C.I.T. whose remarkably consistent performance was previously shown to be better than the computer solutions at the .002 level of significance.[1] The intermediate $FLAC_i$ results are competitive with the median visual solutions.

Computation Time

In problems such as facility layout, where optimum-seeking methods have been ruled out due to excessive computation time, it is important when evaluating any new method to consider not only solution quality but the computation time required to achieve the solutions. Table 17 shows the computation times of FLAC and several well-known algorithms in the solution of the four largest problems of Nugent et al.

The computation time of FLAC is clearly competitive with the fastest algorithms, however precise comparisons are difficult to make for two reasons:

(1) The computation times were recorded on different computer systems and

(2) programming efficiency cannot be taken into account.

As important as the computation time is the rate at which computation time increases with problem size.

A rough comparison of these rates, unbiased by the difference between computer systems and probably also

---

[1] Scriabin and Vergin, op. cit., p. 180.

TABLE 17

COMPUTATION TIMES[a] IN SECONDS PER SOLUTION FOR THE FOUR LARGEST PROBLEMS OF NUGENT *ET AL.*

Algorithm	No. of facilities				Power curve fit ($t=an^b$)		
	12	15	20	30	a	b	r^2
FLAC	3.7	3.6	7.4	22.8	.0154	2.1	.929
H-63[i]	55.0	78.0	168.0	398.0	.2117	2.2	.995
CQL	8	19	32	118	.0076	2.8	.987
HC63-66	19.0	40.0	75.0	285.0	.0145	2.9	.994
FLAC	4.7	8.3	25.1	60.6	.0039	2.9	.986
FRAT	1.25	2.96	10.87	58.30	.000033	4.2	.999
CRAFT[b]	70.0	160.0	528.0	3150.0	.0021	4.2	.999
TSP	6.84	20.13	59.87	383.27	.00015	4.3	.998
MAT	6.1	13.2	47.5	346.7	.000085	4.5	.996
Biased Sampling[b]	65.8	219.2	691.5	4272.4	.0010	4.5	.998

a Times other than for FLAC runs are from Hitchings and Cottam, "Efficient Heuristic Procedure," p. 213. FLAC and FRAT runs are on IBM 360 computer, H-63, HC63-66, CRAFT, MAT, and Biased Sampling are on G.E. 265 (MAT is conversion from IBM 7094), COL is on G.E. 635, and TSP is on I.C.L. 470.

b CRAFT times are for single runs from random initial layouts. Biased Sampling times are one tenth of the run time with sample size 10.

insensitive to programming efficiency, can be obtained by fitting a power curve of the form

$$t = an^b$$

to the computation times for each algorithm. Using only the twelve to thirty-facility problems and ignoring the run times of the smaller problems reduces bias due to any fixed overhead portion of the run times.[1]

The parameters a and b of the resulting power curves in the righthand columns of table 17 show that the algorithms naturally fall into three groups. The first group contains the intermediate results of $FLAC_i$ and the Hillier 1963 algorithm. The computation times of these two algorithms expand at a rate proportional to little more than the square of the problem size.

The second group contains FLAC and the two fast algorithms COL and HC63-66. Their computation time expands at a rate

---

[1] An argument can be made for a more detailed analysis of run times, using for instance a polynomial to isolate fixed overhead. Such an attempt at accuracy is unwarranted here since run times are affected not only by differences between algorithms, but by differences in the quality of initial solutions and the structure of the problems being solved. An accurate comparison would require many runs on a variety of problems by all the algorithms being compared.

proportional to a little less than the cube of the problem size.

The remainder of the algorithms cannot be competitive on larger problems because their computation times are proportional to more than the fourth power of the problem size.

All of the algorithms in the first two groups except $FLAC_i$ and FLAC have been shown in previous literature to produce inferior results to CRAFT.[1]

Three-way Exchange

The experiments in this dissertation have compared the new algorithm, FLAC, with the basic two-way exchange algorithm of CRAFT and FRAT. Since CRAFT has the capacity to also perform three-way exchanges, the exclusion of this variation of CRAFT must be justified.

In order to investigate the effect on computation time and solution quality, two small tests were performed.

In the first test, five CRAFT runs were made of the Nugent twelve-facility problem from random starts, using the three-way exchange algorithm when two-way exchanges could produce no further improvements. No three-way exchange could achieve any further improvement in any of the five runs. Computation time was nevertheless increased by a factor of 2.6 over the two-way exchange algorithm.

---

[1] For details, the reader is referred to the comparison of current algorithms in chapter I, above, p. 12.

In the second test, five similar runs were made of the Grover twenty-facility problem. In four of the five runs, additional three-way exchanges were performed, however the best solution, which was not as good as the intermediate $FLAC_i$ solution recorded in table 14, was achieved by the remaining run from two-way exchanges only. Computation time in this case was increased more than four-fold over the two-way exchange algorithm.

Although these two small tests by no means constitute an in-depth study of the effectiveness of the three-way exchange algorithm, the enormous increase in computation time of an already expensive algorithm indicated that research efforts could be more productively channeled in other areas.

Program Complexity

FLAC is a more complex algorithm than the heuristic methods which currently provide the best solutions to the facility layout problem, but that complexity requires additional effort only once at the programming stage. Although the simplicity of an algorithm is often presented as an advantage,[1] such an argument is less compelling in the case of an automated algorithm than a manual procedure, since the programming time can be amortized over several solutions.

---

[1] For instance, Khalil uses this argument in favour of FRAT in "Facilities Relative Allocation Technique."

Potential Further Improvements in Solution Quality

FLAC is a deterministic procedure, producing only one solution to any given problem. It would be very easy, however, to alter the final stage of FLAC to incorporate either Biased Sampling or to keep track of tie solutions as does TSP, using these as starting points for alternative solutions from which the best can be chosen.

While Biased Sampling has been shown to be no better than multiple runs of CRAFT when starting from a randomly chosen initial solution (see above, p. 17), this approach has much greater potential when applied to a good intermediate solution of $FLAC_i$. Since the number of two-way exchanges performed in stage 3 of FLAC is normally quite small, the parameter C of the Biased Sampling probability function (above, p. 10) would have to be set close to zero (i.e. so that exchanges are selected almost randomly as long as a positive cost reduction is achieved), to avoid duplication of solutions.

Backtracking _à la_ TSP to alternative tie solutions could also provide additional good solutions from which the best can be chosen. In the problems of Nugent _et al._, 1, 5, 2, and 3 ties were encountered in stage 3 of the FLAC solutions to the 12, 15, 20, and 30-facility problems, respectively.

While the attractiveness of such methods of generating additional solutions decreases with problem size due to the increased computation time, the inclusion of such methods in

FLAC may be practical for larger problems than when the methods are used alone, since only the computation time of stage 3 of FLAC would be increased. For example, the largest 40 facility problem (see Table 8, p.103) used in these experiements required an average of 50 iterations in each of the five FRAT runs from random initial layouts, encountering an average of 24 ties which could have been used as a basis for further improvements. In stage 3 of FLAC in the same problem only seven FRAT type exchanges were performed because of the good initial layout provided by $FLAC_i$ and six tie solutions were encountered, all of which would of course provide initial layouts at least as good as the $FLAC_i$ solution.

Potential Further Reduction in Computation Time

It is likely that the computation time of FLAC can be significantly reduced by two simple changes.

First, in all experiments performed for this dissertation, so as not to bias computation time measurements, the maximum and minimum values of the parameter α of the adjustment algorithm in stage 1 of FLAC (above,p.84) were set to extreme values of 50 and 0.01, respectively. No effort was made to reduce computation time through a judicious choice of these limits. An examination of the results shows that these values could have been set to 3 and 0.5 for all the problems with no change in solution quality, but saving many iterations through the adjustment routine.

Second, it is quite likely that the use of an approximation technique, such as Vogel's method, instead of the primal-dual algorithm in stage 2 of FLAC would provide equally good results, especially since we are not even at that stage directly optimizing the final cost objective. The use of such an approximation should greatly reduce the running time of that stage of FLAC.

Summary and Conclusions

An examination of research efforts in facility layout has led to the development of the computer algorithm FLAC (Facility Layout by the Analysis of Clusters) incorporating certain features hitherto used only in visual methods, in an effort to improve the effectiveness of automated methods.

Doubt has been cast on the value of flow dominance as a measure of the complexity of facility layout problems. Flow dominance has also been shown not to be a good measure of the line dominance it was originally purported to represent.

A comparison of current algorithms led to the choice of CRAFT (Computerized Relative Allocation of Facilities Technique) as the standard of comparison in terms of solution quality.

In large layout problems (twelve or more facilities) with intermediate or high flow dominance, the three-stage FLAC algorithm provided layouts significantly better (α=.00011) than the best of five runs of CRAFT or FRAT(Facilities Relative Allocation Technique), which were shown to be equivalent in terms of solution quality. In the same large problems, the intermediate layouts produced by $FLAC_i$ (before execution of the two-way exchange algorithm in stage 3 of FLAC) were shown to be superior to single runs of CRAFT or FRAT at a significance level of .00001.

Line dominance was not shown to have any effect on the relative performance of FLAC and CRAFT/FRAT, although in one test series the difference in efficiency of FLAC and CRAFT/FRAT appeared to increase with line dominance.

In Nugent's problems with low flow dominance FLAC performed better than individual CRAFT runs (α=.00781) while the intermediate $FLAC_i$ results were competitive with individual CRAFT runs.

In small problems (less than twelve facilities) both FLAC and $FLAC_i$ produced results competitive with but not significantly better than individual CRAFT or FRAT runs.

A comparison of FLAC with previously recorded results of visual methods showed that FLAC produces layouts competitive with those produced by visual methods, even in larger problems with high flow dominance, where visual methods had previously shown marked superiority to computer algorithms.

While three-way exchanges were shown to be impractical, other potential modifications to FLAC were identified which could result in further improvements in solution quality.

In addition to providing high quality solutions under widely varying conditions FLAC was shown to be superior in terms of computation time, since the only algorithms with competitive running times on large problems have previously been shown to produce inferior results. The rate of increase in computation time with problem size is very much more attractive than that of CRAFT or FRAT, with computation time of FLAC increasing at a rate proportional to less than the cube of problem size.

Suggestions have also been made which could lead to significant further reductions in computation time with little or no reduction in solution quality.

