1985

Cost Allocation Heuristics For Solving The Fixed Charge Problem

Don David Wright

COST ALLOCATION HEURISTICS FOR SOLVING THE FIXED CHARGE PROBLEM

by
Don David Wright

School of Business Administration

Submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy

Faculty of Graduate Studies
The University of Western Ontario
London, Ontario
November 1984

ABSTRACT

The fixed charge problem extends the linear programming problem by incorporating a discontinuity in the objective function at an activity level of zero. The objective function value is zero with an activity level of zero. Any activity level above zero has a finite non-variable component in the objective function plus a component which is proportional to the activity level. Other discontinuities in the objective function can be represented by different reformulations. These formulations are useful in approaching a number of managerial problems in areas such as facility location, production planning or manpower planning.

As fixed charge problems become large, various methods of obtaining the optimal solution have excessive computational requirements. As a result, a number of methods have been developed for obtaining good but not necessarily optimal solutions. These approximate methods are able to solve much larger problems.

Considerable success has been achieved with both optimizing and approximate algorithms for problems with a special structure. However, algorithms, both optimizing and approximate, capable of solving any fixed charge problem have been successful with much smaller problems. With many

problems requiring a general formulation, there is a need for an effective method of solving large general fixed charge problems.

A new approximate solution technique will be introduced which will be based on necessary conditions which must be met by a solution plus quasi-sufficient conditions which will indicate either a good solution or an improvement which can be made. The new technique will use heuristics to incorporate the fixed charges into the objective function through a series of cost allocations.

The new solution technique will be evaluated on a number of large general fixed charge problems including test problems and a wide variety of actual applications. In addition, a comparative analysis is made with alternative solution methods. The results indicate that the new solution technique provides a significant improvement to existing methods for solving large fixed charge problems.

TABLE OF CONTENTS

TABLE OF CONTENTS

CHAPTER 3 - A NEW APPROXIMATE SOLUTION TECHNIQUE FOR LARGE GENERAL FIXED CHARGE PROBLEMS

CHAPTER 4 - RESULTS

CHAPTER 5 - CONCLUSION

LIST OF EXHIBITS

CHAPTER 1

INTRODUCTION

The fixed charge problem as defined by Hirsch and Dantzig [45] is an extension of the linear programming problem to include a fixed component in the objective function whenever a decision variable is strictly greater than zero. Thus, there is a discontinuity in the objective function at an activity level of zero. The objective function value is zero when the activity level is zero while an activity level above zero implies a finite fixed component as well as variable component proportional to the activity level. The Hirsch and Dantzig formulation can be modified to include other discontinuities in the objective function such as economies of scale and volume discounts. These formulations are useful in a wide variety of managerial applications which require models with various discontinuities in the objective function.

By ignoring the fixed charges, a linear programming algorithm can provide a feasible solution to a fixed charge problem. However, the objective function value of such a solution may be poor in relation to the optimal value. A number of techniques have been developed for determining the optimal solution to a fixed charge problem. Due to the combinatorial nature of the fixed charge problem, the computational requirements of optimizing techniques become excessive as the size of the problem increases. Consequently, a number of techniques have been developed which provide good, but not necessarily optimal, solutions to fixed charge problems which are satisfactory for decision making purposes. These approximate methods are able to solve much larger fixed charge problems with less computational effort.

Considerable success has been reported for solution techniques, both optimizing and approximate, which will solve problems with particular structures. These structures are exploited by the techniques to gain computational efficiencies. However, current algorithms which are capable of solving all varieties of fixed charge problems have been successful at consistently generating optimal or even good solutions for problems of, at best, a modest size. Since many problems can not be solved by the specialized solution techniques, there is a need for a technique which is capable of obtaining optimal or good solutions to any large fixed charge problem.

A new solution technique will be introduced for solving all varieties of fixed charge problems including large problems. The new technique will be based first on a number of necessary conditions which must be met before any solution can be considered to be the optimal solution. While these conditions must be met, they are not sufficient to guarantee an optimal solution. Due to the combinatorial nature of the fixed charge problem, truly sufficient conditions which are easy to apply for large problems are difficult to develop. Thus, the new technique will be an approximate solution method obtaining good but not necessarily optimal solutions to any fixed charge problem. The second part of the new technique will introduce a number of quasi-sufficient conditions which, if met, will indicate a good solution which can not be improved with reasonable effort. Both the necessary and quasi-sufficient conditions will be based on an allocation of the fixed charges. Thus, the acronym COAL for COst ALlocation will be used in referring to the new technique. The COAL technique is intended for obtaining good solutions to large problems from all application areas where the fixed charge formulation is appropriate. This focus, all and large, will be maintained in the design process.

The fixed charge problem formulation can be applied to a wide variety of application areas which includes facility location, production planning and manpower planning. Within each area, all the problems would have a similar underlying structure such as the transportation problem. If the problem can be described completely by such an underlying structure, it will be referred to as a specialized fixed charge problem. It is these specialized areas where the successful solution of relatively large problems has been reported. The various optimizing and approximate methods exploit the basic structure to gain computational efficiencies.

If a problem has additional features or requirements in addition to the basic structure, such as a blending requirement, the problem is no longer a specialized fixed charge problem. These problems can only be classified as general fixed charge problems. The various algorithms designed for the structure of a specialized problem can not be used for such fixed charge problems. The techniques capable of providing optimal or even good solutions to general fixed charge problems are limited to much smaller problems. More detailed descriptions of both the problem areas and the general and specialized solution techniques are given in Chapter 2.

Chapter 3 will introduce the conceptual foundation underlying the new COAL technique involving the allocation of the fixed charges. Necessary and quasi-sufficient conditions must be met before a solution can be considered a candidate for the optimal solution or as a good solution. Heuristics will be used to modify the allocations in order to find improved solutions. The conditions and heuristics will be combined into four different algorithms each possibly generating different solutions and requiring different computational effort.

The four algorithms of the COAL technique are evaluated in Chapter 4 using not only test problems from the literature but also several actual applications from the different problem areas. As well as evaluating the COAL techniques on their own, a comparative analysis is made with a number of techniques for solving large general fixed charge problems. The various techniques are evaluated on the same computer system. The actual implementation of alternative algorithms is programmed as accurately and efficiently as possible. The impact of different aspects, such as problem size, which have an effect on the difficulty of solving a fixed charge problem is also investigated.

The solution of fixed charge problems involves two dimensions: the quality of the solution and the resources required to obtain the solution. The trade-off between the two dimensions will be described through the use of an efficiency frontier. Algorithms on the efficiency frontier will obtain a certain quality of solution with the minimum resource requirement. An efficiency frontier will be developed for each application area allowing an evaluation to be made of the different algorithms within the particular area. The consistency of the algorithms across the different areas can also be determined.

The new COAL technique is consistently on the efficiency frontier for all application areas. Other methods which obtain approximate solutions to fixed charge problems do not demonstrate this consistency. For larger problems, the execution times required by the COAL algorithms are considerably less than alternative algorithms which obtain the same quality of solution. In addition, the COAL technique achieves good solutions with relatively modest resource requirements again in all the areas tested.

While the COAL technique is developed with the objective of solving large general fixed charge problems, other possible designs or heuristics could be incorporated for other purposes. Some of these aspects are discussed in Chapter 5. Elements from alternative algorithms could be integrated in the new COAL technique. The COAL technique could be modified to apply to problems other than large general fixed charge problems. The design focus, large and all problem areas, has an impact on the choices made in the development of the four COAL algorithms. Different choices in the design could be made. These include modifications to the heuristics as well as how the heuristics are used in the new COAL algorithms. These aspects provide many possibilities for further productive research.

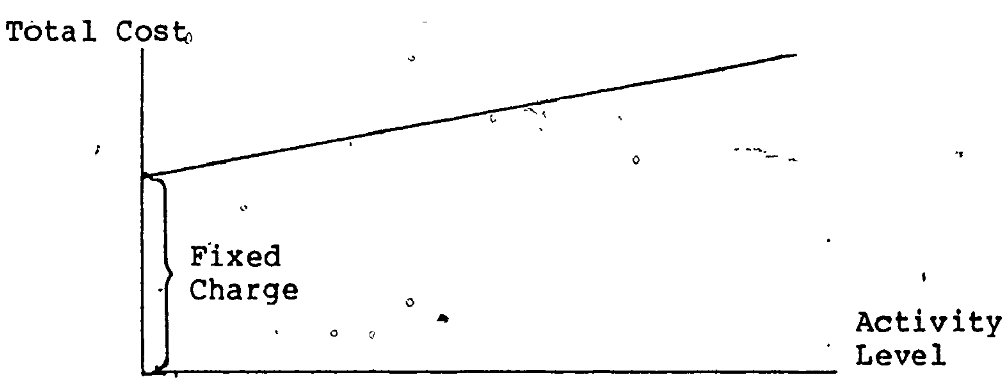
CHAPTER 2

THE FIXED CHARGE PROBLEM

2.1 Overview

A wide variety of decision problems in management can be characterized by the existence of fixed charges. A fixed charge can be defined as a finite non-variable cost associated with an activity level which is greater than zero. Although fixed charges can occur in many problem settings, the fixed charge problem was defined by Hirsch and Dantzig [45] in 1954 as a linear programming problem with fixed charges in the cost structure. Their cost structure is illustrated in Exhibit 2-1 which demonstrates how the total cost due to both a fixed charge and a linear cost varies with the activity level of a variable.

Exhibit 2-1: A Fixed Charge and A Linear Cost



The problem as defined by Hirsch and Dantzig will be referred to as the "general fixed charge problem" which, of course, can be applied to all linear programming problems with fixed charges. In contrast, the characteristics of certain problems may be represented by specialized structures such as the transportation problem. If a problem can be described completely by such a specialized structure with fixed charges, it will be referred to as a "specialized fixed charge problem". A problem which has a specialized structure plus additional features or a mixture of structures can not be classified as a specialized fixed charge problem.

The areas were the fixed charge problem can be applied can be classified by their structure. Problems from the same area will, of course, face similar decisions and difficulties and hence will require the same structure. While not all problems in a particular area will be "pure", there will be a common structural element in these problems.

The approaches to solving fixed charge problems can be classified into those techniques which can handle any fixed charge problem (the general fixed charge problem) and those which are restricted to a particular specialized fixed charge problem. These specialized techniques gain tremendous efficiencies by exploiting various structural features but are more restrictive in their applicability. The latter techniques require that the problem be a specialized fixed charge problem. Problems having a mixture of structures require a solution technique which can handle the general fixed charge problem.

Before discussing the applications and the solution techniques of the fixed charge problem, a mathematical formulation will be presented.

2.2. Mathematical Formulation

The integration of the fixed charges into a linear programming model leads to a non-linear programming formulation. The term, (NLFCP), will be used to refer to the non-linear fixed charge problem formulation which is given in Exhibit 2-2.

Of course, the fixed charge problem can be formulated as a mixed integer programming problem. This formulation of the mixed-integer fixed charge problem or (MIFCP) is very common and is given in Exhibit 2-3. A binary variable, Y_j, is used as a means of representing the discontinuous nature of the fixed charges. To be consistent with (NLFCP), the coefficient, "u", will have to be larger than any upper limit for all X_j. However, most problems using (MIFCP) will replace the "u" in each equation by a "u_j" which will represent the actual upper limit for the respective X_j. Unfortunately, this formulation requires an additional variable, Y_j, and equation for each fixed charge.

The original concept of a fixed charge is demonstrated

Exhibit 2-2: The Non-Linear Fixed Charge Problem

(NLFCP) minimize $z = \sum_j (c_j X_j + f_j \delta(X_j))$

subject to:

$$\sum_j a_{ij} X_j = b_i \qquad \forall\ i$$

$$\delta(X_j) = \begin{cases} 1 \text{ if } X_j > 0 \\ 0 \text{ if } X_j = 0 \end{cases} \qquad \forall\ j$$

$$X_j \geq 0$$

where:

j = index of variable

X_j = activity level of j

c_j = linear cost coefficient of j

f_j = fixed charge of j

---

in Exhibit 2-1 which illustrates how the total cost varies with X_j. This concept of handling fixed charges can be extended to other discontinuities such as economies of scale, minimum threshold level, price breaks and fixed charges at different levels. These extensions are presented in among others Gray [37], Walker [93] and Rousseau [78] and are discussed below.