FLAC has therefore been shown to be a valuable addition to the repertoire of computer algorithms for facility layout. It must however be emphasized that FLAC is in itself not a complete solution to the problems faced by industrial engineers in laying out facilities, and to be effective any computer program incorporating FLAC should include methods for handling unequal areas and other practical considerations. Fortunately the design of FLAC allows the easy incorporation of such features since they would logically be added to stage 3. In fact, stage 3 could be replaced, if necessary, by the full blown version of CRAFT, which has already proven useful in practice.

It must also be observed that FLAC's usefulness is currently limited to those layout problems in which all n facilities are to be located, while several of the other algorithms, including CRAFT, can be used in the more general, albeit easier, case in which one or more facilities are fixed.

Finally, although the design of FLAC was influenced by the investigation of visual methods, it must be emphasized that even better solutions could be achieved by an interactive system incorporating man's visual capability and judgment as well as the FLAC algorithm.

APPENDIX A

FLOW-DISTANCE MATRICES FOR FIVE PROBLEMS WITH POISSON FLOWS AND VARYING FLOW DOMINANCE

FLOW-DISTANCE MATRIX FOR 30-FACILITY PROBLEM WITH POISSON FLOWS
FLOW DOMINANCE 96 PERCENT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	0	0	0	2	2	2	0	0	0	2	1	1	2	0	1	2	4	3	2	2	1	2	1	2	1	1	1	2	1
2	1	0	1	2	1	0	1	1	2	1	0	1	0	1	0	1	1	1	2	1	4	0	2	0	0	0	3	0	2	0
3	2	1	0	0	2	0	1	2	3	2	0	2	1	1	1	0	1	2	2	3	3	2	0	0	1	2	2	2	1	1
4	3	2	1	0	1	1	2	2	0	2	0	0	2	4	0	1	2	1	3	1	3	2	1	0	2	0	0	0	2	0
5	4	3	2	1	0	3	2	2	1	2	5	1	0	1	0	0	0	0	1	2	2	3	2	1	2	1	1	1	1	1
6	5	4	3	2	1	0	1	4	2	0	1	0	2	1	0	0	1	1	0	0	1	1	1	0	2	0	0	1	2	3
7	1	2	3	4	5	6	0	1	2	0	0	0	2	2	0	1	0	1	2	0	1	1	1	0	1	0	1	1	0	1
8	2	1	2	3	4	5	1	0	1	0	0	0	1	1	1	2	0	2	0	0	1	2	0	1	2	2	0	2	1	2
9	3	2	1	2	3	4	2	1	0	2	1	0	1	1	0	1	3	0	0	0	2	2	1	2	1	0	2	1	3	0
10	4	3	2	1	2	3	3	2	1	0	1	0	2	0	1	0	0	2	0	1	3	2	1	0	0	1	1	0	0	0
11	5	4	3	2	1	2	4	3	2	1	0	1	1	0	0	2	0	0	2	1	0	0	3	1	1	0	0	1	1	1
12	6	5	4	3	2	1	5	4	3	2	1	0	1	2	1	2	0	0	0	1	2	3	1	1	1	4	1	0	1	1
13	2	3	4	5	6	7	1	2	3	4	5	6	0	1	1	1	0	1	1	0	1	0	2	2	1	1	1	0	1	0
14	3	2	3	4	5	6	2	1	2	3	4	5	1	0	1	0	0	3	2	2	0	0	1	2	1	1	1	0	1	1
15	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	1	0	2	1	1	0	1	0	1	2	0	1	0	1	1
16	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	1	1	1	1	1	3	1	3	1	1	0	1	0	1
17	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	1	2	1	2	0	1	0	0	1	2	0	1	1
18	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0	0	1	0	2	1	0	2	0	3	2	2	0
19	3	4	5	6	7	8	2	3	4	5	6	7	1	2	3	4	5	6	0	3	0	2	0	2	2	0	2	1	0	1
20	4	3	4	5	6	7	3	2	3	4	5	6	2	1	2	3	4	5	1	0	1	0	0	1	0	1	2	1	2	1
21	5	4	3	4	5	6	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	0	2	0	0	0	1	2	0	1
22	6	5	4	3	4	5	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	0	2	0	1	1	2	1	1
23	7	6	5	4	3	4	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	2	0	3	1	3	0	2
24	8	7	6	5	4	3	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0	0	1	0	2	1	0
25	4	5	6	7	8	9	3	4	5	6	7	8	2	3	4	5	6	7	1	2	3	4	5	6	0	0	2	0	0	0
26	5	4	5	6	7	8	4	3	4	5	6	7	3	2	3	4	5	6	2	1	2	3	4	5	1	0	1	0	1	0
27	6	5	4	5	6	7	5	4	3	4	5	6	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	0	0	0
28	7	6	5	4	5	6	6	5	4	3	4	5	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	3	0
29	8	7	6	5	4	5	7	6	5	4	3	4	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	1
30	9	8	7	6	5	4	8	7	6	5	4	3	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0

FLOW-DISTANCE MATRIX FOR 30-FACILITY PROBLEM WITH POISSON FLOWS
FLOW DOMINANCE 115 PERCENT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	0	0	0	1	2	1	0	0	0	1	1	0	2	0	1	1	3	2	1	1	1	1	1	1	0	1	0	1	0
2	1	0	1	1	1	0	1	0	2	0	0	0	0	1	0	1	1	1	1	1	3	0	1	0	0	0	2	0	1	0
3	2	1	0	0	1	0	0	2	2	2	0	1	1	0	0	0	1	2	1	2	3	2	0	0	0	1	1	1	0	0
4	3	2	1	0	1	1	1	1	0	1	0	0	1	3	0	0	1	1	2	0	2	1	0	0	1	0	0	0	2	0
5	4	3	2	1	0	2	1	2	1	1	4	0	0	0	0	0	0	0	0	1	2	2	1	0	1	1	0	0	1	1
6	5	4	3	2	1	0	1	3	1	0	1	0	2	1	0	0	0	1	0	0	1	1	1	0	2	0	0	1	1	2
7	1	2	3	4	5	6	0	1	1	0	0	0	1	1	0	0	0	1	1	0	1	1	0	0	1	0	0	1	0	1
8	2	1	2	3	4	5	1	0	0	0	0	0	1	0	1	1	0	1	0	0	0	1	0	0	2	1	0	1	1	2
9	3	2	1	2	3	4	2	1	0	2	0	0	1	1	0	1	3	0	0	0	1	1	0	2	1	0	1	0	3	0
10	4	3	2	1	2	3	3	2	1	0	1	0	2	0	1	0	0	2	0	1	2	1	1	0	0	1	1	0	0	0
11	5	4	3	2	1	2	4	3	2	1	0	1	0	0	0	1	0	0	1	0	0	0	3	1	1	0	0	1	1	1
12	6	5	4	3	2	1	5	4	3	2	1	0	1	1	0	1	0	0	0	0	1	2	1	1	1	3	1	0	1	0
13	2	3	4	5	6	7	1	2	3	4	5	6	0	0	1	1	0	1	1	0	1	0	2	1	1	0	0	0	1	0
14	3	2	3	4	5	6	2	1	2	3	4	5	1	0	1	0	0	2	1	1	0	0	1	1	0	0	0	0	0	1
15	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	1	0	1	1	1	0	1	0	0	1	0	1	0	1	1
16	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	1	1	1	1	1	3	1	2	0	1	0	0	0	1
17	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	1	2	0	1	0	0	0	0	1	2	0	0	1
18	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0	0	1	0	1	0	0	2	0	2	2	1	0
19	3	4	5	6	7	8	2	3	4	5	6	7	1	2	3	4	5	6	0	3	0	1	0	1	1	0	1	1	0	1
20	4	3	4	5	6	7	3	2	3	4	5	6	2	1	2	3	4	5	1	0	1	0	0	1	0	1	1	1	1	1
21	5	4	3	4	5	6	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	0	1	0	0	0	1	2	0	1
22	6	5	4	3	4	5	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	0	2	0	0	1	1	0	1
23	7	6	5	4	3	4	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	1	0	3	1	2	0	2
24	8	7	6	5	4	3	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0	0	0	0	2	1	0
25	4	5	6	7	8	9	3	4	5	6	7	8	2	3	4	5	6	7	1	2	3	4	5	6	0	0	1	0	0	0
26	5	4	5	6	7	8	4	3	4	5	6	7	3	2	3	4	5	6	2	1	2	3	4	5	1	0	1	0	1	0
27	6	5	4	5	6	7	5	4	3	4	5	6	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	0	0	0
28	7	6	5	4	5	6	6	5	4	3	4	5	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	3	0
29	8	7	6	5	4	5	7	6	5	4	3	4	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	1
30	9	8	7	6	5	4	8	7	6	5	4	3	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0