Economies of scale are illustrated by the solid line in Exhibit 2-4 where a lower unit cost is incurred once a certain threshold level (t_j) is reached. This can be

Exhibit 2-3: The Mixed-Integer Fixed Charge Problem

(MIFCP) minimize $z = \sum_j (c_j X_j + f_j Y_j)$

subject to:

$$\sum_j a_{ij} X_j = b_i \qquad \forall\ i$$

$$X_j - u\ Y_j \leq 0 \qquad \forall\ j$$

$$X_j \geq 0 \qquad \forall\ j$$

$$Y_j = 0,1 \qquad \forall\ j$$

where:

u = a large number

---

transformed into a fixed charge problem by using the relationship $X_j = X_j(1) + X_j(2)$ to represent the variable X_j with economies of scale. As shown in Exhibit 2-4, the variable, $X_j(1)$, has only a linear cost. Variable $X_j(2)$ has a fixed charge and a smaller linear cost representing the economies of scale. Walker [93] shows that no other constraints are required. A technique which obtains the optimal solution will choose the correct variable, either $X_j(1)$ or $X_j(2)$.

A minimum threshold level can also be transformed into a fixed charge. With a minimum threshold, a variable must be greater than a managerially determined minimum level or else zero. This is transformed into a fixed charge in

Exhibit 2-4: Economies of Scale

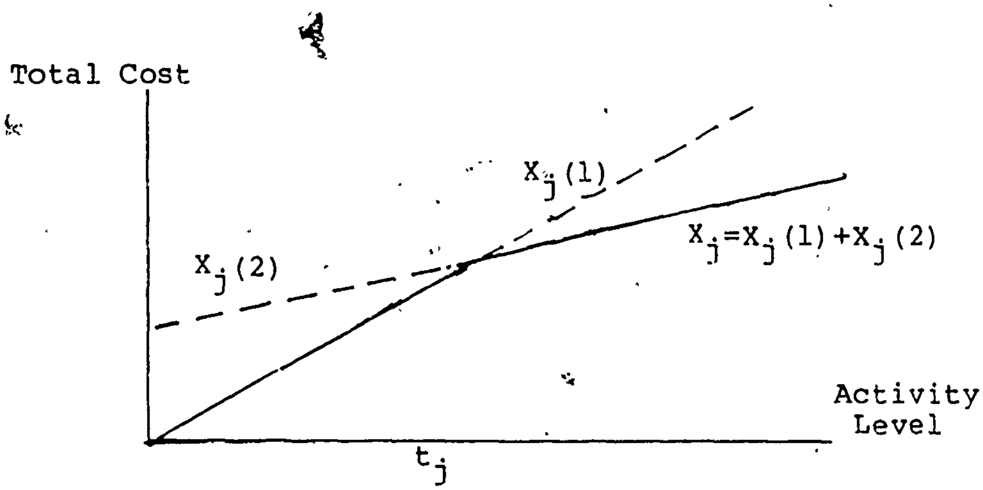
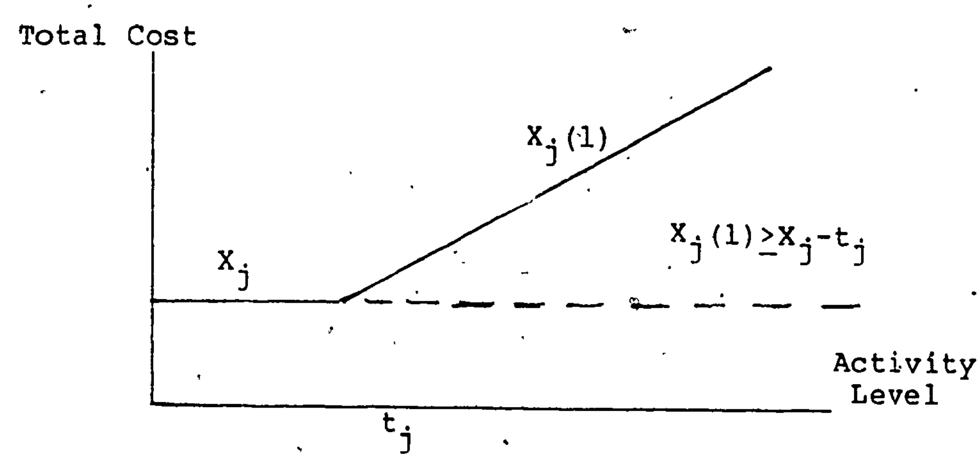


Exhibit 2-5. The variable, X_j, has a fixed charge and no linear cost. The minimum threshold is denoted by t_j. The fixed charge is equal to the actual linear cost times the minimum threshold level. Therefore, once the minimum threshold cost is overcome, there is no additional cost to X_j. However, an additional variable is required to represent the increased costs above the minimum threshold. This variable, $X_j(1)$, will have the linear cost but no fixed charge. The constraint, $X_j(1) \geq X_j - t_j$, will insure that the variable takes on its proper value to account for the incremental cost above the minimum threshold level.

By combining the previous extensions, other discontinuities can be modeled. Price breaks were a lower unit price is charged above a certain level, t_j, would include both economies of scale and minimum threshold level

Exhibit 2-5: Minimum Threshold Level



(Exhibit 2-6). Incorporating fixed charges at different levels is illustrated in Exhibit 2-7.

The Hirsch and Dantzig [45] fixed charge variable illustrated in Exhibit 2-1 assumes one fixed charge associated with one variable. This could be extended to one fixed charge associated with a group of variables by replacing each X_j with the sum of several X_j. These extensions greatly increase the flexibility and applicability of the fixed charge problem.

2.3. Application Areas

The development of linear programming by Dantzig [17] in 1947 and the advances in computer technology fostered the identification and formulation of problems involving fixed charges and their (mathematical) analysis. There exists

Exhibit 2-6: Price Breaks

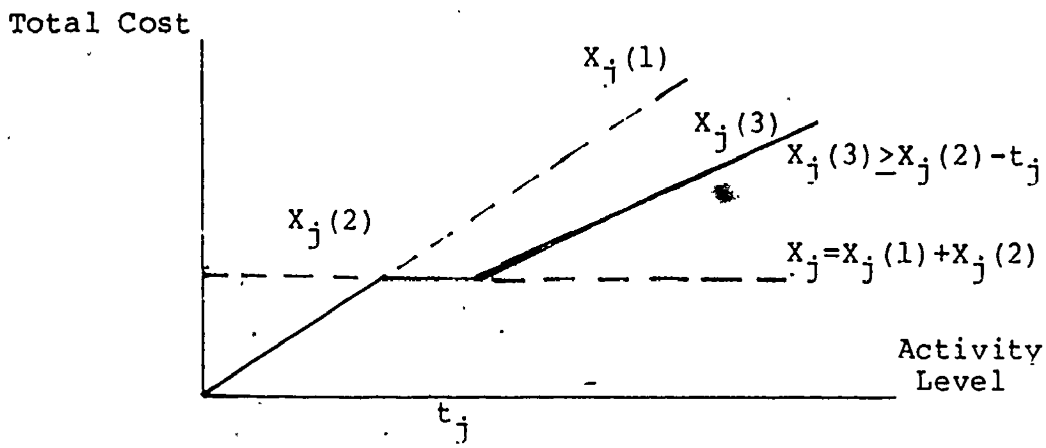
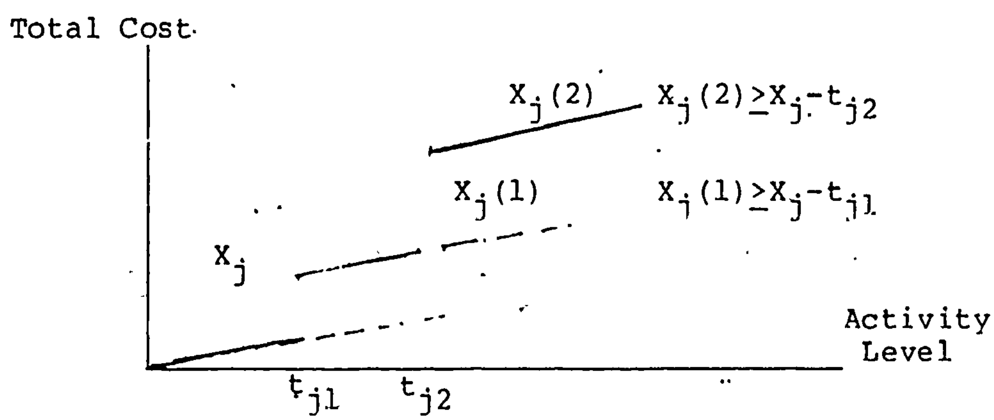


Exhibit 2-7: Fixed Charges at Different Levels



different areas of applications the most important of which are described in the following sections.

2.3.1. Facility Location

The most common area of application for fixed charge problems is in facility location. In this type of problem, facilities may be built or operated on various locations subject to a number of restrictions such as meeting demand or not exceeding supply or capacity. Fixed charges are used to represent such expenses as the initial construction costs, maintenance costs or operating costs which are incurred if the facility is open and operating but are not dependent on the volume processed. In addition, a variable cost component is used for expenses which are proportional to the volume processed by the facility.

The transportation problem forms the structural basis of facility location problems. When a particular problem is a transportation problem with fixed charges, it can be classified as a specialized fixed charge problem. A discussion of the different specialized problems in facility location follows in the next section.

However, the transportation problem structure will appear with additional constraints in other problems. Examples include waste disposal problems discussed by Jenkins [49, 50] as well as Walker, Aquino and Schur [94] with intermediate treatment centers, a coal blending problem by Ravindran and Hanline [77] and the location of power stations by Dutton et. al. [21] with two types of demand. The mathematical formulation differs from problem to problem to account for the particular problem but follow the same basic structure. This basic structure is outlined under the different specialized facility location problems which follow.

2.3.1.1. Capacitated Warehouse Location Problem

The capacitated warehouse location problem or (CWLP) was introduced by Kuehn and Hamburg [54] in 1963 as a new method of formulating and solving problems in distribution systems. They address the problem of locating warehouses throughout the United States. Demand is represented by a number of concentrated centers with a limited number of pre-determined possible warehouse locations. These problems have been widely used as test problems.

(CWLP) is a transportation problem with fixed charges on the warehouses or supply facilities. (CWLP) puts capacity constraints on the size of the facilities, meets a specified demand and minimizes the variable and fixed costs. Each combination of demand and supply centers is represented by an X_{ij} and is referred to as an arc. There is no limit on the capacity of an arc.

Davis and Ray [18] give a formulation for (CWLP) which is presented in Exhibit 2-8 and differs from the formulation used by Kuehn and Hamburger [54]. Francis, McGinnis and White [31] suggest that this formulation for the (CWLP) provides a very efficient solution with a linear relaxation of the binary variables.

Geoffrion [33] in a discussion of distribution systems planning includes several features which are not included in the above formulation. Aspects which can be formulated by extending (CWLP) are several stages of production and distribution, economies of scale, identification of the point of origin and a minimum operating level for a warehouse if opened. Aspects which can not be included as a fixed charge problem and are more difficult to handle include customer service from one facility and limits on the number of sites.

Geoffrion and Graves [34] present a large multicommodity distribution problem which has become a classic application problem in this area. This problem includes 14 plants, 17 product groups, 43 possible distribution centers and 121 customer demand centers. Geoffrion, Graves and Lee [35] report examining considerably larger problems with up to 100 products, 100 sources, 100 distribution centers and 400 customer groups. However, their typical industrial applications are considerably smaller.

Exhibit 2-8: Capacitated Warehouse Location Problem

(CWLP) minimize $z = \sum_i \sum_j c_{ij} X_{ij} + \sum_i f_i Y_i$

subject to:

$$\sum_i X_{ij} = 1 \qquad \forall\ j$$

$$\sum_j d_j X_{ij} \leq q_i \qquad \forall\ i$$

$$0 \leq X_{ij} \leq Y_i \qquad \forall\ i,j$$

$$X_{ij} \geq 0 \qquad \forall\ i,j$$

$$Y_i = 0,1 \qquad \forall\ i$$

where:

i = index of a supply center
j = index of a demand center

X_{ij} = fraction of demand from center j supplied by center i

Y_i = binary variable indicating whether supply center i is open

d_j = demand at center j

q_i = supply available at center i

c_{ij} = total variable cost of supplying demand center j from supply center i

f_i = cost of opening supply center i

2.3.1.2. Uncapacitated Facility Location Problem

The uncapacitated facility location problem or (UFLP) was introduced by Balinski [5] in 1964 as an example of an integer programming problem. This is a special case of the capacitated warehouse location problem involving the removal of the capacity constraint on the supply centers. In this problem, there is no restriction on the capacity of a supply facility once opened nor the capacity of an arc. As a result, the only requirement to meet demand involves insuring that at least one supply facility is open that can handle each demand center. Consequently, the formulation for (UFLP) is the same as (CWLP) with no capacity constraint and is not given separately.

When this problem is solved, all X_{ij} will be either zero or one. This implies that if it is worthwhile to supply any of the demand at j from supply center i, all the demand at j will be supplied from i. Consequently, one does not need to require X_{ij} to be binary although it will be.

A number of algorithms for solving (CWLP), such as Van Roy [89], use a relaxation of the capacity constraints as part of their method. This results in a (UFLP) which is much easier to solve.

Nauss and Markland [71], Stone [84] and Fielitz and White [29], present large lock box location problems as uncapacitated facility location problems. A lock box is a post office box operated by a bank for a corporation or an account with the bank. Payments are either made to or from the box. Charges usually involve a fixed monthly fee and a variable processing fee per check. In addition, interest from the deposits is also considered. When making payments, interest is increased due to the float resulting from the time taken to clear the cheques. When receiving payments, interest is increased by moving the funds to accounts earning higher interest. They discuss problems with up to 112 lock box locations (supply centers) and 400 customer zones (demand centers). Stone [84] indicates that the lock box problem is the most common assignment location problem with over 1300 design studies by various banks for Fortune 1200 corporations during 1977.

2.3.1.3. Fixed Cost Transportation Problem

The fixed cost transportation problem or (FCTP) was also introduced by Balinski [4] in 1961. In (FCTP), the fixed charges are associated with the arcs, or each X_{ij}. This contrasts with (CWLP) where the fixed charges are associated with the opening of a supply facility or warehouse or several X_{ij}. The formulation for (FCTP) is given in Exhibit 2-9.

Ravindran and Hanline [77] include fixed charges in the cost of transporting coal from different mine sites to different coal fired power generating sites in his formulation of the problem. This is in addition to the fixed charges resulting from setting up the coal blending plants which are similar to the fixed charges in the capacitated warehouse location problem.

Jarvis et. al. [48] describe a problem in a wastewater

Exhibit 2-9: Fixed Cost Transportation Problem

(FCTP) minimize $z = \sum_{i}\sum_{j} (c_{ij}X_{ij} + f_{ij}Y_{ij})$

subject to:

$$\sum_{i} X_{ij} = d_j \qquad \forall\, j$$

$$\sum_{j} X_{ij} \leq q_i \qquad \forall\, i$$

$$X_{ij} \leq u_{ij}Y_{ij} \qquad \forall\, i,j$$

$$X_{ij} \geq 0 \qquad \forall\, i,j$$

$$Y_{ij} = 0,1 \qquad \forall\, i,j$$

where:

i = index of a supply center
j = index of a demand center

X_{ij} = amount of demand of center j supplied from center i

Y_{ij} = binary variable indicating that demand from center j can be supplied from center i

d_j = demand at center j

q_i = supply available at center i

c_{ij} = variable cost of supplying demand center j from supply center i

u_{ij} = $\min\{ d_j, q_i \}$

f_{ij} = fixed cost of supplying demand center j from supply center i.

---

system where the sewage lines have economies of scale. Jarvis et. al. incorporate these line costs using piece-wise linear approximations with fixed charges. The resulting model has imbedded fixed cost transportation problems. Stroup [85] uses a fixed cost transportation model to handle the assignment of launch vehicles to space missions.

2.3.2. Production Planning

Production planning problems which can be formulated as fixed charge problems also appear frequently in the literature. Typical production decisions involve the determination of work force level, scheduling of overtime, production run quantities and their sequencing subject to some capacity constraints. A production run typically involves a set up process which incurs some cost. These problems are commonly referred to as lot size problems.

The main structural component in these problems is an inventory balance over a number of periods. The inventory balance insures that all the material in the opening inventory in a period plus production is used to meet demand or goes into the ending inventory.

For many problems, capacity constraints are also included. With capacity constraints, the problem is referred to as the capacitated lot size problem or (CLSP). However, (CLSP) typically includes a factor for down time during the set up operation which has an impact on the capacity. If this down time is included, the problem is no longer a fixed charge problem. Therefore, we will restrict our discussion to problems where the down time associated the set up is negligible which will be referred to as the fixed charge lot size problem.

2.3.2.1. The Fixed Charge Lot Size Problem

The fixed charge lot size problem or (FCLSP) is presented in Exhibit 2-10. This problem includes a number of product groups and limits on regular and overtime capacity. The costs include overtime production costs, inventory holding costs and set-up costs. Other factors including regular production costs and work force payroll can be easily integrated. Typically, regular production costs are invariant with time and the regular payroll must be paid thus both are constant for the decision period.

Hax and Meal [43] present a multiple plant, multiple product, scheduling problem with seasonal demand. They use this problem to illustrate the development of their hierarchical production planning process which partitions the problem into a number of sub-problems. However, the overall problem as well as the sub-problems can be described by (FCLSP).

Hax and Golovin [42] apply hierarchical production planning problem to a problem in the manufacture of automobile tires. Falk [27] applies the hierarchical structure to a large scale continuous flow manufacturing operation at Proctor and Gamble. These two problems result

Exhibit 2-10: Fixed Charge Lot Size Problem

(FCLSP) minimize $z = \sum_t (c_t O_t + \sum_j (h_{jt} I_{jt} + s_{jt} Y_{jt}))$

subject to:

$$P_{jt} + I_{j,t-1} - I_{jt} = v_{jt} \quad \forall\ j,t$$

$$\sum_j w_j P_{jt} - O_t \le r_t \quad \forall\ t$$

$$O_t \le q_t \quad \forall\ t$$

$$P_{jt} - m_{jt} Y_{jt} \le 0 \quad \forall\ j,t$$

$$O_t, P_{jt}, I_{jt}, \ge 0 \quad \forall\ j,t$$

$$Y_{jt} = 0,1 \quad \forall\ j,t$$

where:

j = product group
t = time periods

O_t = overtime worked in period t

I_{jt} = inventory, group j in t

P_{jt} = production, group j in t

Y_{jt} = binary variable for production of group j in period t

c_t = overtime premium in period t

h_{jt} = holding cost, group j period t

s_{jt} = set up cost, group j period t

r_t = regular time limit, period t

q_t = over time limit, period t

w_j = production time required/unit, group j

m_{jt} = maximum production, group j in t

v_{jt} = demand, group j in t

in fixed charge problems which may be formulated by (FCLSP).

Graves [36] maintains the hierarchical structure in his sub-problems which correspond to the Hax and Meal [43] framework. Shadow prices for inventory costs are used as a feedback mechanism to avoid the problems of sub-optimization. Grave's problem can also be formulated by (FCLSP).

Van Wassenhove and de Bodt [91] describe a problem in injection moulding which is a multi-facility capacitated lot size problem with downtime associated with the set up. By a series of approximations, they convert the problem to a single facility capacitated lot size problem with no downtime. The downtime was accounted for by averaging the historical requirements for set up and subtracting this from available capacity.

2.3.2.2. Single Item Capacitated Lot Size Problem

When there is only one type of item in (FCLSP), the problems becomes a single item capacitated lot size problem. There will be only one production variable in a capacity constraint. If the production variable is positive, the set up time must be incurred and subtracted from available capacity. Therefore, the available capacity must be reduced by the setup requirements. If the production level is zero, the capacity utilization will be less than the available capacity minus the set up requirements. Thus, the capacity constraints do not require an integer variable to account for the setup and the problem can be handled by a fixed charge formulation such as (FCLSP) with one product group.

Florian and Klein [30] work with a single item problem with a constant capacity. Love [59] includes an upper limit on the size of the inventory levels. Jagnathan and Rao [47] extend the Florian and Klein problem by including a general cost function. Baker et. al. [3] also extend this problem by looking at varying capacity with time.

2.3.2.3. Uncapacitated Lot Size Problem

When the capacity constraints in (FCLSP) are removed, the problem becomes the uncapacitated lot size problem. The formulation of (ULSP) would be the same as (FCLSP) with no equations for capacity constraints and of course no decision variables for overtime. As an example of an uncapacitated lot size problem, the standard economic order quantity formula which involves a fixed charge set up cost will determine the optimum production quantity for constant demand and lead time. Wagner and Whitin [92] formulate a sequential decision making problem for uncertain demand for the economic lot size problem. Zangwill [97] extends this formulation to include back orders and also presents the problem as a network flow. Blackburn and Millen [9] and Afentakis, Gavish and Karmarkar [1] apply (ULSP) to multi-stage production planning problems associated with material requirements planning. They discuss, in particular, the dramatic increase in problem size when incorporating this problem into MRP.

2.3.3. Manpower Planning

Problems in manpower planning involve complex relationships. These may involve decisions relating to issues such as recruiting, firing, promoting or training. In addition, the members of the organization are typically categorized in homogeneous groups which are then treated as decision variables.

Linear programming provides an effective mechanism for expressing these relationships and their interdependencies which, using a criterion, can be optimized (Price et. al. [75], Edwards [23]). The most common form of normative manpower planning models involve linear programming including goal programming.

There are no standard classes of problems for manpower planning as in facility location or production planning. However, all problems have a set of manpower balance equations similar to the inventory balance equations in production planning. These insure that all individuals are accounted for by remaining in their current group, being promoted or leave the system from one period to the next.

Hax [41] includes a variable work force model as part of aggregate production planning. He states that the most common solution procedure is linear programming which can be extended to a variety of situations. Since the formulation of the fixed charge problem is more flexible than linear programming, it can be applied to manpower planning. Mangiameli and Krajewski [60] examine the policy implications of different workforce strategies by extending a multistage multiproduct lot size problem with setup costs to include variable workforce levels. They compare three different workforce strategies using the optimal solution for each strategy to develop the appropriate cost. The first method evaluated is a "chase" strategy where the workforce level is varied to meet the demand. The second method was "level inflexible" strategy where the workforce remains constant and inventories are used to smooth production. Finally, a "level flexible" strategy where the workforce is shifted from task to task to maintain a constant level was evaluated.

Haehling von Lanzenauer et. al. [38] have used a fixed charge formulation in a problem in the development of manpower planning policies. The hiring and training practices for the sales force of a life insurance are reviewed and a comprehensive plan is developed. There are fixed charges associated with hiring and training of employees. An optimal solution is obtained for the current work force. From this solution, optimal policies are developed to aid decision making in the hiring and training of the sales force.

2.3.4. Other Formulations

Facility location and production planning are the most common areas of application of fixed charge problems cited in the literature. However, the fixed charge problem is by no means restricted to these areas.

2.3.4.1. Accounting

Manes, Park and Jensen [61] use a fixed charge problem formulation for making decisions on internal versus external acquisition of services. The problem involves a company which produces a number of products by different divisions. In addition, there are a number of service divisions which supply intermediate services to the production divisions and each other. These various services could also be supplied by external sources. Their model is an extension of the reciprocal cost problem and encompasses all avoidable costs, variable and non-variable. The variable costs are based on the proportions of service department outputs utilized by the aggregate production of final goods. The non-variable costs relate to different levels of production which may entail different fixed costs.

2.3.4.2. Marketing

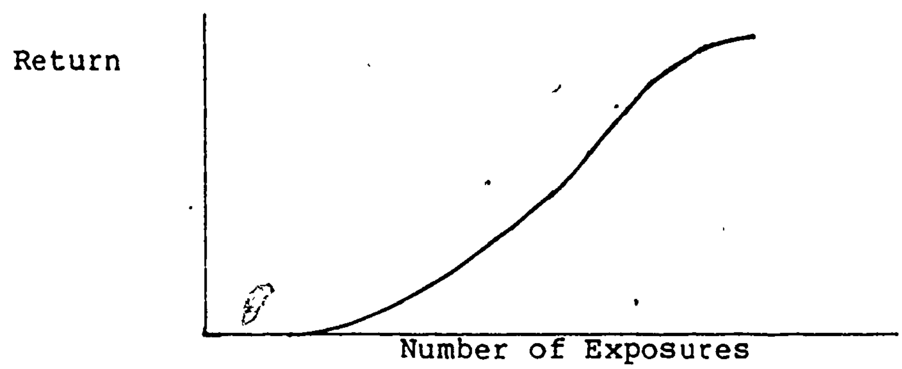
In marketing, linear programming with fixed charges is applied to the distribution problem and the media selection problem (Montgomery and Urban [69]). However, the distribution problem in this case becomes the capacitated warehouse location problem discussed above. This model has been criticized for not properly reflecting the interdependencies between possible distribution centers.