FLOW-DISTANCE MATRIX FOR 30-FACILITY PROBLEM WITH POISSON FLOWS
FLOW DOMINANCE 144 PERCENT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	0	0	0	1	1	1	0	0	0	1	0	0	1	0	0	1	2	2	1	1	1	1	0	1	0	0	0	1	0
2	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	2	0	1	0	0	0	1	0	1	0
3	2	1	0	0	1	0	0	1	1	1	0	1	1	0	0	0	0	1	1	2	2	1	0	0	0	1	1	1	0	0
4	3	2	1	0	0	0	1	1	0	1	0	0	1	2	0	0	1	0	2	0	2	1	0	0	1	0	0	0	1	0
5	4	3	2	1	0	2	1	1	0	1	3	0	0	0	0	0	0	0	0	1	1	2	1	0	1	1	0	0	0	0
6	5	4	3	2	1	0	0	2	1	0	0	0	1	1	0	0	0	1	0	0	1	0	1	0	1	0	0	1	1	2
7	1	2	3	4	5	6	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0
8	2	1	2	3	4	5	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1	1	0	1	1	1
9	3	2	1	2	3	4	2	1	0	1	0	0	0	0	0	1	2	0	0	0	1	1	0	1	0	0	1	0	2	0
10	4	3	2	1	2	3	3	2	1	0	1	0	1	0	1	0	0	1	0	0	2	1	0	0	0	0	0	0	0	0
11	5	4	3	2	1	2	4	3	2	1	0	0	0	0	0	1	0	0	1	0	0	0	2	0	1	0	0	0	0	1
12	6	5	4	3	2	1	5	4	3	2	1	0	1	1	0	1	0	0	0	0	1	1	0	0	0	2	1	0	0	0
13	2	3	4	5	6	7	1	2	3	4	5	6	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	0
14	3	2	3	4	5	6	2	1	2	3	4	5	1	0	1	0	0	2	1	1	0	0	0	1	0	0	0	0	0	0
15	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	1	0
16	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	0	0	1	0	1	2	1	2	0	0	0	0	0	1
17	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	1	1	0	1	0	0	0	0	0	1	0	0	1
18	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0	0	0	0	1	0	0	1	0	1	1	1	0
19	3	4	5	6	7	8	2	3	4	5	6	7	1	2	3	4	5	6	0	2	0	1	0	1	1	0	1	0	0	0
20	4	3	4	5	6	7	3	2	3	4	5	6	2	1	2	3	4	5	1	0	0	0	0	0	0	0	1	1	1	1
21	5	4	3	4	5	6	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	0	1	0	0	0	0	1	0	0
22	6	5	4	3	4	5	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	0	1	0	0	0	1	0	0
23	7	6	5	4	3	4	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	1	0	2	0	2	0	1
24	8	7	6	5	4	3	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0	0	0	0	1	1	0
25	4	5	6	7	8	9	3	4	5	6	7	8	2	3	4	5	6	7	1	2	3	4	5	6	0	0	1	0	0	0
26	5	4	5	6	7	8	4	3	4	5	6	7	3	2	3	4	5	6	2	1	2	3	4	5	1	0	1	0	0	0
27	6	5	4	5	6	7	5	4	3	4	5	6	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	0	0	0
28	7	6	5	4	5	6	6	5	4	3	4	5	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	2	0
29	8	7	6	5	4	5	7	6	5	4	3	4	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	0
30	9	8	7	6	5	4	8	7	6	5	4	3	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0

FLOW-DISTANCE MATRIX FOR 30-FACILITY PROBLEM WITH POISSON FLOWS
FLOW DOMINANCE 199 PERCENT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	1	2	1	1	0	0	1	0	0	0	0	0	1	0
2	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2	0	1	0	0	0	1	0	1	0
3	2	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
4	3	2	1	0	0	0	0	1	0	1	0	0	1	2	0	0	1	0	1	0	1	1	0	0	1	0	0	0	1	0
5	4	3	2	1	0	1	1	1	0	0	3	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0	0	0	0	0
6	5	4	3	2	1	0	0	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1
7	1	2	3	4	5	6	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
8	2	1	2	3	4	5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1	0	1
9	3	2	1	2	3	4	2	1	0	1	0	0	0	0	0	0	1	0	0	0	1	1	0	1	0	0	0	0	2	0
10	4	3	2	1	2	3	3	2	1	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0
11	5	4	3	2	1	2	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
12	6	5	4	3	2	1	5	4	3	2	1	0	0	1	0	0	0	0	0	0	1	1	0	0	0	2	0	0	0	0
13	2	3	4	5	6	7	1	2	3	4	5	6	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
14	3	2	3	4	5	6	2	1	2	3	4	5	1	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0
15	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
16	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	0	0	0	0	0	2	0	1	0	0	0	0	0	0
17	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	0	1	0	1	0	0	0	0	0	1	0	0	0
18	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0	0	0	0	1	0	0	1	0	1	1	1	0
19	3	4	5	6	7	8	2	3	4	5	6	7	1	2	3	4	5	6	0	2	0	0	0	1	1	0	1	0	0	0
20	4	3	4	5	6	7	3	2	3	4	5	6	2	1	2	3	4	5	1	0	0	0	0	0	0	0	1	0	0	0
21	5	4	3	4	5	6	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	0	0	0	0	0	0	1	0	0
22	6	5	4	3	4	5	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	0	1	0	0	0	1	0	0
23	7	6	5	4	3	4	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	1	0	2	0	1	0	1
24	8	7	6	5	4	3	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0	0	0	0	1	0	0
25	4	5	6	7	8	9	3	4	5	6	7	8	2	3	4	5	6	7	1	2	3	4	5	6	0	0	1	0	0	0
26	5	4	5	6	7	8	4	3	4	5	6	7	3	2	3	4	5	6	2	1	2	3	4	5	1	0	0	0	0	0
27	6	5	4	5	6	7	5	4	3	4	5	6	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	0	0	0
28	7	6	5	4	5	6	6	5	4	3	4	5	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	2	0
29	8	7	6	5	4	5	7	6	5	4	3	4	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	0
30	9	8	7	6	5	4	8	7	6	5	4	3	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0

FLOW-DISTANCE MATRIX FOR 30-FACILITY PROBLEM WITH POISSON FLOWS
FLOW DOMINANCE 282 PERCENT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
3	2	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0
4	3	2	1	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0
5	4	3	2	1	0	1	0	1	0	0	2	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
6	5	4	3	2	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
7	1	2	3	4	5	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	2	1	2	3	4	5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
9	3	2	1	2	3	4	2	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0
10	4	3	2	1	2	3	3	2	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
11	5	4	3	2	1	2	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
12	6	5	4	3	2	1	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
13	2	3	4	5	6	7	1	2	3	4	5	6	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
14	3	2	3	4	5	6	2	1	2	3	4	5	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
15	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
17	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
18	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0	0	0	0	0	0	0	0	0	1	0	0	0
19	3	4	5	6	7	8	2	3	4	5	6	7	1	2	3	4	5	6	0	1	0	0	0	0	0	0	0	0	0	0
20	4	3	4	5	6	7	3	2	3	4	5	6	2	1	2	3	4	5	1	0	0	0	0	0	0	0	0	0	0	0
21	5	4	3	4	5	6	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	0	0	0	0	0	0	1	0	0
22	6	5	4	3	4	5	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	0	1	0	0	0	0	0	0
23	7	6	5	4	3	4	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	0	0	1	0	1	0	1
24	8	7	6	5	4	3	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0	0	0	0	1	0	0
25	4	5	6	7	8	9	3	4	5	6	7	8	2	3	4	5	6	7	1	2	3	4	5	6	0	0	0	0	0	0
26	5	4	5	6	7	8	4	3	4	5	6	7	3	2	3	4	5	6	2	1	2	3	4	5	1	0	0	0	0	0
27	6	5	4	5	6	7	5	4	3	4	5	6	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	0	0	0
28	7	6	5	4	5	6	6	5	4	3	4	5	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	1	0
29	8	7	6	5	4	5	7	6	5	4	3	4	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	0
30	9	8	7	6	5	4	8	7	6	5	4	3	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0

APPENDIX B

FLOW-DISTANCE MATRICES FOR FIVE PROBLEMS WITH VARYING LINE DOMINANCE

FLOW-DISTANCE MATRIX FOR 20-FACILITY LINEMIX PROBLEM
LINE DOMINANCE: 120 LINES OF LENGTH 1 (NO LINE DOMINANCE)
FLOW DOMINANCE 143 PERCENT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	0	0	0	0	0	0	7	4	10	2	3	0	0	0	2	0	14	0
2	1	0	4	4	7	0	5	4	0	0	0	2	5	0	2	0	0	0	0	0
3	2	1	0	0	0	0	6	1	7	0	0	0	0	5	3	0	0	0	0	0
4	3	2	1	0	0	0	1	6	4	0	0	0	3	5	0	3	0	0	0	0
5	4	3	2	1	0	2	0	0	3	4	0	9	5	0	0	3	3	0	0	3
6	1	2	3	4	5	0	0	2	0	5	0	2	0	5	3	1	5	0	2	0
7	2	1	2	3	4	1	0	0	4	2	1	0	0	0	0	2	0	6	0	8
8	3	2	1	2	3	2	1	0	0	0	0	0	0	1	0	5	3	0	2	1
9	4	3	2	1	2	3	2	1	0	3	2	5	3	0	0	1	0	0	0	1
10	5	4	3	2	1	4	3	2	1	0	0	5	6	4	2	4	10	0	4	7
11	2	3	4	5	6	1	2	3	4	5	0	8	0	3	4	0	3	9	7	0
12	3	2	3	4	5	2	1	2	3	4	1	0	0	0	11	5	5	4	0	0
13	4	3	2	3	4	3	2	1	2	3	2	1	0	2	2	0	0	4	1	5
14	5	4	3	2	3	4	3	2	1	2	3	2	1	0	0	0	0	0	4	1
15	6	5	4	3	2	5	4	3	2	1	4	3	2	1	0	0	2	5	1	0
16	3	4	5	6	7	2	3	4	5	6	1	2	3	4	5	0	0	0	0	0
17	4	3	4	5	6	3	2	3	4	5	2	1	2	3	4	1	0	4	4	0
18	5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	2	1	0	0	0
19	6	5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	2	1	0	3
20	7	6	5	4	3	6	5	4	3	2	5	4	3	2	1	4	3	2	1	0

FLOW-DISTANCE MATRIX FOR 20-FACILITY LINEMIX PROBLEM
LINE DOMINANCE: 24 LINES OF LENGTH 5
FLOW DOMINANCE 140 PERCENT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	9	0	4	0	3	0	0	5	0	0	0	0	0	0	2	4	0	0
2	1	0	0	2	0	0	9	0	1	0	3	4	0	0	3	3	0	2	0	5
3	2	1	0	8	0	0	0	2	3	3	0	2	5	0	0	0	3	0	0	1
4	3	2	1	0	5	3	5	0	2	4	4	2	10	7	3	4	2	0	0	6
5	4	3	2	1	0	2	0	0	0	0	0	3	0	4	0	0	0	5	1	0
6	1	2	3	4	5	0	0	3	1	4	0	0	5	0	0	0	3	0	0	8
7	2	1	2	3	4	1	0	3	0	5	0	2	1	0	2	3	0	5	0	4
8	3	2	1	2	3	2	1	0	0	0	1	1	1	6	0	0	1	0	0	0
9	4	3	2	1	2	3	2	1	0	5	0	0	3	0	3	0	1	4	0	3
10	5	4	3	2	1	4	3	2	1	0	0	0	0	0	5	0	9	5	2	2
11	2	3	4	5	6	1	2	3	4	5	0	0	5	0	0	0	3	0	0	0
12	3	2	3	4	5	2	1	2	3	4	1	0	3	3	0	0	0	2	0	1
13	4	3	2	3	4	3	2	1	2	3	2	1	0	3	8	0	3	3	0	9
14	5	4	3	2	3	4	3	2	1	2	3	2	1	0	3	0	0	0	0	3
15	6	5	4	3	2	5	4	3	2	1	4	3	2	1	0	0	3	0	0	2
16	3	4	5	6	7	2	3	4	5	6	1	2	3	4	5	0	0	0	0	7
17	4	3	4	5	6	3	2	3	4	5	2	1	2	3	4	1	0	3	2	0
18	5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	2	1	0	0	5
19	6	5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	2	1	0	0
20	7	6	5	4	3	6	5	4	3	2	5	4	3	2	1	4	3	2	1	0

FLOW-DISTANCE MATRIX FOR 20-FACILITY LINEMIX PROBLEM
LINE DOMINANCE: 15 LINES OF LENGTH 8
FLOW DOMINANCE 142 PERCENT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	9	0	0	0	5	0	1	0	0	1	0	0	0	0	0	0	0	4
2	1	0	0	0	0	0	4	4	0	0	0	0	0	0	0	5	5	0	0	2
3	2	1	0	5	3	0	4	4	0	5	5	3	4	0	0	5	1	0	4	0
4	3	2	1	0	0	3	0	0	0	5	0	0	0	2	5	4	2	4	0	19
5	4	3	2	1	0	0	0	4	3	5	0	0	0	5	0	5	7	2	4	0
6	1	2	3	4	5	0	10	5	0	0	0	0	4	12	0	10	4	0	0	0
7	2	1	2	3	4	1	0	0	5	0	3	1	0	4	0	0	4	0	4	5
8	3	2	1	2	3	2	1	0	4	3	0	2	5	0	7	4	0	2	0	2
9	4	3	2	1	2	3	2	1	0	0	0	0	3	0	0	4	0	1	0	2
10	5	4	3	2	1	4	3	2	1	0	8	3	11	0	0	6	5	6	0	1
11	2	3	4	5	6	1	2	3	4	5	0	3	1	1	2	0	10	2	3	0
12	3	2	3	4	5	2	1	2	3	4	1	0	5	0	0	7	0	0	3	2
13	4	3	2	3	4	3	2	1	2	3	2	1	0	0	0	0	5	0	3	2
14	5	4	3	2	3	4	3	2	1	2	3	2	1	0	6	2	4	0	0	0
15	6	5	4	3	2	5	4	3	2	1	4	3	2	1	0	0	0	2	0	4
16	3	4	5	6	7	2	3	4	5	6	1	2	3	4	5	0	2	0	0	4
17	4	3	4	5	6	3	2	3	4	5	2	1	2	3	4	1	0	0	0	4
18	5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	2	1	0	6	5
19	6	5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	2	1	0	0
20	7	6	5	4	3	6	5	4	3	2	5	4	3	2	1	4	3	2	1	0

FLOW-DISTANCE MATRIX FOR 20-FACILITY LINEMIX PROBLEM
LINE DOMINANCE: 12 LINES OF LENGTH 10
FLOW DOMINANCE 143 PERCENT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	5	5	1	0	2	3	3	0	9	0	0	0	0	0	0	0	2	0	0
2	1	0	1	10	2	0	0	0	3	0	0	1	0	0	4	0	0	5	1	0
3	2	1	0	0	0	5	0	3	4	4	0	3	0	2	0	6	0	5	5	0
4	3	2	1	0	5	0	5	2	0	3	0	4	3	0	2	4	7	0	0	0
5	4	3	2	1	0	4	8	4	0	2	2	9	0	1	5	0	0	0	5	0
6	1	2	3	4	5	0	0	0	6	0	0	5	3	0	0	0	4	0	0	5
7	2	1	2	3	4	1	0	1	0	0	0	0	0	4	0	0	0	3	0	0
8	3	2	1	2	3	2	1	0	4	1	0	4	0	0	0	4	0	0	9	8
9	4	3	2	1	2	3	2	1	0	0	0	0	0	0	0	8	1	4	3	0
10	5	4	3	2	1	4	3	2	1	0	5	7	0	0	0	0	1	0	1	2
11	2	3	4	5	6	1	2	3	4	5	0	7	0	5	0	0	0	2	1	0
12	3	2	3	4	5	2	1	2	3	4	1	0	3	2	2	2	0	0	0	0
13	4	3	2	3	4	3	2	1	2	3	2	1	0	7	3	3	4	3	0	1
14	5	4	3	2	3	4	3	2	1	2	3	2	1	0	0	1	4	0	4	0
15	6	5	4	3	2	5	4	3	2	1	4	3	2	1	0	8	0	0	9	0
16	3	4	5	6	7	2	3	4	5	6	1	2	3	4	5	0	3	0	2	0
17	4	3	4	5	6	3	2	3	4	5	2	1	2	3	4	1	0	0	0	0
18	5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	2	1	0	0	1
19	6	5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	2	1	0	1
20	7	6	5	4	3	6	5	4	3	2	5	4	3	2	1	4	3	2	1	0

FLOW-DISTANCE MATRIX FOR 20-FACILITY LINEMIX PROBLEM
LINE DOMINANCE: 8 LINES OF LENGTH 15
FLOW DOMINANCE 143 PERCENT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	9	0	2	1	3	3	0	0	1	0	0	5	5	0	0	0	3	0	4
2	1	0	1	6	2	2	0	0	3	0	0	1	0	0	0	0	2	5	1	0
3	2	1	0	0	0	5	1	0	1	4	0	3	0	2	4	6	0	5	2	0
4	3	2	1	0	2	0	6	2	0	3	4	0	3	0	2	1	7	0	0	0
5	4	3	2	1	0	0	5	4	0	2	2	4	1	1	0	0	0	0	5	0
6	1	2	3	4	5	0	0	0	6	1	0	2	0	2	0	0	0	0	0	0
7	2	1	2	3	4	1	0	1	0	0	0	0	1	5	0	0	0	0	2	0
8	3	2	1	2	3	2	1	0	1	1	0	3	2	0	0	4	2	2	5	3
9	4	3	2	1	2	3	2	1	0	0	0	0	0	0	0	5	1	0	0	0
10	5	4	3	2	1	4	3	2	1	0	0	4	0	0	0	0	1	4	1	2
11	2	3	4	5	6	1	2	3	4	5	0	2	0	1	4	0	0	0	1	0
12	3	2	3	4	5	2	1	2	3	4	1	0	0	7	7	2	0	2	0	0
13	4	3	2	3	4	3	2	1	2	3	2	1	0	7	0	3	11	0	0	1
14	5	4	3	2	3	4	3	2	1	2	3	2	1	0	0	1	0	1	4	0
15	6	5	4	3	2	5	4	3	2	1	4	3	2	1	0	5	0	3	2	4
16	3	4	5	6	7	2	3	4	5	6	1	2	3	4	5	0	3	0	2	0
17	4	3	4	5	6	3	2	3	4	5	2	1	2	3	4	1	0	2	0	0
18	5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	2	1	0	1	1
19	6	5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	2	1	0	1
20	7	6	5	4	3	6	5	4	3	2	5	4	3	2	1	4	3	2	1	0

APPENDIX C

LINEMIX FUNCTION PROGRAM LISTING

LISTING OF LINEMIX APL FUNCTION USED TO GENERATE PROBLEMS WITH LINE DOMINANCE

```
∇LINEMIX[⎕]∇
      ∇ RTLEN LINEMIX ROUTS;I;J;K;NR;RR;STRING
[1]    STRING←'LINEMIX ROUTE LENGTH ',(4 0 ⍕RTLEN),'  MAX. RT. VOL.'
[2]    QOUT STRING,(4 0 ⍕MX),'  NO. OF ROUTES ', 4 0 ⍕ROUTS
[3]    F←(N,N)⍴0
[4]    NR←0
[5]   LOOP:→(ROUTS<NR←NR+1)/OUT3
[6]    J←RTLEN⌊N-1
[7]    K←?MX
[8]    I←0
[9]   LOOP2:→(J<I←I+1)/OUT2
[10]   F[I+1;I]←F[I;I+1]←F[I;I+1]+K
[11]   →LOOP2
[12]  OUT2:F←F[RR;RR←N?N]
[13]   →LOOP
[14]  OUT3:F
[15]   →0
      ∇
```

APPENDIX D

FLOW-DISTANCE MATRICES FOR SEVEN PROBLEMS WITH HIGH LINE DOMINANCE AND VARYING PROBLEM SIZE

FLOW-DISTANCE MATRIX FOR 12-FACILITY
LINEMIX PROBLEM
LINE DOMINANCE: 7 LINES OF LENGTH 6
FLOW DOMINANCE 147 PERCENT

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	4	2	1	0	1	0	0	4	0	0	4
2	1	0	0	5	1	0	1	4	6	0	0	9
3	2	1	0	0	4	4	0	0	4	0	4	0
4	3	2	1	0	0	0	0	0	0	5	1	0
5	1	2	3	4	0	2	5	0	0	13	2	0
6	2	1	2	3	1	0	1	0	0	6	0	0
7	3	2	1	2	2	1	0	2	0	0	6	4
8	4	3	2	1	3	2	1	0	2	0	0	6
9	2	3	4	5	1	2	3	4	0	0	0	5
10	3	2	3	4	2	1	2	3	1	0	8	6
11	4	3	2	3	3	2	1	2	2	1	0	0
12	5	4	3	2	4	3	2	1	3	2	1	0

FLOW-DISTANCE MATRIX FOR 15-FACILITY
LINEMIX PROBLEM
LINE DOMINANCE: 10 LINES OF LENGTH 7
FLOW DOMINANCE 146 PERCENT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	3	1	0	0	9	3	0	5	0	0	2	9
2	1	0	3	6	4	0	0	0	5	2	1	3	0	0	2
3	2	1	0	3	3	6	0	0	13	0	7	5	1	0	5
4	3	2	1	0	2	2	3	0	2	5	0	0	0	0	0
5	1	2	3	4	0	0	2	0	2	0	0	0	0	0	2
6	2	1	2	3	1	0	0	0	0	7	0	0	0	0	0
7	3	2	1	2	2	1	0	0	5	0	0	7	6	5	0
8	4	3	2	1	3	2	1	0	0	2	5	5	7	0	0
9	2	3	4	5	1	2	3	4	0	0	2	10	9	0	0
10	3	2	3	4	2	1	2	3	1	0	2	0	0	0	0
11	4	3	2	3	3	2	1	2	2	1	0	0	0	7	2
12	5	4	3	2	4	3	2	1	3	2	1	0	0	0	0
13	3	4	5	6	2	3	4	5	1	2	3	4	0	4	7
14	4	3	4	5	3	2	3	4	2	1	2	3	1	0	4
15	5	4	3	4	4	3	2	3	3	2	1	2	2	1	0