Linear programming has been applied to the media selection problem in advertising. A major criticism has been the linearity assumption required for returns on repeated exposures (Calantone and de Brentani-Todorvic [10]). The diminishing returns, Exhibit 2-11, could be handled as a piecewise linear function with fixed charges by incorporating a minimum threshold and economies of scale. However, problems with return on repeated exposures as well as interdependencies between different media has led to dynamic programming and heuristic solution methods.

2.3.4.3. Portfolio Selection

Sharp's [80] portfolio selection model can use linear programming to select an efficient portfolio which maximizes return subject to a certain amount of risk as measured against a common standard index. Return is obtained from interest or dividends plus capital gains. Of course, in any investment there are costs associated with purchasing and

Exhibit 2-11: Return on Repeated Exposures



selling the security such as commissions which are independent of the period for which a security is held. Transactions costs (fixed charges) greatly increase the difficulty of solving this problem and are usually assumed away. Cooper and Farhangian [14] present a portfolio selection problem with fixed charges. Patel and Subrahmanyam [74] also present a portfolio selection problem with fixed charges based on Markowitz's [63] model which involves quadratic relationships in the objective function to represent a risk factor. Risk is determined by the covariance between the different securities. Hence, there are interdependencies which require quadratic relationships.

2.4. Solution Techniques

Although Hirsch and Dantzig [45] defined the fixed charge problem in 1954, they did not supply a solution technique. Land and Doig [56] in 1956 developed a general purpose solution method for integer programming problems which if extended to mixed-integer programming could also be applied to the fixed charge problem. In addition, solution techniques were being developed for specialized fixed charge problems such as the lot-sizing problem by Wagner and Whitin [92] in 1958, the fixed cost transportation problem by Balinski [4] in 1961, and the capacitated warehouse location problem by Kuehn and Hamburger [54] in 1963. Cooper and Drebes [13] in 1967 introduced the first heuristic for any fixed charge problem.

The classification of solution techniques for the fixed charge problem generally follow the structure of the different problems. Fixed charge problems which require a general linear programming structure require a general solution technique. Problems with well defined structure typically have solution techniques which exploit the structure. Within these two categories are different techniques for obtaining optimal and approximate solutions.

However, all fixed charge problems share the underlying formulation of linear programming. Therefore, many of the properties of linear programming also apply to the fixed charge problem. A brief review of the relevant properties of linear programming follows in the next section.

2.4.1. The Associated Linear Programming Problem

By definition, there is a close relationship between a fixed charge problem and its associated linear programming problem (ALP).

Exhibit 2-12: The Associated Linear Programming Problem

(ALP) minimize $z_c = \sum_j c_j X_j$

subject to:

$$\sum_j a_{ij} X_j = b_i \qquad \forall\, i$$

$$X_j \geq 0 \qquad \forall\, j$$

Both (NLFCP) and (ALP) share the same feasible region. Any linear programming problem is a special case of a fixed charge problem with all the fixed charges being zero. Hirsch and Dantzig [45] have shown that the optimum to the fixed charge problem lies at an extreme point of the feasible region. Of course, the optimum of the associated linear programming problem is also at an extreme point. Many of the properties associated with linear programming problems are applied to the fixed charge problem.

Since (NLFCP) and (ALP) have the same feasible region, then the following can be applied to the fixed charge problem. If (ALP) has no feasible solution, then neither does (NLFCP). Similarly, if the objective function of (ALP) is unbounded, then the objective function of (NLFCP) is also unbounded. Since the problems of infeasibility and unbounded optimum are easily recognized in linear programming, they do not create problems in solving the fixed charge problem. For the following discussions, it will be assumed that the feasible region for the fixed charge problem exists and the optimal solution to the associated linear programming problem is not unbounded.

As both the fixed charge problem and the associated linear programming problem have their optimum at an extreme point within the same feasible region, it is not surprising to see algorithms developed for linear programming incorporated into the algorithms for solving fixed charge problems. In particular, the simplex method, which moves from extreme point to extreme point, forms the basis of most approaches to solving general fixed charge problems.

2.4.2. Solution Techniques for General Fixed Charge Problems

The current approaches to solving fixed charge problems include the use of standard mathematical programming for mixed integer or mixed zero-one problems which can produce optimal solutions. Various enumerative and cutting plane approaches specialized for the fixed charge problem have been developed. Heuristics designed for obtaining "good" but not necessarily optimal solutions have been developed.

2.4.2.1. Optimal Solutions

The standard approach to obtaining an optimal solution to a fixed charge problem is to use one of the commercially available mixed integer programming packages such as MPSX for IBM mainframes. These packages use variations of Land and Doig's [56] original branch and bound algorithm. As a result of it's popularity, this method will be discussed first. Various branch and bound techniques have also been developed specifically for (NLFCP). Cutting planes, both for (MIFCP) and (NLFCP), have also been used to solve fixed charge problems. While branch and bound implicitly enumerates all the possible combinations of fixed charges, vertex generation, the final technique for finding the optimal solution, implicitly enumerates all the vertices of (ALP).

2.4.2.1.1. Branch and Bound for Mixed Integer Fixed Charge Problems

The fixed charge problem is formulated as a mixed integer problem in (MIFCP). This formulation could be solved using various commercially available packages.

Unfortunately, there is an additional equation added for each fixed charge variable. These equations are used to account for the fixed charge with a binary variable and may also be used to impose an upper limit on a variable. Although generalized upper bounding can be used to constrain X_j to its upper limit, Y_j must still be related to X_j to account for the fixed charge.

The commercial mixed-integer mathematical programming codes use variations of the basic Land and Doig [56] branch and bound approach. These algorithms partition the problem into two sub-problems by the constraints $Y_j \leq 0$ and $Y_j \geq 1$.

The essence of branch and bound involves the calculation of a lower bound on each branch which enables the tree to be fathomed when the lower bound exceeds the upper bound for minimization problems. The lower bound is obtained by a linear relaxation of the binary Y_j variables. Therefore, the size of any fixed charge incurred by this relaxation is proportional to the ratio X_j/u_j. A "good" estimate on the upper limit for each variable will help these programs evaluate alternative solutions and speed up the algorithm.

Generally, the commercial codes which are not designed specifically for fixed charge problems may take an inordinate amount of computer time to solve even moderately sized problems. Mangiameli and Krajewski [60] took 34 minutes before stopping MPSX without proving the final solution optimal. This problem has 100 binary variables. The problem from Haehling von Lanzenauer et. al. [38] with 30 binary variables requires 45 minutes on a Cyber 170/173 to solve. These times of course represent the central processor time. Actual elapsed time would be several hours to allow the disk operations required to proceed. Clearly, the general mixed integer branch and bound technique while quite flexible has difficulty with large problems.

2.4.2.1.2. Branch and Bound for the Non-Linear Fixed Charge Problem

Steinberg [82] presents a modification to the Land-Doig branch and bound approach used in most mixed-integer programming codes. Although both approaches use the same tree structure with the same nodes, Steinberg's approach solves (ALP) which is smaller than the corresponding version of (MIFCP) with continuous and binary variables. At each node, the appropriate X_j is either constrained to zero or allowed to be greater than zero. Therefore, there is no need for Y_j and the large number of equations relating Y_j to X_j.

Steinberg estimates a lower bound for each branch as the sum of the lower bound on the continuous portion of the fixed charge problem plus the lower bound on the fixed charge portion of the fixed charge problem. The lower bound for the continuous portion is a straight forward linear programming problem from (ALP). However, the fixed charge portion, (FCPF) (Exhibit 2-13), is as difficult to solve as the original problem. Steinberg estimates a lower bound for the fixed costs by determining which fixed charge variables could be driven to zero. The fixed charge variables which can not be driven to zero represent a minimum value for the fixed charge portion.

Exhibit 2-13: The Fixed Charge Portion
- Fixed Charge Problem

(FCPF) minimize $z_f = \sum_j f_j Y_j$

subject to:

$$\sum_j a_{ij} X_j = b_i \qquad \forall\ i$$

$$X_j - u\, Y_j \le 0 \qquad \forall\ j$$

$$X_j \ge 0 \qquad \forall\ j$$

$$Y_j = 0,1 \qquad \forall\ j$$

McKeown[66, 67] has a similar approach in his branch and bound algorithm for the fixed charge problem. He has the same tree structure and the lower bound is estimated from the lower bound of the continuous problem plus the lower bound for the fixed charge portion of the problem. His bounding mechanism for the fixed charge portion is given by (FCPM) in Exhibit 2-14. (FCPM) provides a lower bound on the sum of the fixed charges and is a set covering problem which is relatively easy to solve.

Exhibit 2-14: The Fixed Charge Portion - McKeown

(FCPM) minimize $z_f = \sum_j f_j Y_j$

subject to:

$$\sum_j d_{ij} Y_j \geq 1 \qquad \forall\ i$$

$$d_{ij} = \begin{cases} 1 \text{ if } a_{ij} > 0 \\ 0 \text{ if } a_{ij} \leq 0 \end{cases} \qquad \forall\ ij$$

$$Y_j = 0,1 \qquad \forall\ j$$

McKeown and Sinha [68] have extended McKeown's algorithm to more efficiently solve fixed charge problems in which all the coefficients in the constraint matrix have zero or positive coefficients. With this property, (FCPM) provides a better representation of the fixed charge portion.

In addition to the advantage of fewer constraints than Land-Doig, these algorithms exploit the feature that any feasible solution to (ALP) is also a feasible solution to (NLFCP) and (MIFCP). Although the Land-Doig approaches have these feasible solutions as part of the solution to the continuous problem, they usually fail to recognize them as a feasible solution to the mixed-integer problem. Thus, Steinberg and McKeown generate feasible solutions and upper bounds much more quickly than the Land-Doig approach.

Steinberg [83] reports in 1976 that the exact methods appear to be impractical for most moderate or large fixed charge problems. These methods have difficulty if the impact of the fixed costs were significant. Steinberg and Mckeown both use randomly generated problems with five equations and ten fixed charge variables originally created by Cooper and Drebes [13]. Larger problems with up to fifteen equations and 30 fixed charge variables are created by concantenating the smaller problems as in Exhibit 2-15.

Exhibit 2-15: Creation of Large Random Problems

$$A = \begin{vmatrix} A_1 & 0 & 0 \\ 0 & A_2 & 0 \\ 0 & 0 & A_3 \end{vmatrix}$$

McKeown modified these problems by changing the constraints to ">" and setting all the aij positive for an additional set which were also used by McKeown and Sinha

[68]. A final set of strictly "pure" fixed charge problems is created by Mckeown by setting $c_j=0$ for all j. McKeown [67] reports some improvement over Steinberg [82]. However, he confined his testing to these small problems due to computational limitations. As Steinberg [83] notes, the effort required to solve a problem grows disproportionately with the number of fixed charges and the size of the problem. Thus, other approaches to solving large fixed charge problems is necessary.

2.4.2.1.3. Cutting Planes

Since (MIFCP) is a mixed integer programming problem, any of the various cutting plane techniques developed for mixed integer programming can be applied to it. Rousseau [78] develops a cutting plane algorithm specialized for the fixed charge problem. Rousseau's algorithm generates cuts for minimizing a concave function. He also incorporates two partitioning schemes, one similar to Steinberg's branch and bound process. Rousseau's approach is similar to a branch and bound technique by Taha [86] which utilizes cutting planes to identify local minimum of a concave function over a convex polyhedron. Rousseau compares his cutting plane to Benders cutting plane. Although Benders has an advantage of a finite convergence, Rousseau's algorithm has more efficient cuts.