FLOW-DISTANCE MATRIX FOR 20-FACILITY LINEMIX PROBLEM
LINE DOMINANCE: 10 LINES OF LENGTH 10
FLOW DOMINANCE 141 PERCENT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	0	6	1	0	0	4	0	0	0	0	0	1	6	0	2	3	3	4
2	1	0	4	0	0	4	4	5	0	0	2	0	2	0	3	4	0	2	0	0
3	2	1	0	0	0	0	5	0	2	0	0	4	0	5	0	0	3	4	0	2
4	3	2	1	0	0	3	0	4	2	1	0	4	4	0	0	0	5	1	1	3
5	4	3	2	1	0	6	0	5	0	1	1	0	5	0	3	0	6	0	4	0
6	1	2	3	4	5	0	0	3	1	1	2	1	0	5	7	2	0	0	0	0
7	2	1	2	3	4	1	0	0	0	0	0	3	0	1	0	5	3	4	1	0
8	3	2	1	2	3	2	1	0	0	0	3	4	0	0	4	0	0	0	0	0
9	4	3	2	1	2	3	2	1	0	5	4	0	0	2	0	0	0	0	1	0
10	5	4	3	2	1	4	3	2	1	0	0	0	5	0	0	0	1	2	0	6
11	2	3	4	5	6	1	2	3	4	5	0	0	0	0	4	0	4	0	0	0
12	3	2	3	4	5	2	1	2	3	4	1	0	0	0	4	5	0	0	0	3
13	4	3	2	3	4	3	2	1	2	3	2	1	0	0	0	0	0	0	0	2
14	5	4	3	2	3	4	3	2	1	2	3	2	1	0	0	0	0	0	0	3
15	6	5	4	3	2	5	4	3	2	1	4	3	2	1	0	0	0	3	0	6
16	3	4	5	6	7	2	3	4	5	6	1	2	3	4	5	0	3	4	4	8
17	4	3	4	5	6	3	2	3	4	5	2	1	2	3	4	1	0	0	3	0
18	5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	2	1	0	5	0
19	6	5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	2	1	0	9
20	7	6	5	4	3	6	5	4	3	2	5	4	3	2	1	4	3	2	1	0

FLOW-DISTANCE MATRIX FOR 20-FACILITY LINEMIX PROBLEM
LINE DOMINANCE: 12 LINES OF LENGTH 10
FLOW DOMINANCE 143 PERCENT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	5	5	1	0	2	3	3	0	9	0	0	0	0	0	0	0	2	0	0
2	1	0	1	10	2	0	0	0	3	0	0	1	0	0	4	0	0	5	1	0
3	2	1	0	0	0	5	0	3	4	4	0	3	0	2	0	6	0	5	5	0
4	3	2	1	0	5	0	5	2	0	3	0	4	3	0	2	4	7	0	0	0
5	4	3	2	1	0	4	8	4	0	2	2	9	0	1	5	0	0	0	5	0
6	1	2	3	4	5	0	0	0	6	0	0	5	3	0	0	0	4	0	0	5
7	2	1	2	3	4	1	0	1	0	0	0	0	0	4	0	0	0	3	0	0
8	3	2	1	2	3	2	1	0	4	1	0	4	0	0	0	4	0	0	9	8
9	4	3	2	1	2	3	2	1	0	0	0	0	0	0	0	8	1	4	3	0
10	5	4	3	2	1	4	3	2	1	0	5	7	0	0	0	0	1	0	1	2
11	2	3	4	5	6	1	2	3	4	5	0	7	0	5	0	0	0	2	1	0
12	3	2	3	4	5	2	1	2	3	4	1	0	3	2	2	2	0	0	0	0
13	4	3	2	3	4	3	2	1	2	3	2	1	0	7	3	3	4	3	0	1
14	5	4	3	2	3	4	3	2	1	2	3	2	1	0	0	1	4	0	4	0
15	6	5	4	3	2	5	4	3	2	1	4	3	2	1	0	8	0	0	9	0
16	3	4	5	6	7	2	3	4	5	6	1	2	3	4	5	0	3	0	2	0
17	4	3	4	5	6	3	2	3	4	5	2	1	2	3	4	1	0	0	0	0
18	5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	2	1	0	0	1
19	6	5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	2	1	0	1
20	7	6	5	4	3	6	5	4	3	2	5	4	3	2	1	4	3	2	1	0

FLOW-DISTANCE MATRIX FOR 30-FACILITY LINEMIX PROBLEM
LINE DOMINANCE: 20 LINES OF LENGTH 15
FLOW DOMINANCE 137 PERCENT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	0	0	0	1	0	0	5	4	6	3	0	0	3	0	0	1	4	3	0	4	0	0	1	0	6	0	1	0	0
2	1	0	0	5	0	7	2	0	5	1	0	1	1	4	11	4	11	2	0	1	0	3	0	0	5	1	1	9	0	1
3	2	1	0	0	4	4	2	4	5	1	1	5	0	0	0	0	0	0	0	0	8	1	4	1	4	0	0	4	0	1
4	3	2	1	0	9	3	3	0	0	0	0	0	0	1	4	0	1	5	0	8	10	0	0	3	0	6	0	4	3	1
5	4	3	2	1	0	0	0	5	0	0	0	1	1	3	0	6	2	3	6	5	7	0	0	4	4	0	0	2	0	0
6	5	4	3	2	1	0	1	0	0	3	5	0	5	0	0	0	1	0	0	1	2	6	2	5	0	5	0	8	0	0
7	1	2	3	4	5	6	0	0	3	0	8	0	4	6	0	0	0	4	0	3	0	0	4	1	4	0	5	6	0	0
8	2	1	2	3	4	5	1	0	5	5	4	1	2	0	2	5	0	0	0	0	4	0	5	1	0	0	0	6	0	5
9	3	2	1	2	3	4	2	1	0	0	0	0	0	0	6	3	3	0	0	0	7	11	0	0	1	5	0	0	4	0
10	4	3	2	1	2	3	3	2	1	0	0	0	0	0	2	0	4	0	0	1	4	1	0	13	0	0	0	0	0	4
11	5	4	3	2	1	2	4	3	2	1	0	0	0	0	5	0	6	0	1	0	5	3	1	0	0	3	0	8	1	4
12	6	5	4	3	2	1	5	4	3	2	1	0	0	0	1	8	0	4	3	5	0	11	0	0	7	3	0	0	0	0
13	2	3	4	5	6	7	1	2	3	4	5	6	0	4	3	1	0	4	3	0	0	0	1	0	3	0	0	0	4	3
14	3	2	3	4	5	6	2	1	2	3	4	5	1	0	5	6	5	3	2	0	3	0	3	0	7	5	0	0	3	0
15	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	8	4	0	0	0	3	0	7	0	1	1	0	0	5	5
16	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	0	9	0	6	1	0	0	0	0	0	0	1	2	0
17	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	3	0	0	0	6	1	0	7	1	5	0	0	0
18	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0	8	3	1	0	0	9	0	0	3	3	4	5
19	3	4	5	6	7	8	2	3	4	5	6	7	1	2	3	4	5	6	0	0	3	5	0	0	0	0	3	1	5	1
20	4	3	4	5	6	7	3	2	3	4	5	6	2	1	2	3	4	5	1	0	3	9	3	10	1	1	0	0	0	0
21	5	4	3	4	5	6	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	5	3	0	4	0	0	0	0	0
22	6	5	4	3	4	6	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	3	3	0	1	0	0	0	0
23	7	6	5	4	3	4	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	1	3	0	0	0	0	9
24	8	7	6	5	4	3	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0	0	8	5	8	0	3
25	4	5	6	7	8	9	3	4	5	6	7	8	2	3	4	5	6	7	1	2	3	4	5	6	0	0	0	0	8	0
26	5	4	5	6	7	8	4	3	4	5	6	7	3	2	3	4	5	6	2	1	2	3	4	5	1	0	5	1	5	4
27	6	5	4	5	6	7	5	4	3	4	5	6	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	0	5	0
28	7	6	5	4	5	6	6	5	4	3	4	5	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	0	0
29	8	7	6	5	4	5	7	6	5	4	3	4	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	0
30	9	8	7	6	5	4	8	7	6	5	4	3	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0

FLOW-DISTANCE MATRIX FOR 30-FACILITY LINEMIX PROBLEM
LINE DOMINANCE: 20 LINES OF LENGTH 15
FLOW DOMINANCE 138 PERCENT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	3	4	2	0	2	7	2	0	5	5	0	4	0	8	1	5	0	1	0	1	7	3	6	0	2	0	0	0	5
2	1	0	0	1	5	0	9	2	0	0	5	4	0	0	0	2	0	0	0	0	0	6	10	0	0	3	6	0	0	5
3	2	1	0	4	0	0	4	0	0	1	0	1	2	0	0	1	0	0	5	0	0	2	0	5	0	5	0	0	5	11
4	3	2	1	0	1	0	6	7	7	0	0	1	7	9	0	0	0	5	0	0	0	0	4	0	11	3	0	0	0	5
5	4	3	2	1	0	5	0	7	0	0	7	1	5	0	2	2	0	0	0	0	2	0	5	0	7	0	2	0	5	0
6	5	4	3	2	1	0	1	0	2	4	3	6	0	4	0	2	11	2	5	2	2	0	0	0	3	5	8	0	0	0
7	1	2	3	4	5	6	0	0	7	4	1	4	10	7	6	0	3	0	0	0	0	0	0	5	0	1	5	0	0	0
8	2	1	2	3	4	5	1	0	5	0	0	0	2	4	6	1	3	0	0	0	2	0	2	0	0	0	1	10	0	0
9	3	2	1	2	3	4	2	1	0	0	7	0	0	1	2	3	0	2	0	0	6	7	0	5	9	0	7	0	0	0
10	4	3	2	1	2	3	3	2	1	0	5	0	0	0	4	10	6	0	0	0	5	2	5	0	1	0	2	7	1	0
11	5	4	3	2	1	2	4	3	2	1	0	0	1	3	0	5	0	5	1	1	0	0	0	1	0	5	0	2	1	2
12	6	5	4	3	2	1	5	4	3	2	1	0	3	5	0	5	0	0	12	0	4	2	3	1	4	5	1	0	0	0
13	2	3	4	5	6	7	1	2	3	4	5	6	0	3	0	5	0	0	0	0	0	0	0	5	2	1	5	0	4	5
14	3	2	3	4	5	6	2	1	2	3	4	5	1	0	0	2	5	0	5	5	6	2	0	6	0	0	0	0	2	9
15	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	0	0	5	6	0	0	0	0	0	1	0	2	0	5	0
16	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	0	0	0	0	0	4	0	0	0	0	6	0	3	5
17	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	0	0	6	0	0	5	1	1	6	0	0	0	0
18	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0	0	0	0	0	5	0	5	5	0	0	0	0
19	3	4	5	6	7	8	2	3	4	5	6	7	1	2	3	4	5	6	0	5	2	1	0	0	6	0	0	0	0	0
20	4	3	4	5	6	7	3	2	3	4	5	6	2	1	2	3	4	5	1	0	0	0	0	0	0	5	2	7	5	0
21	5	4	3	4	5	6	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	4	0	1	1	2	3	1	0	0
22	6	5	4	3	4	5	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	2	9	0	0	6	0	0	0
23	7	6	5	4	3	4	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	1	0	2	3	5	0	10
24	8	7	6	5	4	3	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0	2	0	0	0	0	0
25	4	5	6	7	8	9	3	4	5	6	7	8	2	3	4	5	6	7	1	2	3	4	5	6	0	0	5	5	5	4
26	5	4	5	6	7	8	4	3	4	5	6	7	3	2	3	4	5	6	2	1	2	3	4	5	1	0	4	0	5	0
27	6	5	4	5	6	7	5	4	3	4	5	6	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	0	1	0
28	7	6	5	4	5	6	6	5	4	3	4	5	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	0	0
29	8	7	6	5	4	5	7	6	5	4	3	4	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	2
30	9	8	7	6	5	4	8	7	6	5	4	3	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0

FLOW-DISTANCE MATRIX FOR 40-FACILITY LINEMIX PROBLEM
LINE DOMINANCE: 25 LINES OF LENGTH 20
FLOW DOMINANCE 134 PERCENT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	5	2	0	0	0	1	0	5	9	0	0	0	0	0	3	0	4	2
2	1	0	0	4	0	0	0	6	3	0	6	6	0	0	0	0	7	0	0	0
3	2	1	0	6	0	3	3	0	0	1	0	0	0	0	2	0	1	5	0	0
4	3	2	1	0	0	4	0	3	2	4	0	2	0	0	3	0	4	0	3	5
5	4	3	2	1	0	0	3	0	7	5	0	1	0	0	0	5	0	3	0	0
6	5	4	3	2	1	0	0	0	0	0	2	3	4	0	3	0	5	0	0	4
7	6	5	4	3	2	1	0	0	0	0	0	3	3	0	0	0	6	6	0	0
8	7	6	5	4	3	2	1	0	0	0	0	5	0	0	3	0	0	0	1	0
9	1	2	3	4	5	6	7	8	0	0	0	0	0	0	0	0	0	4	1	5
10	2	1	2	3	4	5	6	7	1	0	0	2	0	0	0	0	0	0	5	3
11	3	2	1	2	3	4	5	6	2	1	0	4	0	0	0	7	0	0	0	0
12	4	3	2	1	2	3	4	5	3	2	1	0	4	0	0	0	0	8	3	1
13	5	4	3	2	1	2	3	4	4	3	2	1	0	3	3	0	3	0	5	0
14	6	5	4	3	2	1	2	3	5	4	3	2	1	0	0	0	11	7	0	3
15	7	6	5	4	3	2	1	2	6	5	4	3	2	1	0	0	0	4	1	0
16	8	7	6	5	4	3	2	1	7	6	5	4	3	2	1	0	0	0	0	0
17	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	0	0	5	15
18	3	2	3	4	5	6	7	8	2	1	2	3	4	5	6	7	1	0	5	0
19	4	3	2	3	4	5	6	7	3	2	1	2	3	4	5	6	2	1	0	0
20	5	4	3	2	3	4	5	6	4	3	2	1	2	3	4	5	3	2	1	0
21	6	5	4	3	2	3	4	5	5	4	3	2	1	2	3	4	4	3	2	1
22	7	6	5	4	3	2	3	4	6	5	4	3	2	1	2	3	5	4	3	2
23	8	7	6	5	4	3	2	3	7	6	5	4	3	2	1	2	6	5	4	3
24	9	8	7	6	5	4	3	2	8	7	6	5	4	3	2	1	7	6	5	4
25	3	4	5	6	7	8	9	10	2	3	4	5	6	7	8	9	1	2	3	4
26	4	3	4	5	6	7	8	9	3	2	3	4	5	6	7	8	2	1	2	3
27	5	4	3	4	5	6	7	8	4	3	2	3	4	5	6	7	3	2	1	2
28	6	5	4	3	4	5	6	7	5	4	3	2	3	4	5	6	4	3	2	1
29	7	6	5	4	3	4	5	6	6	5	4	3	2	3	4	5	5	4	3	2
30	8	7	6	5	4	3	4	5	7	6	5	4	3	2	3	4	6	5	4	3
31	9	8	7	6	5	4	3	4	8	7	6	5	4	3	2	3	7	6	5	4
32	10	9	8	7	6	5	4	3	9	8	7	6	5	4	3	2	8	7	6	5
33	4	5	6	7	8	9	10	11	3	4	5	6	7	8	9	10	2	3	4	5
34	5	4	5	6	7	8	9	10	4	3	4	5	6	7	8	9	3	2	3	4
35	6	5	4	5	6	7	8	9	5	4	3	4	5	6	7	8	4	3	2	3
36	7	6	5	4	5	6	7	8	6	5	4	3	4	5	6	7	5	4	3	2
37	8	7	6	5	4	5	6	7	7	6	5	4	3	4	5	6	6	5	4	3
38	9	8	7	6	5	4	5	6	8	7	6	5	4	3	4	5	7	6	5	4
39	10	9	8	7	6	5	4	5	9	8	7	6	5	4	3	4	8	7	6	5
40	11	10	9	8	7	6	5	4	10	9	8	7	6	5	4	3	9	8	7	6

	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	4	0	0	7	4	0	0	4	3	6	0	1	0	0	3	5	3	0	8	6
2	2	0	0	10	3	1	0	0	0	2	0	6	0	5	0	3	3	4	0	3
3	3	0	2	6	0	3	12	0	4	0	0	3	6	0	0	0	1	0	0	6
4	3	0	7	3	5	0	6	0	0	0	4	4	2	0	3	0	0	0	0	0
5	5	3	0	0	4	2	5	0	1	0	0	0	3	0	0	0	0	0	0	0
6	0	0	0	6	0	6	5	0	1	0	3	5	2	8	0	0	5	3	3	8
7	8	0	0	2	3	2	0	4	0	1	5	0	10	10	2	3	3	0	3	0
8	3	3	0	0	3	0	3	2	4	3	0	3	5	3	0	0	0	0	4	0
9	0	4	3	0	0	3	0	8	0	3	2	0	6	0	0	1	0	3	3	3
10	4	0	3	0	5	0	5	0	6	0	9	6	0	4	0	0	3	6	0	0
11	6	0	6	0	0	0	0	4	0	6	1	5	0	5	0	5	0	0	2	0
12	0	0	5	0	2	5	0	5	4	0	5	0	0	8	0	3	0	3	3	0
13	0	0	0	4	0	0	5	0	0	0	3	0	4	3	0	0	0	0	0	3
14	8	0	0	0	0	4	5	0	0	3	4	0	0	6	4	9	0	3	10	0
15	5	3	0	0	6	0	3	0	3	0	1	0	0	0	0	7	0	3	0	0
16	0	0	3	0	2	3	0	3	1	0	0	0	0	0	3	5	0	3	2	1
17	5	4	2	0	5	0	0	3	6	0	1	3	0	0	0	0	2	0	5	3
18	0	12	0	3	5	0	0	5	5	4	0	3	0	0	9	8	0	1	0	3
19	0	3	3	3	0	3	0	0	3	0	0	3	0	0	3	0	0	1	4	6
20	10	0	0	3	3	0	0	4	4	0	5	0	4	0	0	0	0	3	0	2
21	0	3	3	6	3	3	0	0	0	0	2	0	0	0	4	7	3	0	1	0
22	1	0	3	4	0	0	4	0	3	3	0	0	0	3	0	8	2	0	2	3
23	2	1	0	4	8	3	0	3	0	0	0	9	0	3	0	0	1	1	0	3
24	3	2	1	0	0	8	3	5	0	3	0	10	0	11	2	0	4	3	3	3
25	5	6	7	8	0	4	3	0	0	3	3	6	3	1	2	0	3	1	3	0
26	4	5	6	7	1	0	0	0	0	5	7	0	8	0	4	4	8	0	0	0
27	3	4	5	6	2	1	0	10	3	0	0	0	0	3	0	0	0	0	5	0
28	2	3	4	5	3	2	1	0	0	5	3	0	0	0	3	0	0	0	0	3
29	1	2	3	4	4	3	2	1	0	0	0	3	3	0	0	1	5	3	0	0
30	2	1	2	3	5	4	3	2	1	0	5	0	0	0	0	0	4	0	0	5
31	3	2	1	2	6	5	4	3	2	1	0	4	0	10	2	0	3	0	5	3
32	4	3	2	1	7	6	5	4	3	2	1	0	2	0	3	0	2	0	0	8
33	6	7	8	9	1	2	3	4	5	6	7	8	0	5	4	0	0	0	0	1
34	5	6	7	8	2	1	2	3	4	5	6	7	1	0	0	6	3	3	6	0
35	4	5	6	7	3	2	1	2	3	4	5	6	2	1	0	2	3	4	0	0
36	3	4	5	6	4	3	2	1	2	3	4	5	3	2	1	0	0	3	0	0
37	2	3	4	5	5	4	3	2	1	2	3	4	4	3	2	1	0	3	0	10
38	3	2	3	4	6	5	4	3	2	1	2	3	5	4	3	2	1	0	0	0
39	4	3	2	3	7	6	5	4	3	2	1	2	6	5	4	3	2	1	0	11
40	5	4	3	2	8	7	6	5	4	3	2	1	7	6	5	4	3	2	1	0