As there is little computational experience, it is difficult to generalize about the effectiveness of cutting planes in solving the general fixed charge problem. Although the algorithm proposed by Taha [86] applies to any concave objective function, he reports results for the fixed charge problems described above. He indicates his algorithm will have problems with degeneracy. Rousseau [78] reports good results with two random problems with ten equations and fifteen fixed charge variables with upper bounds. However, he reports more difficulty with a fixed cost transportation problem with 7 supply centers and 5 demand centers resulting in a problem with 35 fixed charge variables and 11 equations. Rousseau requires the use of a partitioning method in addition to his cutting planes to solve the problem. Rousseau has more difficulty with a capacitated warehouse location problem with 4 demand centers and 7 supply centers. This problem has 30 variables with 7 fixed charges and and 11 equations. Extensive computations could not be carried out for this problem as difficulties with numerical accuracy were encountered. Rousseau [78] suggests using a combination of approximate, enumerative and cutting plane methods for solving fixed charge problems.

2.4.2.1.4. Vertex Generation

Murty [70] and McKeown [65] present methods for solving the fixed charge problem by ranking the extreme points of (ALP). Initially the optimal solution to (ALP) is obtained which provides a solution to (NLFCP) and an upper bound on the objective function. Vertices are generated by pivoting away from the optimal solution to (ALP) and ranked in order of increasing objective function value. A lower bound on the fixed cost portion of (NLFCP) is estimated from the smallest sum of fixed charges required to satisfy the constraints. The algorithm terminates the search for vertices when the continuous objective function plus the lower bound for the fixed costs exceed the best solution to the fixed charge problem obtained so far. These algorithms face the same problem estimating lower bounds for the fixed charge portion of (NLFCP) as is encountered in the branch bound techniques.

The computational results for these algorithms are limited to the small 5x10 random problems generated by Cooper and Drebes [13]. Vertex generation works best when the fixed charge optimum is close to the optimum solution of the associated linear programming problem and the size of the fixed charges is small relative to the continuous charges. A large number of vertices may be required, even for small problems. Unless the bounding mechanism for the fixed charges works well or the fixed charge portion is small, vertex generation appears to be impractical for even moderate sized problems.

2.4.2.2. Approximate Solutions

The difficulties associated with solving large integer programming problems in general and the fixed charge problem in particular have led to the development of a number of methods for obtaining approximate solutions to the fixed charge problem. These solution techniques obtain "good" but not necessarily optimal solutions. A number of authors have given heuristics for solving the fixed charge problem with approximate methods (Balinski [4], Cooper and Drebes [13], Denzler [19], Hiraki [44], Steinberg [82], Walker [93]).

2.4.2.2.1. The Balinski Approximation

The simplest heuristic for solving (NLFCP) was obtained by solving (FCPB) in Exhibit 2-16 (Balinski [4]). (FCPB) is equivalent to the linear relaxation of (MIFCP) and its solution is the same as the initial solution generated by most mixed integer programming algorithms. Consequently, (FCPB) provides a lower bound to (NLFCP). To work at all well, a good estimate for u_j must be available. The Balinski approximation will provide good solutions to the formulation of (CWLP) given in Exhibit 2-8 as well as (FCTP). In both problems, an all integer or nearly all integer solution is often obtained.

Exhibit 2-16: The Fixed Charge Problem
- Balinski Approximation

(FCPB) minimize $z = \sum_j c_j' X_j$

subject to:

$$\sum_j a_{ij} X_j = b_i \qquad \forall\ i$$

$$X_j \geq 0 \qquad \forall\ j$$

where:

$$c_j' = c_j + f_j/u_j \qquad \forall\ j$$

2.4.2.2.2. Adjacent Extreme Point Heuristics

The remaining heuristics for (NLFCP) are termed adjacent extreme point heuristics as they use the simplex method to go from extreme point to extreme point in (ALP) looking for improved solutions. These heuristics generally use a three step procedure. The first step is:

Step 1. Solve (ALP).

Any feasible solution to (ALP) will suffice. All the heuristics use the optimal solution to (ALP) which ignores the fixed charges. Although the Balinski approximation is also suggested, none of the heuristics actually use it.

The appropriate heuristic is applied in Step 2. Cooper and Drebes, Denzler, Steinberg and Walker all obtain improved solutions by checking all extreme points within one pivot from the current solution. This solution is referred to as a fixed charge local optimum.

Step 2. Obtain the fixed charge local optimum.

A fixed charge local optimum is a solution in which all solutions within one pivot are inferior. Step 2 generally improves the solution from Step 1. The initial fixed charge local optimum is always obtained relatively quickly.

After Step 2, a test phase is entered. Step 3 moves away from the solution of Step 2:

Step 3. Move more than one pivot from the current solution. If an improved solution is found return to Step 2.

The algorithms cycle between Steps 2 and 3 until a solution is obtained which can not be improved upon with practical computational effort. Hiraki does not use the relatively fast Step 2 in his heuristic. Rather, he has a one step slow procedure similar to the Step 3 in the Walker heuristics.

When a fixed charge local optimum is reached (all adjacent extreme points have inferior objective function values), the test phase is entered. The test phase tries to move away from the local optimum by inserting variables into the basis in various combinations. Several methods are suggested for bringing in combinations of variables. The differences between the Cooper and Drebes, Denzler, Walker and Steinberg heuristics consist of the different methods of bringing in combinations of variables. The methods include:

1. All combinations with randomly selected variables. (Denzler)
2. All combinations with the non-basic variables which if brought into the basis, increase the objective function the least. (Steinberg)
3. All combinations with the non-basic variables which if brought into the basis, increase the objective function the most. (Steinberg)
4. All combinations of pairs of non-basic variables. (Walker)
5. All combinations of three non-basic variables. (Walker)

The objective behind many of the combinations is to get away from the current local optimum. There is a natural tendency to move directly back to the current local optimum. The number of possible combinations of variables to introduce into the basis may become quite large. For example, if a problem had 100 equations and 250 variables, there would be 150 non-basic variables requiring 22,350 combinations for all possible pairs of non-basic variables. This would include all variables with no fixed charge as well as the fixed charge variables. With many problems, there can be a very large number of variables with no fixed charge. As a result, the test phase can become quite lengthy.

It is assumed in these algorithms that there is no degeneracy. There is no rule analogous to the "rate of greatest improvement" rule in the simplex method in linear programming. If a variable is brought in at the 0 level, there is no way of inferring if there has been an improvement. In order to get around degeneracy, these methods must enter the more time consuming test phase and bring variables into the basis in various combinations. Generally, with large problems, there is more degeneracy.

Cooper and Drebes [13] propose a method that develops surrogates for the reduced costs in an attempt to incorporate the fixed charges into a linear programming formulation. The coefficients in the objective function are modified by:

$$c_j' = \begin{cases} c_j + f_j / X_j & \text{if } X_j > 0 \\ c_j & \text{if } X_j = 0 \end{cases} \quad \forall j \qquad (2.1)$$

The reduced costs are recalculated. Potential candidates for entry into the basis are generated from those variables with negative reduced costs. These candidate variables are brought into the solution and the resulting objective function value is compared with the current best solution. If a new "best" solution is found, the costs are recalculated and the process is repeated. By reducing the number of variables investigated as candidates for entry into the basis, the computational effort is reduced.

Steinberg states that the Denzler, Walker and his own heuristics (which he says differ only in the test phase) provide better solutions in terms of obtaining the optimum than the Cooper and Drebes heuristic. This is logical since the former heuristics consider all the non-basic variables while Cooper and Drebes consider only a subset. Cooper and Drebes advantage lay in a faster computational algorithm.

Obtaining a fixed charge local optimum has several computational advantages particularly when the regular simplex method is used. There is no increase in memory requirements over linear programming. The information required to calculate the objective function value at any adjacent extreme point is available in the current simplex tableau.

Unfortunately, when the revised simplex method is used, more computational effort is required to calculate the objective function value at each adjacent extreme point. With the revised simplex, each column in the current tableau has to be generated by multiplying the inverse of the current basis by the original column in the constraint matrix resulting in an increase in computational effort. The revised simplex method using the product form of the inverse is the most practical method for solving large linear programming problems.

Although obtaining a fixed charge local optimum requires no additional storage over linear programming using the simplex method, the test phases require the storage of information relating to the best solution obtained. If the regular simplex method is used, the complete tableau has to be stored. The computational difficulties involved force the restoration of the tableau of the local optimum as accurately as possible. However, these tableaus could be stored on peripheral devices such as a disk.

The primary set of test problems by the adjacent extreme point heuristics are the sames random problems generated by Cooper and Drebes [13]. Steinberg [82] combines up to ten of these problems creating problems with 100 equations and 150 fixed charge variables. The solutions obtained that were sub-optimal have objective function values that are very close to the optimal solution. In addition, a number of 5x7 and 6x8 fixed cost transportation problems have been used as test problems by Walker [93] and Steinberg [82]. Good results are reported in terms of both obtaining "good" solutions and execution times.

2.4.3. Solution Techniques for Specialized Fixed Charge Problems

In general, the methods outlined under solution to the general fixed charge problem will have difficulties as the size of a problem increases. By taking advantage of special structures, various solution techniques obtain significant efficiencies with respect to solution times. These techniques must be classified by the type of problem as they are very specific.

The techniques which obtain optimal solutions can solve problems considerably larger than an equivalent problem using a solution method for the general fixed charge problem. Methods which obtain approximate solutions to specialized fixed charge problems are used for very large problems.

2.4.3.1. Capacitated Warehouse Location Problem

Several surveys of solution techniques for (CWLP) are given in Geoffrion [33], McGinnis [64], Erlenkotter [26] and Francis et. al. [31]. Sa [79] first developed a branch and bound algorithm to solve (CWLP) which is later refined by Akinc and Khumawala [2]. Their algorithm uses a linear relaxation to convert the problem into a transportation problem. Naus [71] and Christofides and Beasley [11] apply a Lagrangean relaxation to the demand constraint resulting in a knapsack problem. Geoffrion and Graves [34] use Benders decomposition to solve their classic distribution problem. Van Roy [90] uses a Lagrangean relaxation of the a capacity constraint to generate an uncapacitated facility location problem. He uses Erlenkotter's [25] dual based procedure for solving the uncapacitated facility location problem. Van Roy's results for solving the Kuehn and Hamburger [54] test problems are the best reported.

Jacobsen [46] reviews a number of heuristics for (CWLP) with some computational results. The methods evaluated include the ADD heuristic and SHIFT heuristic by Kuehn and Hamburger [54], DROP by Feldman, Lehrer and Ray [28], the Alternate Location Allocation method by Rapp [76] and Cooper [12], the Vertex Substitution Method by Teitz and Bart [88] plus composite heuristics by Sa [79] and Khumawala [53]. Jacobsen also extends the ADD heuristic to an ADD-LO which he reports as performing better than others on problems with tight capacity constraints. As a rule of thumb, he states that exact algorithms can solve medium scale problems (100 locations and 100 customers) while heuristics are required for large scale problems (1000 locations and 1000 customers).

2.4.3.2. Uncapacitated Facility Location

Several algorithms have been developed which solve the uncapacitated problem very quickly. Van Roy and Erlenk[illegible]ter

[90], Naus and Markland [72], and Fielitz and White [29] provide special purpose algorithms which will solve the uncapacitated facility location problem specifically. The authors report solution times of several seconds for problems which would be considered quite large and require lengthy computer runs if they are formulated and solved as a more general facility location problem. Francis et. al. [31] also review the solution methods available for the uncapacitated facility problem and conclude this problem is considerably less difficult than (CWLP).

2.4.3.3. Fixed Cost Transportation Problem

Gray [37] provides the first exact algorithm for solving (FCTP). Subsequent branch and bound approaches were developed by Kennington and Unger [52] and Barr, Glover and Klingman [7]. Barr et. al. review the algorithms available for the fixed cost transportation problem and introduce one of their own. They test these algorithms on large randomly generated sparse networks. The algorithm given by Barr et. al. which uses a very efficient method for solving transportation problems appears to be superior to the others. It solves problems with 3,000 constraints and 1,200 fixed charges in nine seconds on a CDC 6600.