APPENDIX E

FLOW-DISTANCE MATRICES FOR FIVE PROBLEMS WITH POISSON FLOWS, INTERMEDIATE FLOW DOMINANCE, AND VARYING PROBLEM SIZE

FLOW-DISTANCE MATRIX FOR 12-FACILITY
PROBLEM WITH POISSON FLOWS
FLOW DOMINANCE 137 PERCENT

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1	1	0	1	1	2	1	0	0	0	0
2	1	0	0	1	2	0	0	0	1	1	0	1
3	2	1	0	1	0	1	0	2	0	1	0	1
4	3	2	1	0	0	1	0	0	2	0	1	2
5	1	2	3	4	0	1	0	0	0	0	1	0
6	2	1	2	3	1	0	0	0	0	0	0	0
7	3	2	1	2	2	1	0	1	0	0	1	0
8	4	3	2	1	3	2	1	0	0	0	2	1
9	2	3	4	5	1	2	3	4	0	1	0	0
10	3	2	3	4	2	1	2	3	1	0	0	0
11	4	3	2	3	3	2	1	2	2	1	0	1
12	5	4	3	2	4	3	2	1	3	2	1	0

FLOW-DISTANCE MATRIX FOR 15-FACILITY
PROBLEM WITH POISSON FLOWS
FLOW DOMINANCE 137 PERCENT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	0	1	0	2	1	1	0	1	4	0	0	0
2	1	0	0	0	0	0	0	1	2	2	1	0	1	1	0
3	2	1	0	0	1	1	1	3	1	0	1	0	1	1	0
4	3	2	1	0	0	0	1	0	0	1	1	1	0	1	0
5	1	2	3	4	0	0	1	1	2	1	1	0	0	0	1
6	2	1	2	3	1	0	1	0	0	0	1	1	0	1	1
7	3	2	1	2	2	1	0	0	0	1	0	0	0	0	1
8	4	3	2	1	3	2	1	0	0	0	0	0	1	0	0
9	2	3	4	5	1	2	3	4	0	1	0	1	0	0	0
10	3	2	3	4	2	1	2	3	1	0	1	0	0	1	1
11	4	3	2	3	3	2	1	2	2	1	0	0	1	1	2
12	5	4	3	2	4	3	2	1	3	2	1	0	1	0	0
13	3	4	5	6	2	3	4	5	1	2	3	4	0	0	0
14	4	3	4	5	3	2	3	4	2	1	2	3	1	0	0
15	5	4	3	4	4	3	2	3	3	2	1	2	2	1	0

FLOW-DISTANCE MATRIX FOR 20-FACILITY PROBLEM
WITH POISSON FLOWS
FLOW DOMINANCE 141 PERCENT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	1	0	0	0	0	1	1	1	1	1	0	1	0	0	0	0	1	0	1
2	1	0	0	1	0	0	1	1	0	1	1	0	2	1	2	0	1	0	0	0
3	2	1	0	1	1	0	0	1	0	0	0	1	0	1	0	0	0	0	2	0
4	3	2	1	0	1	0	2	0	1	0	1	0	0	1	0	0	0	0	0	2
5	4	3	2	1	0	2	1	1	0	1	0	0	0	0	0	1	0	0	0	0
6	1	2	3	4	5	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0
7	2	1	2	3	4	1	0	0	0	1	0	2	0	1	1	1	2	0	1	1
8	3	2	1	2	3	2	1	0	0	0	0	0	0	1	0	0	0	0	0	3
9	4	3	2	1	2	3	2	1	0	1	0	2	0	1	1	0	0	0	0	2
10	5	4	3	2	1	4	3	2	1	0	0	0	1	0	1	1	1	1	0	0
11	2	3	4	5	6	1	2	3	4	5	0	1	0	1	0	0	1	0	1	0
12	3	2	3	4	5	2	1	2	3	4	1	0	0	2	0	0	0	0	0	2
13	4	3	2	3	4	3	2	1	2	3	2	1	0	2	0	0	1	0	2	1
14	5	4	3	2	3	4	3	2	1	2	3	2	1	0	0	1	2	0	0	2
15	6	5	4	3	2	5	4	3	2	1	4	3	2	1	0	1	0	0	0	1
16	3	4	5	6	7	2	3	4	5	6	1	2	3	4	5	0	1	1	0	0
17	4	3	4	5	6	3	2	3	4	5	2	1	2	3	4	1	0	0	0	0
18	5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	2	1	0	0	0
19	6	5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	2	1	0	1
20	7	6	5	4	3	6	5	4	3	2	5	4	3	2	1	4	3	2	1	0

FLOW-DISTANCE MATRIX FOR 20-FACILITY PROBLEM
WITH POISSON FLOWS
FLOW DOMINANCE 136 PERCENT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	1	0	0	0	1	0	0	1	1	0	1	1	2	1	0	0	0	0
2	1	0	0	1	2	0	0	0	1	1	0	1	1	0	1	0	2	0	1	0
3	2	1	0	1	0	1	0	0	2	0	1	2	1	0	0	0	0	1	0	0
4	3	2	1	0	0	0	0	0	0	1	0	0	1	0	0	0	2	1	1	0
5	4	3	2	1	0	0	0	0	1	1	1	0	1	0	0	0	0	1	2	0
6	1	2	3	4	5	0	0	0	3	1	0	0	0	0	0	1	0	1	0	0
7	2	1	2	3	4	1	0	1	0	2	1	0	0	0	0	1	0	1	0	0
8	3	2	1	2	3	2	1	0	2	1	1	0	0	1	1	0	0	0	0	0
9	4	3	2	1	2	3	2	1	0	1	0	0	1	1	0	0	0	0	0	1
10	5	4	3	2	1	4	3	2	1	0	0	0	0	1	1	0	0	1	0	1
11	2	3	4	5	6	1	2	3	4	5	0	2	1	2	0	1	0	0	0	1
12	3	2	3	4	5	2	1	2	3	4	1	0	1	1	0	1	0	0	0	0
13	4	3	2	3	4	3	2	1	2	3	2	1	0	0	1	0	0	1	0	1
14	5	4	3	2	3	4	3	2	1	2	3	2	1	0	0	1	0	0	1	0
15	6	5	4	3	2	5	4	3	2	1	4	3	2	1	0	2	1	1	0	2
16	3	4	5	6	7	2	3	4	5	6	1	2	3	4	5	0	0	1	0	1
17	4	3	4	5	6	3	2	3	4	5	2	1	2	3	4	1	0	1	0	1
18	5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	2	1	0	1	0
19	6	5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	2	1	0	0
20	7	6	5	4	3	6	5	4	3	2	5	4	3	2	1	4	3	2	1	0

FLOW-DISTANCE MATRIX FOR 30-FACILITY PROBLEM WITH POISSON FLOWS
FLOW DOMINANCE 144 PERCENT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	0	0	0	1	1	1	0	0	0	1	0	0	1	0	0	1	2	2	1	1	1	1	0	1	0	0	0	1	0
2	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	2	0	1	0	0	0	1	0	1	0
3	2	1	0	0	1	0	0	1	1	1	0	1	1	0	0	0	0	1	1	2	2	1	0	0	0	1	1	1	0	0
4	3	2	1	0	0	0	1	1	0	1	0	0	1	2	0	0	1	0	2	0	2	1	0	0	1	0	0	0	1	0
5	4	3	2	1	0	2	1	1	0	1	3	0	0	0	0	0	0	0	0	1	1	2	1	0	1	1	0	0	0	0
6	5	4	3	2	1	0	0	2	1	0	0	0	1	1	0	0	0	1	0	0	1	0	1	0	1	0	0	1	1	2
7	1	2	3	4	5	6	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0
8	2	1	2	3	4	5	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1	1	0	1	1	1
9	3	2	1	2	3	4	2	1	0	1	0	0	0	0	0	1	2	0	0	0	1	1	0	1	0	0	1	0	2	0
10	4	3	2	1	2	3	3	2	1	0	1	0	1	0	1	0	0	1	0	0	2	1	0	0	0	0	0	0	0	0
11	5	4	3	2	1	2	4	3	2	1	0	0	0	0	0	1	0	0	1	0	0	0	2	0	1	0	0	0	0	1
12	6	5	4	3	2	1	5	4	3	2	1	0	1	1	0	1	0	0	0	0	1	1	0	0	0	2	1	0	0	0
13	2	3	4	5	6	7	1	2	3	4	5	6	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	0
14	3	2	3	4	5	6	2	1	2	3	4	5	1	0	1	0	0	2	1	1	0	0	0	1	0	0	0	0	0	0
15	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	1	0
16	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	0	0	1	0	1	2	1	2	0	0	0	0	0	1
17	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	1	1	0	1	0	0	0	0	0	1	0	0	1
18	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0	0	0	0	1	0	0	1	0	1	1	1	0
19	3	4	5	6	7	8	2	3	4	5	6	7	1	2	3	4	5	6	0	2	0	1	0	1	1	0	1	0	0	0
20	4	3	4	5	6	7	3	2	3	4	5	6	2	1	2	3	4	5	1	0	0	0	0	0	0	0	1	1	1	1
21	5	4	3	4	5	6	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	0	1	0	0	0	0	1	0	0
22	6	5	4	3	4	5	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	0	1	0	0	0	1	0	0
23	7	6	5	4	3	4	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	1	0	2	0	2	0	1
24	8	7	6	5	4	3	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0	0	0	0	1	1	0
25	4	5	6	7	8	9	3	4	5	6	7	8	2	3	4	5	6	7	1	2	3	4	5	6	0	0	1	0	0	0
26	5	4	5	6	7	8	4	3	4	5	6	7	3	2	3	4	5	6	2	1	2	3	4	5	1	0	1	0	0	0
27	6	5	4	5	6	7	5	4	3	4	5	6	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	0	0	0
28	7	6	5	4	5	6	6	5	4	3	4	5	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	2	0
29	8	7	6	5	4	5	7	6	5	4	3	4	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	0
30	9	8	7	6	5	4	8	7	6	5	4	3	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0

BIBLIOGRAPHY

Apple, J.M. and M.P. Deisenroth. "A Computerized Plant Layout Analysis and Evaluation Technique (PLANET)." Proceedings, AIIE, 23rd Annual Conference and Convention, 1972.