2.4.3.4. Capacitated Lot Size Problem

Since (FCLSP) is a special case of the capacitated lot size problem, (FCLSP) can be solved using an algorithm for (CLSP). Manne [62], Dzielinski and Gomory [22] and Lasden and Terjung [57] solve the capacitated lot size problem by generating a variable for each possible combination of different set ups and treating (CLSP) as a linear problem. This solution, which is typically nearly integer (Manne [62]), is then rounded to a feasible solution. Newson [73] proposes a heuristic which decomposes the problem into an aggregate planning problem and a detailed scheduling problem. He includes the work force as a variable for capacity decisions in the aggregate planning problem. Eisenhut [24] uses a period by period approach with production lots assigned on the basis of a priority index. This index is a measure of the viabililty of producing the lot now or postponing production. Lambrecht and Vanderveken [55] and Dixon and Silver [20] introduce a number of refinements which lead to improved cost performance.

Van Wassenhove and de Bodt [91] apply the capacitated lot size heuristics from Eisenhut [24], Dixon and Silver [20] and Lambrecht and Vanderveken [55] to their problem in injection modeling. He makes four approximations to convert the problem into a fixed charge single machine capacitated lot size problem. The heuristics use little computer time and perform much better than a simple EOQ formula.

2.4.3.5. The Fixed Charge Lot Size Problem

Graves [36] divides the problem (FCLSP) into two sub-problems similar to Newson's [73] approach to (CLSP). Graves develops an aggregate planning model which minimizes overtime costs and a disaggregation sub-problem which are linked with Lagrange multipliers representing the inventory costs. This provides a feedback mechanism between the aggregate model and the sub-problem which are then solved iteratively. Using the production schedule at each iteration, a feasible solution is generated with a simple heuristic which assumes unlimited overtime.

Graves solves a number of problems with 240 binary variables in 90 to 364 seconds (average, 236 seconds) and 480 zero-one variables 147 to 405 seconds (average, 311 seconds) on a PRIME 400. His solutions come within 4.4 per cent of a lower bound on the optimum.

Barany et. al. [6] add a number of additional constraints to make a tighter formulation for (FCLSP). The additional constraints allow a standard mixed-integer programming to be more efficient although the number of constraints is greatly increased. For example, the Graves' problem with a small set up cost was solved to optimality in approximately 200 seconds on a Data General MV8000. The problem has approximately 1100 constraints although a formulation using (FCLSP) would require 252 equations.

However, the problem with a large set up cost required approximately 9,000 seconds to solve to optimality.

Hax [42] suggests that the algorithms presented by Balinski [4], Cooper and Drebes [13], Denzler [19], Rousseau [78] and Steinberg [82] provide effective heuristics for the capacitated lot size problem when the downtime associated with a set up is negligible.

The Hax and Meal [43] hierarchical framework for production planning solves (FCLSP) ignoring the fixed charges in the set up. This provides an aggregate plan over the planning horizon. An advantage of this aggregate plan is the need for forecasts for types of products rather than on a more detailed level. The aggregate plan is disaggregated to account for the fixed charges in the set up and provide a detailed operational plan. The detailed plan is implemented and the process is repeated next period with a rolling planning horizon. This procedure obviously works best when the fixed charges on the set up are relatively small.

Hax and Golovin [42] use a problem similar to Graves with 65 binary variables and considerably smaller set up costs. For the detailed scheduling sub-problem, they apply heuristics from Hax and Meal [43], a Knapsack method from Bitran and Hax [7], Winters [95] method and an Equalization of Run Out Times method. These heuristics provided good solutions to the problems. As Hax and Golovin mention, high fixed charges on the set up will affect the performance of these methods.

2.4.3.6. Uncapacitated Lot Size Problem

The uncapacitated lot size problem, (ULSP), is much easier to solve, both optimally and heuristically, than an equivalent size capacitated lot size problem. Afentakis et. al. [1] review the algorithms available to solve the uncapacitated lot size problem in multi-stage assembly system as well as presenting a new formulation and optimization algorithm. Afentakis et. al. use a branch and bound procedure with a lagrangean relaxation to obtain a shortest route problem. By comparison, Blackburn and Millen [9] evaluate three single level algorithms applied to a multi-stage problem. These are the Minimum-Cost-per-Period from Silver and Meal [81], Part-Period-Cost-Balancing and Wagner and Whitin's algorithm [92]. Blackburn and Millen extend these methods to heuristically handle multiple levels.

Afentakis et. al. [1] solve a number of randomly generated problems with up to 50 items in 15 stages in 18 periods. These require up to forty seconds on an IBM 3033 for the longest solutions.

Blackburn and Millen present a problem with five stages in the production process and a 12 period planning horizon. The structure of such a problem would be similar to (FCLSP) requiring 60 binary variables. Additional constraints are required for the different stages. The constraints on production capacity would be removed and there would be only one product group. Blackburn and Millen use a dynamic programming approach from Crowston and Wagner [16] to obtain the optimal solution requiring 3.3 seconds on a DEC-1099. The various heuristics solve the problem in orders of magnitude less time.

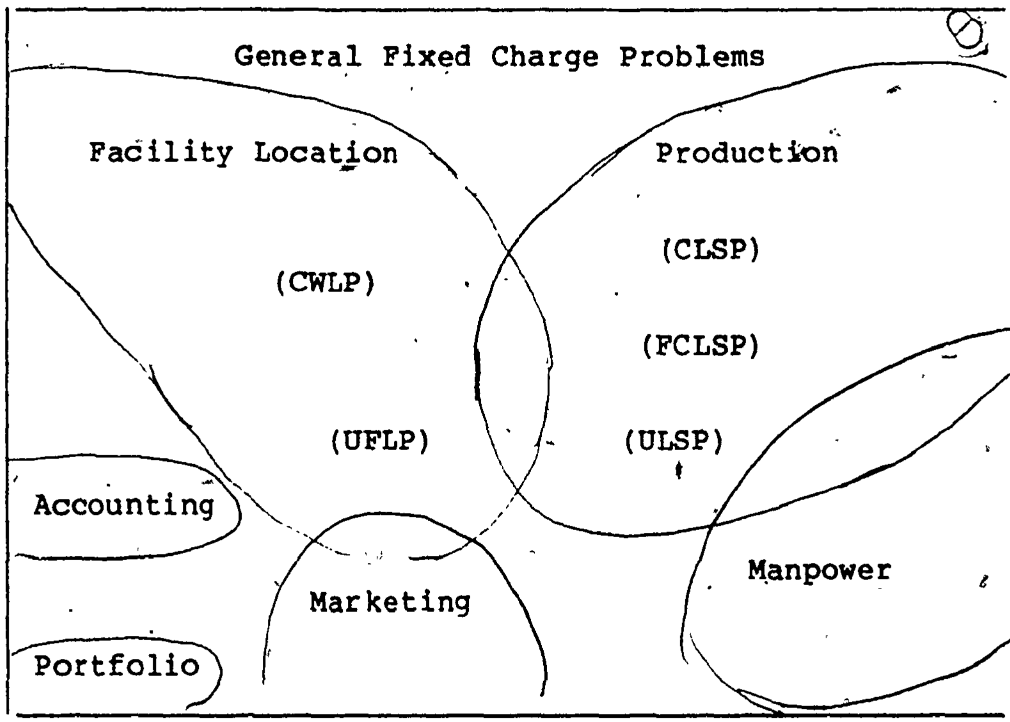
2.5. Summary and Research Objectives

The fixed charge problem can be applied to a very general set of problems which have a linear programming structure but also contain a significant fixed cost components. This cost structure can be applied to a number of related costs such as economies of scale, price breaks or minimum levels of production. This provides more flexibility than is first apparent.

The fixed charge problems are classified by application area which include facility location, production planning, manpower planning, media selection or portfolio selection. Within these application areas, the problems are further specialized by different characteristics. Therefore the fixed charge problem, as a general problem, can be broken down into many different specialized areas (Exhibit 2-17).

These specialized areas have been developed for basically two reasons: their structures adequately

Exhibit 2-17: Classification of Fixed Charge Problems



represent a particular problem; and solution techniques are available which will solve larger specialized problems. Solution techniques which solve the general fixed charge problem can also be used for the specialized structures. However, these general techniques have difficulty as problems get large.

The relationship between large and difficult is not straight forward. While ordinary continuous variables may contribute to the size of the problem, they have much less impact on the difficulty. As well, the size of the fixed charge component in the value of the objective function is important (Kennington [52] and McGinnis [64]). Francis, McGinnis and White [31] state that (CWLP) with very small and very large fixed charges are easier to solve than those in between. There is a rather broad relationship between difficulty, number of fixed charges, size of the fixed charge component and the number of continuous variables. While the number of fixed charge variables is the most important factor in determining difficulty, the other factors are not insignificant. A solution technique will have to be able to solve problems of a significant size and difficulty within reasonable limits in order to be effective.

2.5.1. Evaluation of Solution Techniques

The solution techniques for general fixed charge problems appear to have difficulty as problem size increases.

The general purpose branch and bound mixed integer programming techniques are very flexible and find the optimal solutions to small problems well. However, as the problem size increases, they become impractical. They are hampered by the fact that they are not designed specifically for the fixed charge problem.

Branch and bound techniques for the general fixed charge problem are extensions of similar techniques from mixed integer programming. There is an absence in the literature of test results for larger problems. Consequently, it would appear that these techniques suffer similar problems with size increases as most branch and bound methods. This would also apply to the various cutting plane and vertex generation methods discussed.

The approximate methods developed for the general fixed charge problem are intended, in particular, for large fixed charge problems. However, the results so far have not proven their effectiveness. The test problems used have been limited to the fixed cost transportation problems and the concantenated random test problems. There could be some question as to the difficulty of the latter problem. Jenkins [49] reports that the Walker algorithm requires lengthy computer runs and obtains poor solutions to waste disposal problems.

An analysis of the heuristics indicates that ordinary continuous variables with no fixed charge must be treated the same as fixed charge variables. Typically, large problems will contain many continuous variables with no fixed charge as in the waste disposal problem. However, the adjacent extreme point algorithms handle these problems as if all variables were fixed charge variables. The standard test problems from Cooper and Drebes [13] and the fixed costs transportation problem are pure fixed charge problems.

The techniques for specialized problems exploit more efficient algorithms than linear programming for their solution. As a result, solution techniques have been developed specifically for specialized problems which are considerably more efficient. However, their application is generally restricted.

The methods which obtain the optimal solution to specialized fixed charge problems will handle problems with relative ease which would be considered very large if applied to a solution technique for solving general fixed charge problems. For example, the methods which generate an optimal solution for the fixed cost transportation problem perform better than the methods for obtaining an approximate solution to general fixed charge problems (Kennington and Unger [52]). Methods for obtaining approximate solutions to specialized fixed charge problems will handle much larger problems than the corresponding techniques for optimal solutions.

2.5.2. Research Objectives

The solution techniques for the fixed charge problem can be classified by the type of problem they solve (general or specific) and by the type of solution they are capable of producing (optimal or approximate). These can be evaluated by how well they do on small and large, general or specialized problems (Exhibit 2-18). While performance is good for all classes of small problems, only the specialized algorithms have been shown to work well for large problems. Of course, if a problem has any other features, it is no longer a specialized problem and a more general technique must be used.

Exhibit 2-18: Performance of Solution Techniques

Problem		Algorithm: General, Optimal	Algorithm: General, Approximate	Algorithm: Specialized
Small	General	good	good	n/a
	Specialized	good	good	good
Large	General	poor	?	n/a
	Specialized	poor	?	good

The major problem area is large general fixed charge problems. Results for optimizing techniques for the general fixed charge problem have not been encouraging due to the nature of branch and bound methods. Although some work has been done on developing heuristics for the general fixed charge problem, this work has been quite limited in scope and requires additional testing. The most promising area for research appears to be the testing of the current heuristics and the development of new heuristic methods for solving large general fixed charge problems. The research objectives can be summarized as follows:

1. Develop a new solution technique for solving large general fixed charge problems
2. Evaluate the new solution technique on a cross-section of fixed charge problems.
3. Compare the new solution technique with alternative methods for solving large fixed charge problems.
4. Evaluate the current approximate algorithms on a cross section of fixed charge problems.

CHAPTER 3

A NEW APPROXIMATE SOLUTION TECHNIQUE FOR LARGE GENERAL FIXED CHARGE PROBLEMS

3.1. Overview

As identified in the research objectives in the previous chapter, there exists a need for a method of solving large fixed charge problems which do not have the structure required for the efficient specialized techniques (Exhibit 2-18). The techniques which can provide the optimal solutions to fixed charge problems appear to work well only with small problems. Larger problems quickly require large amounts of computer time. The most promising area for exploration is the development of techniques for obtaining approximate solutions for these problems. The current approximate methods, primarily the adjacent extreme point heuristics, have had limited success. Their major success has been reported on some moderate size problems which have special structures.