Armour, G.C., and E.S. Buffa. "A Heuristic Algorithm and Simulation Approach to Relative Location of Facilities." Management Science, Vol. 9, No.2, Jan. 1963, pp.294-309.

Balinski, M.L., and K. Spielberg. "Methods for Integer Programming." Progress in Operations Research, Vol. III. Ed. Julius S. Aronofsky. New York, John Wiley & Sons, 1969, pp.195-292.

Block, T.E.. "A Note on 'Comparison of Computer Algorithms and Visual Based Methods for Plant Layout' by M. Scriabin and R.C. Vergin." Management Science, Vol. 24, No.2, Oct. 1977, pp.235-237.

________. "On the Complexity of Facilities Layout Problems." Management Science, Vol. 25, No.3, March 1979, pp.280-285.

Buffa, E.S. "On a Paper by Scriabin and Vergin." Management Science, Vol. 23, No. 1, Sept. 1976, p.104.

________, G.C. Armour, and T.E. Vollmann. "Allocating Facilities with CRAFT." Harvard Business Review, Vol. 42, No.2, 1964, pp.136-159.

Coleman, D.R.. "Plant Layout: Computers versus Humans." Management Science, Vol. 24, No.1, Sept. 1977, pp.107-112.

Conway, R.W. and W.L. Maxwell. "A Note on the Assignment of Facility Location." The Journal of Industrial Engineering, Vol. 12, No.1, Jan.-Feb. 1961, pp.34-36.

Denholm, D.H., and G.H. Brooks. "A Comparison of Three Computer Assisted Plant Layout Techniques." Proceedings, American Institute of Industrial Engineers, 21st Annual Conference and Convention, Cleveland, 1970, pp.77-84.

Edwards, H.K., B.E. Gillett, and M.E. Hale. "Modular Allocation Technique (MAT)." Management Science, Vol. 17, Nov. 1970, pp. 161-169.

El-Rayah, T.E., and R.H. Hollier. "A Review of Plant Design Techniques." The International Journal of Production Research, Vol. 8, No. 3, 1970, pp.263-279.

Francis, R.L. and J.A. White. Facility Layout and Location: An Analytical Approach. Englewood Cliffs, N.J., Prentice-Hall, 1974.

Gaunt, S.. "A Non-Computer Method Using Search for Resolving the Travelling Salesman Problem." Journal of the Canadian Operational Research Society, Vol. 6, 1968, pp. 44-54.

Gavett, J.W., and N.V. Plyter. "The Optimal Assignment of Facilities to Locations by Branch and Bound." Operations Research, Vol. 14, 1966, pp. 210-232.

Gilmore, P.C.. "Optimal and Sub-Optimal Algorithms for the Quadratic Assignment Problem." Journal of the Society for Industrial and Applied Mathematics, Vol. 10, No. 2, 1962, pp. 305-313.

Grover, K.L.. "An Evaluation of Plant Layout Algorithms." Seattle, University of Washington, unpublished M.B.A. Research Report, 1969.

Hadley, G.. Linear Programming. Reading, Mass., Addison-Wesley, 1962.

Hatch, R.S.. "Bench Marks Comparing Transportation Codes Based on Primal-Simplex and Primal-Dual Algorithms." Operations Research, Vol. 23, No. 6, Nov.-Dec. 1975, pp. 1167-1172.

Hayes, J.R.. Cognitive Psychology: Thinking and Creating. Homewood, Ill., Dorsey Press, 1978.

Hillier, F.S.. "Quantitative Tools for Plant Layout." The Journal of Industrial Engineering, Vol. 14, No. 1, Jan.-Feb. 1963, pp.33-40.

________, and M.M. Connors. "Quadratic Assignment Problem Algorithms and the Location of Indivisible Facilities." Management Science, Vol. 13, No. 1, Sept. 1966, pp.42-57.

Hitchings, G.G., and M. Cottam. "An Efficient Heuristic Procedure for Solving the Layout Design Problem." OMEGA, The International Journal of Management Science, Vol. 4, No. 2, 1976, pp.205-214.

James, M.L., G.M. Smith, and J.C. Wolford. Applied Numerical Methods for Digital Computation with FORTRAN. Scranton, Penn., International Textbook Co., 1967.

Jennings, P.R.. "The Second World Computer Chess Championships." Byte, Jan. 1978, pp.108-118.

Khalil, T.M.. "Facilities Relative Allocation Technique (FRAT)." International Journal of Production Research, Vol. 11, No. 2, 1973, pp.183-194.

Koopmans, T.C., and M. Beckmann. "Assignment Problems and the Location of Economic Activities." Econometrica, Vol. 25, No. 1, 1957, pp.53-76.

Kruskal, J.B.. "Multidimensional Scaling by Optimizing Goodness of Fit to a Non-metric Hypothesis." Psychometrika, Vol. 29, No. 1, March 1964, pp.1-27.

________. "Non-metric Multidimensional Scaling: A Numerical Method." Psychometrika, Vol. 29, No. 2, June 1964, pp.115-129.

Lawler, E.L.. "The Quadratic Assignment Problem." Management Science, Vol. 9, 1963, pp.586-599.

Lee, R.C., and J.M. Moore. "CORELAP--Computerized Relationship Layout Planning." Journal of Industrial Engineering, Vol. 18, No. 3, 1967, pp.195-200.

Maital, S.. "Multidimensional Scaling: Some Econometric Applications." Journal of Econometrics, Vol. 8, 1978, pp. 33-46.

Mojena, R., T.E. Vollmann, and Y. Okamoto. "On Predicting Computational Time of a Branch and Bound Algorithm for the Assignment of Facilities." Decision Science, Vol. 7, 1976, pp.856-867.

Moore, J.M.. Plant Layout and Design. New York, Macmillan Co., 1962.

Muther, R.. Practical Plant Layout. New York, McGraw-Hill, 1955.

________. Systematic Layout Planning. Boston, Industrial Education Institute, 1961.

________. Systematic Materials Handling. Boston, Industrial Education Institute, 1969.

Nugent, C.E., T.E. Vollmann, and J. Ruml. "An Experimental Comparison of Techniques for the Assignment of Facilities to Locations." Operations Research, Vol. 16 No. 1, Jan. 1968, pp.150-173.

Ritzman, L.P.. "The Efficiency of Computer Algorithms for Plant Layout." Management Science, Vol. 18, No. 5, Jan. 1972, Part I, pp.240-248.

________. "The Efficiency of Computer Algorithms for Plant Layout." Unpublished doctoral dissertation, Michigan State University, 1968.

Rushton, P.G. and T.A. Marsland. "Current Chess Programs: A Summary of their Potential and Limitations." Canadian Journal of Operational Research and Information Processing, Vol. II, 1973, pp.13-20.

Scriabin, M., and R.C. Vergin. "Comparison of Computer Algorithms and Visual Based Methods for Plant Layout." Management Science, Vol. 22, No. 2, Oct. 1975, pp.172-181.

________. "Computer and Visual Methods for Plant Layout--A Rejoinder." Management Science, Vol. 23, No. 1, Sept. 1976, p. 105.

Seehof, J.M., and W.O. Evans. "Automated Layout Design Program." The Journal of Industrial Engineering, Vol. 18, No. 12, 1967, pp.690-695.

Shepard, R.N.. "The Analysis of Proximities: Multi-dimensional Scaling with an Unknown Distance Function I." Psychometrika, Vol. 27, No. 2, June 1962, pp.125-140.

________. "The Analysis of Proximities: Multidimensional Scaling with an Unknown Distance Function II." Psychometrika, Vol. 27, No. 3, Sept. 1962, pp.219-246.

________, A.K. Romney, and S.B. Nerlove. Multidimensional Scaling: Theory and Applications in the Behavioral Sciences. Seminar Press, New York, 1972.

Tompkins, C.B. and W.L. Wilson, Jr.. Elementary Numerical Analysis. Englewood Cliffs, New Jersey, Prentice-Hall, 1969.

Torgerson, W.S. . "Multidimensional Scaling: I. Theory and Method." Psychometrika, Vol. 17, 1952, pp.401-419.

________. "Multidimensional Scaling of Similarity." Psychometrika, Vol. 30, No. 4, Dec. 1965, pp.379-393.

________. Theory and Methods of Scaling. New York, Wiley, 1958.

Vollmann, T.E.. "An Investigation of the Bases for the Relative Location of Facilities." Unpublished doctoral dissertation, University of California, Los Angeles, 1964.

________, C.E. Nugent, and R.L. Zartler. "A Computerized Model for Office Layout." The Journal of Industrial Engineering, Vol. 19, July 1968, pp.321-327.

Wallace, H., G.G. Hitchings, and D.R. Towill. "Parameter Estimation for Distributions Associated with the Facilities Design Problem." International Journal of Production Research, Vol. 14, No. 2, 1976, pp.263-274.

Wimmert, R.J.. "A Quantitative Approach to Equipment Location in Intermittent Manufacturing." Unpublished doctoral dissertation, Purdue University, 1957.

________."A Mathematical Method of Equipment Location." The Journal of Industrial Engineering, Vol. 9, No. 6, Nov.-Dec. 1958, pp.498-505.

Young, F.W.. "A FORTRAN IV Program for Non-metric Multidimensional Scaling." Research Report, the L.L. Thurstone Psychometric Laboratory, University of North Carolina, Chapel Hill, N.C., March 1968.

________, and W.S. Torgerson. "TORSCA, a FORTRAN IV Program for Shepard-Kruskal Multidimensional Scaling Analysis." Behavioral Science, Vol. 12, 1967, p.498.

Young, G. and A.S. Householder. "Discussion of a Set of Points in Terms of their Mutual Distances." Psychometrika, Vol. 3, 1938, pp.19-22.

Zoller, K. and K. Adendorff. "Layout Planning by Computer Simulation." AIIE Transactions, Vol. 4, No. 2, June 1972, pp.116-125.