Although the new algorithms to be developed for solving fixed charge problems will be approximate methods, some features will be borrowed from optimization approaches. Necessary conditions for a solution to be an optimal solution will be developed. Since sufficient conditions are difficult to develop, heuristic rules will provide quasi-sufficient conditions for an optimal solution. In addition, these heuristic rules will indicate how to improve the solution.

The new algorithms will exploit features found in the revised simplex method using the product form of the inverse which is the standard method for solving large linear programming problems. Continuous variables without a fixed charge will be ignored by new algorithms with their values determined by the linear programming algorithm. In contrast, the adjacent extreme point heuristics treat all variables as fixed charge variables and work best on the regular simplex method which maintains a full tableau

3.2. Conceptual Foundation

Hirsch and Dantzig [45] show that the optimum of a fixed charge problem exists at an extreme point of the associated linear programming problem (ALP) (Exhibit 2-12). All algorithms developed for the fixed charge problem use this feature. Vertex generation, in particular, implicitly enumerates all the vertices of (ALP).

The adjacent extreme point heuristics move from extreme point to extreme point until, by using heuristic rules, they terminate their search. Although all the feasible extreme points of (ALP) could be examined, these methods only look at a subset of all the extreme points in (ALP). This subset is developed as the algorithms progress using various heuristic rules.

The new solution technique examines a subset of the extreme points of (ALP), which is defined a priori. This subset is defined by the necessary conditions for optimality. Heuristics rules are subsequently introduced to develop quasi-sufficient conditions for optimality. These quasi-sufficient conditions if not met will indicate how to improve the solution to (MIFCP). In the following discussions, (MIFCP) will be used for simplicity. However, the discussion can also be applied to (NLFCP) as well.

3.2.1. Necessary Conditions for Optimality

3.2.1.1. Definition

As will be shown, it is only necessary to examine a subset of the feasible extreme points in (ALP). If the Y_j, which represent the actual fixed charge, are set to either zero or one for all j in (MIFCP), then the term in the objective function given by;

$$\sum_j f_j Y_j$$

would be constant. If a feasible solution exists, the optimal solution to (MIFCP) can be obtained by solving the fixed charge problem with the continuous costs only or (FCPC) (Exhibit 3-1). (FCPC) is, of course, a simple linear programming problem. Therefore, once all the Y_j have been determined, it will be relatively easy to determine the X_j.

Exhibit 3-1: Fixed Charge Problem Continuous Portion

(FCPC) minimize $z_c = \sum_j c_j X_j$

subject to:

$$\sum_j a_{ij} X_j = b_i \qquad \forall\, i$$

$$X_j \le u\, Y_j \qquad \forall\, j$$

$$X_j \ge 0 \qquad \forall\, j$$

where:

u = a large number

For a facility location problem, determining the value for Y_j is equivalent to deciding a priori which facilities will be open or closed and then minimizing the variable costs. For example, if it was decided, with out respect to the optimal solution, which warehouses to have open and closed, the capacitated warehouse location problem becomes a standard transportation problem.

A lagrangean relaxation of the constraints in (FCPC) involving Y_j for all j would shift these constraints into the objective function. If m_j is defined as the lagrangean multiplier, the Kuhn-Tucker conditions require that equation

(3.1) holds before a solution can be considered an optimal solution to (FCPC).

$$m_j \, (X_j - u \, Y_j) = 0 \quad \forall \, j \qquad (3.1)$$

Since by definition, X_j is strictly less than u, the two components of equation 3.1 can be treated independently requiring both equations (3.2) and (3.3) to hold.

$$m_j \, X_j = 0 \quad \forall \, j \qquad (3.2)$$

$$m_j \, Y_j = 0 \quad \forall \, j \qquad (3.3)$$

If equation (3.2) holds, it implies that equation (3.3) also holds. Therefore, the term involving Y_j can be removed from the objective function of the lagrangean relaxation of (FCPC). An equivalent formulation to (FCPC) for solving (MIFCP) would be given by solving the associated linear programming problem with modified variable costs or (ALPM) given in Exhibit 3-2. The lagrange multipliers, m_j, must be selected according to equation (3.4) which is consistent with equation (3.2).

$$m_j \begin{cases} = 0 & \text{if } X_j>0 \\ \geq q_j \geq 0 & \text{if } X_j=0 \end{cases} \quad \forall j \tag{3.4}$$

where:

q_j = critical quantity required to keep X_j out of the solution.

The critical quantity, q_j, itself, is not explicitly required as m_j is sufficient to solve for X_j. Since m_j is a range, it is easier to determine than an exact value for q_j. The m_j's can be selected such that solving (ALPM) will generate the optimal solution to (MIFCP).

Exhibit 3-2: The Associated Linear Programming Problem with Modified Variable Costs

---

(ALPM) minimize $z_c = \sum_j (c_j+m_j)X_j$

subject to:

$$\sum_j a_{ij}X_j = b_i \qquad \forall i$$

$$X_j \geq 0 \qquad \forall j$$

For given values of m_j (however selected), the X_j can be determined from the linear programming solution to (ALPM). The Y_j can be determined by a simple inspection: if $X_j>0$, then $Y_j=1$; if $X_j=0$, then $Y_j=0$.

Therefore, a solution to (ALPM) which also satisfies equation (3.4) will be a member of the subset identified above. Every member of the subset could potentially be the optimal solution of the original problem. However, the optimal solution must be a member of the subset. Thus, permissible values must be determined for the m_j such that a solution to (ALPM) constitutes a member of the subset. The following procedure will accomplish this task.

3.2.1.2. Test Procedure

Consider J to be the set of all fixed charge variables. Exhibit 3-3 defines three mutually exclusive subsets of J. All variables must belong to one of the subsets J^{in}, J^{out}, or J^{free} for the solution to meet the necessary conditions for optimality. Define J^{fail} as the set of variables $\{j|X_j>0,m_j>0\}$ not meeting the necessary conditions. Then for all solutions, $J=J^{in}\cup J^{out}\cup J^{free}\cup J^{fail}$.

If $X_j>0$, the fixed charge is not relevant to small changes in X_j and hence need not be considered, a small change being any change which does not alter any of the fixed charges. Since $X_j>0$, the only change which would affect the fixed charge would be to set $X_j=0$. Any other change has no impact on the fixed charge. Therefore, the incremental cost of these changes would be the variable cost only. If $X_j=0$, m_j represents the unit cost of absorbing the fixed charge resulting from any small change (increase) in

Exhibit 3-3: Necessary Conditions of X_j and m_j for Optimality

1. $J^{in} = \{j|X_j>0, m_j=0\}$
2. $J^{out} = \{j|X_j=0, m_j>0\}$
3. $J^{free} = \{j|X_j=0, m_j=0\}$

X_j. Any change in the value of X_j must bring the variable into the solution and incur the fixed charge.

However, if $X_j>0$ and $m_j>0$, the solution to (ALPM) is not optimal for (MIFCP). To restore necessary conditions for optimality will require that the values for m_j be changed. After modifying the m_j, (ALPM) must be resolved to obtain new values for X_j.

Restricting the discussion to the single variable j where $X_j>0$ and $m_j>0$, two possible changes to m_j can be made:

1. Increase m_j such that X_j is forced to zero, or
2. Set $m_j=0$.

Any other changes to m_j will result in a solution to (ALPM) that still violates the necessary conditions for optimality. For example, increasing m_j such that X_j is still greater than zero will not meet the necessary conditions and m_j must be increased further. Decreasing m_j but not to zero will also result in $X_j > 0$ and $m_j > 0$ which continues to violate the necessary conditions for optimality.

The initial procedure for solving (MIFCP) involves selecting m_j for all j, solve (ALPM) and then possibly modifying the m_j to meet the necessary conditions and resolving (ALPM). Except for special cases which will be discussed later, all the m_j for the variables in the set J^{fail} will be set to zero and is outlined in Exhibit 3-4. Since there are a finite number of fixed charge variables, the procedure must converge although it will generally require one or two iterations. This identifies one member of the subset of extreme points of (ALP) which can be considered as an optimal solution to (MIFCP).

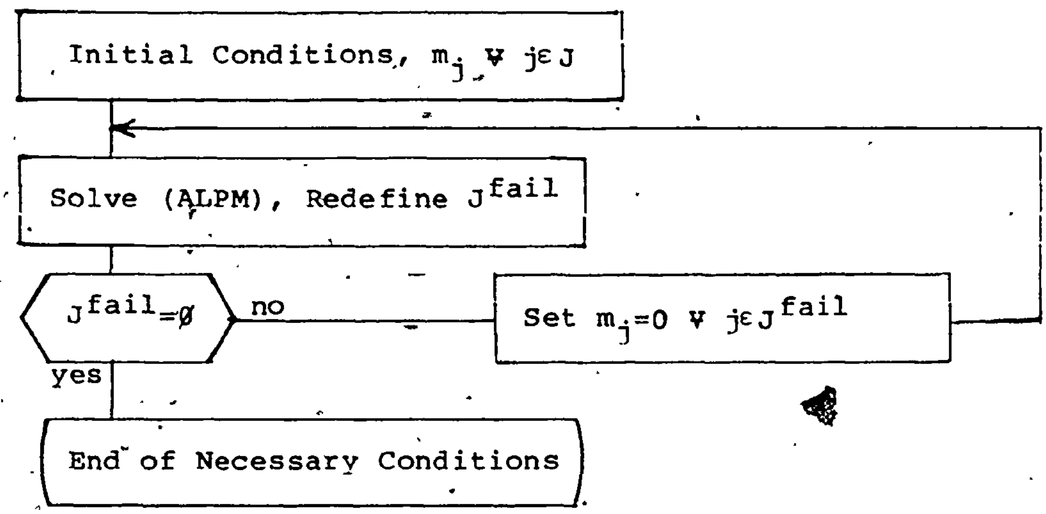
3.2.2. Quasi-Sufficient Conditions for Optimality

3.2.2.1. Definition

The conditions outlined in Exhibit 3-3 specify necessary conditions for optimality but they do not guarantee an optimal solution. Rules will be developed to determine if a solution is a "good" solution to (MIFCP). Since determining if, in fact, a solution is optimal is difficult, these rules will not guarantee an optimal solution and will be referred to as quasi-sufficient conditions for optimality.

Starting with a solution which meets the conditions in Exhibit 3-4, changes will be made to various m_j and (ALPM)

Exhibit 3-4: Testing for Necessary Conditions



will be re-solved in an effort to find solutions to (MIFCP) with improved objective function values. If a better solution is found, the process is repeated. If the changes to various m_j fail to find a solution with an improved objective function value, then the current solution will be deemed to meet the quasi-sufficient conditions for optimality.

Clearly, the number of possibilities for changing various m_j is enormous. The number of possible simultaneous changes is the factorial of the number of fixed charge variables. In addition, the actual value for each m_j must be specified.

If the discussion is restricted to changing one m_j, the number of possible changes is equal to the number of fixed charge variables. Using the restriction of changing only one m_j, there are only two types of changes which can occur (Exhibit 3-5). These changes are referred to as an Allocation for increasing m_j by allocating the fixed charge, and a Deallocation for setting $m_j=0$ and removing a previous allocation. For clarity, the variable which will have an allocation made will be referred to by the index i and the variable which will have a deallocation will be referred to by the index k.

All other changes either produce solutions which violate the necessary conditions (Exhibit 3-3) or actually result in no change to the solution. For example, increasing m_j if $X_j=0$ will obviously not change the solution. Decreasing m_j, but not to zero, when $X_j=0$ will either leave $X_j=0$ (i.e. no change) or allow X_j to increase which violates the necessary conditions. If $X_j>0$ and m_j is increased, X_j must be changed to zero otherwise a necessary condition is violated.

Exhibit 3-5: Possible Single Changes to m_j

Process	Status of X_j & m_j	Change	Possible Consequence
a) Allocation	$X_i>0$ ($m_i=0$)	increase m_i	force X_i out
b) Deallocation	$m_k>0$ ($X_k=0$)	set $m_k=0$	allow X_k in

a) The first single change in Exhibit 3-5 attempts to force variable i out of the solution. If, after re-solving (ALPM), $X_i>0$ then the necessary conditions will be violated. In order to meet the necessary conditions, m_i will have to either be increased further or set back to zero depending upon the procedure being used which will be detailed below.

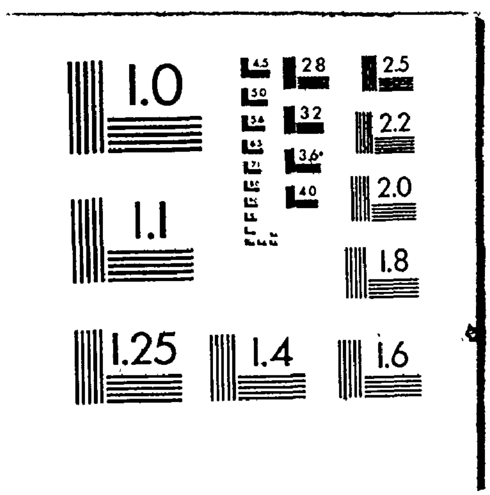
b) The second single change in Exhibit 3-5 allows variable k to come into the solution. The necessary conditions for the variable k would be met by definition after re-solving (ALPM) since m_k would be zero.

Allowing two variables to change their values of m_j would result in (number of fixed charges)*(number of fixed charges-1) possible combinations. In addition, there would be many possibilities for the value of the m_j for each change. Due to the large numbers, examination of multiple changes will be restricted. With the single change, there are two types of changes. The first change, an Allocation which increases m_i, attempts to force X_i out of the solution. The second change, a Deallocation setting m_k to zero, allows X_k to come into the solution. The combination change will combine the two single changes by combining each Allocation with all possible Deallocations.

A solution to (ALPM) will meet the quasi-sufficiency condition when all the single (combination) changes to m_j along with subsequent changes to meet the necessary conditions for optimality fail to produce a solution with an improved objective function value for (MIFCP). A solution which produces a better objective function value for (MIFCP) also indicates how to improve the values of different m_j to obtain the new improved solution.

Although the quasi-sufficiency conditions for optimality will show how to improve the solution, it is not obvious what to do as soon as a new improved solution is found when a quasi-sufficiency condition is not met. One approach would to be to use this new improved solution as the basis for further testing. Unlike linear programming where any changes which consistently improve the objective function will eventually lead to the optimum, the order in which changes are made is important for the fixed charge problem.

2



Primarily due to its intuitive appeal, it was decided to select the change which made the best improvement. Of course, determining the change which makes the best improvement implies all the possible single (combination) changes must be made in order to evaluate them. Clearly, this is not the only rule that could be used. This would be an area for further research.

The next task is to define the specific test for the quasi-sufficient conditions.

3.2.2.2. Test Procedure

The previous section outlined the two types of quasi-sufficiency conditions: a single change and a combination change. Each type of condition will require a test procedure. For each test, the best improvement in the objective function of (MIFCP) is used to generate the next solution. If no improvement is found, the quasi-sufficient conditions for optimality are met and the test terminates.

3.2.2.2.1. Quasi-Sufficiency Test for a Single Change

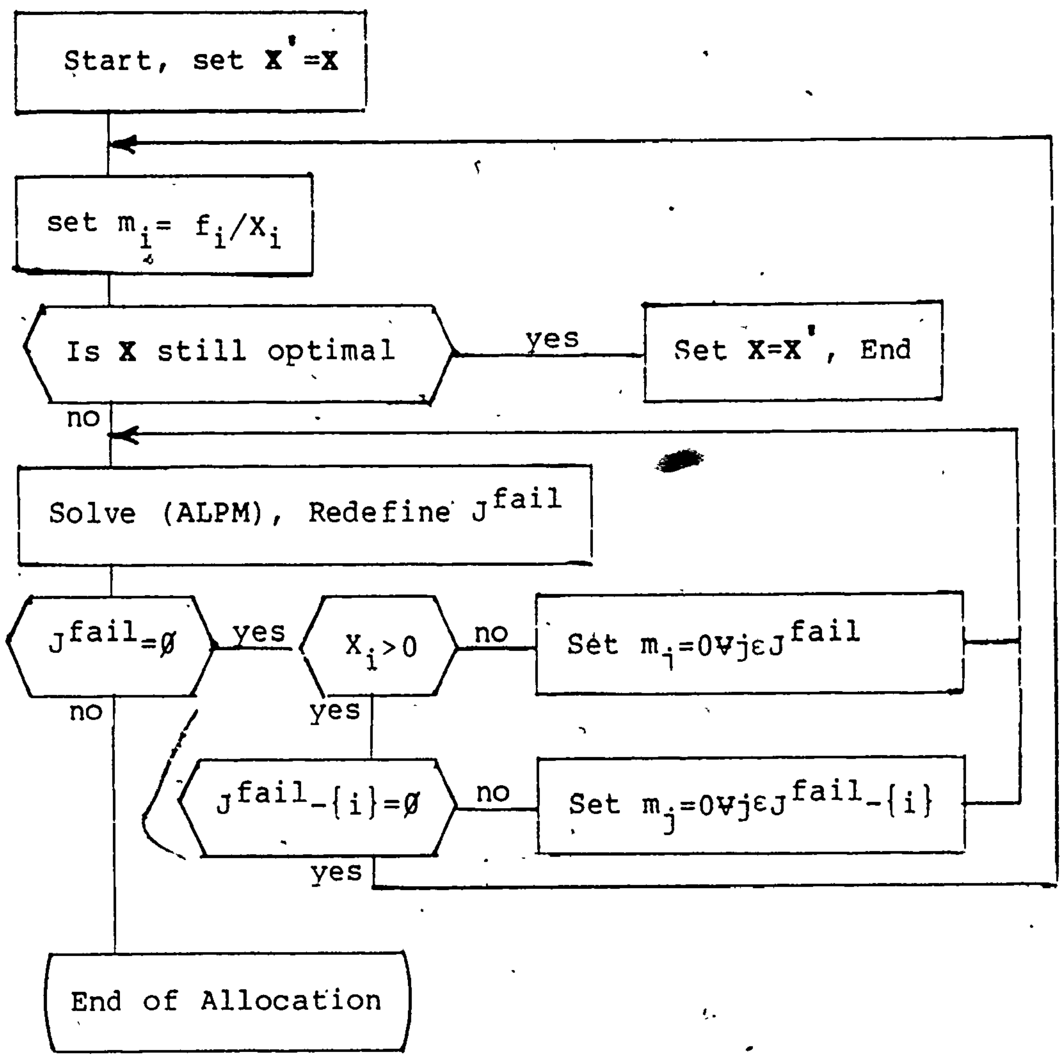
The quasi-sufficiency test for a single change involves selecting one of the changes given in Exhibit 3-5 and solving (ALPM). The test for necessary conditions is made (Exhibit 3-3) which may require a modification to the value of some m_j and resolving (ALPM). This process produces a new solution whose objective function value for (MIFCP) is determined and compared with the objective function value of the current best solution.

A different procedure is required for each of the two types of changes, an Allocation and a Deallocation. The allocation of a fixed charge is described as follows. If a variable i with $X_i>0$ and $m_i=0$ is to be tested, the only change that can be made is to increase m_i. After the change, (ALPM) is solved, the test for necessary conditions performed with subsequent changes to various m_j, if required, and the resolving of (ALPM) as outlined in Exhibit 3-6. The vector containing a solution is referred to as X with no subscript.

The initial increase in m_i is calculated by allocating the fixed charge with f_i/X_i. This increase represents the prorated change in the objective function if X_i is forced to zero. After making the allocation and solving (ALPM), a test for the necessary conditions must be made. First, all variables other than variable i are examined to insure that there is no variable j where both $X_j>0$ and $m_j>0$. If any are, the appropriate m_j is set to zero and (ALPM) is resolved.

If X_i is still greater than zero after the allocation, there are two possibilities which can occur. These are determined by comparing X_i to X_i', the previous value of X_i which was saved. If $X_i<X_i'$, the allocation has succeeded in decreasing X_i but not to zero. The allocation process is

Exhibit 3-6: Quasi-Sufficiency Test for Single Change
- Allocation of Fixed Charge to Variable i



repeated with a new and higher allocation.

If $X_i=X_i'$, the allocation produced no change to X_i. It is deemed that there will be no improvement to the objective function of (MIFCP) by removing variable i from the solution. Although it would be possible to increase the value of m_i further to attempt to remove the variable i from the solution, this is not done. Thus, the effort required to remove i from the solution is not incurred.

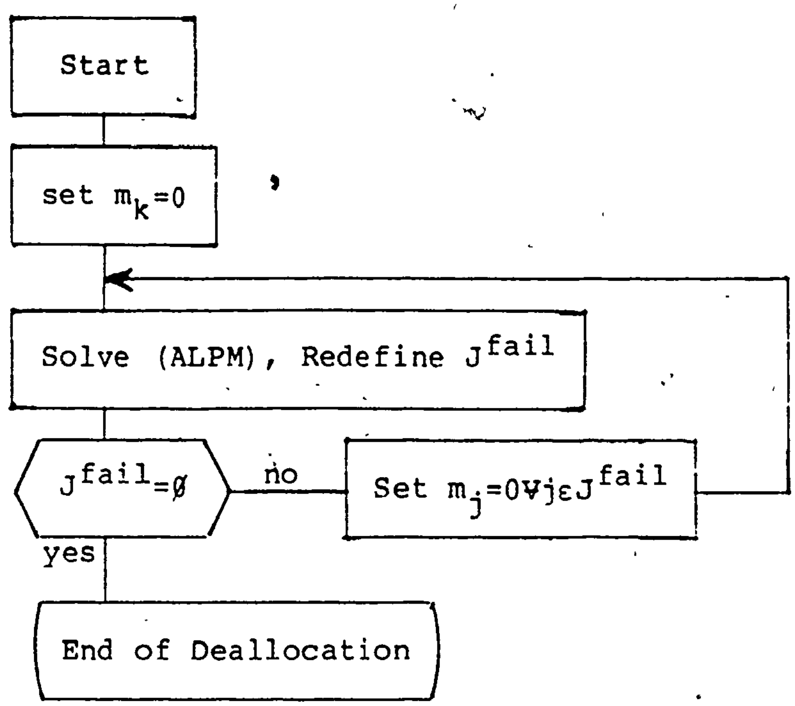
The <u>deallocation of the fixed charge</u> is described as follows. If a variable k, with $m_k>0$ and $X_k=0$, is to be tested, the only acceptable change is $m_k=0$. Obviously, increasing m_k will not change the solution. A decrease in m_k to a value larger than zero may, after solving (ALPM), result in $X_k>0$. However, this would violate a necessary condition for optimality.

The deallocation procedure involves removing a previous allocation and solving (ALPM). A test is made for necessary conditions. Possible changes may be made to various m_j and (ALPM) resolved. The new solution to (MIFCP) is evaluated by determining objective function value of the new solution and comparing it to current solutions. This procedure is outlined in Exhibit 3-7.

3.2.2.2.2. Quasi-Sufficiency Test for a Combination Change

The combination change involves one allocation and one deallocation. This test attempts to bring in each variable which is being kept out of the solution with an allocation while at the same time remove any fixed charge variable currently in the solution.

Exhibit 3-7: Quasi-Sufficiency Test for Single Change - Deallocation of Fixed Charge to Variable k



If variable i, with $X_i>0$ and $m_i=0$, and variable k, with $X_k=0$ and $m_k>0$, are to be tested jointly, then variable k can be allowed into the solution by a deallocation (set $m_k=0$) and variable i can be forced out of the solution by an allocation (set $m_i>0$). This procedure is outlined in Exhibit 3-8. The combination test attempts to replace variable i with variable k, although with the necessary conditions to be met, other variables may also change their status. The allocation and deallocation procedures are identical to those found in the test for a single change

(Exhibits 3-6 and 3-7). The deallocation procedure is implemented first in order to avoid the problem of variable i coming back into the solution following the deallocation to variable k.

3.2.3. Synthesis of Necessary and Quasi-Sufficient Conditions

The basic procedure for solving the fixed charge problem will be to select some initial value for m_j for all j, solve (ALPM), test for necessary conditions with possible changes to various m_j and resolving (ALPM) and then test for quasi-sufficient conditions which will also show how to improve the solution, if required.

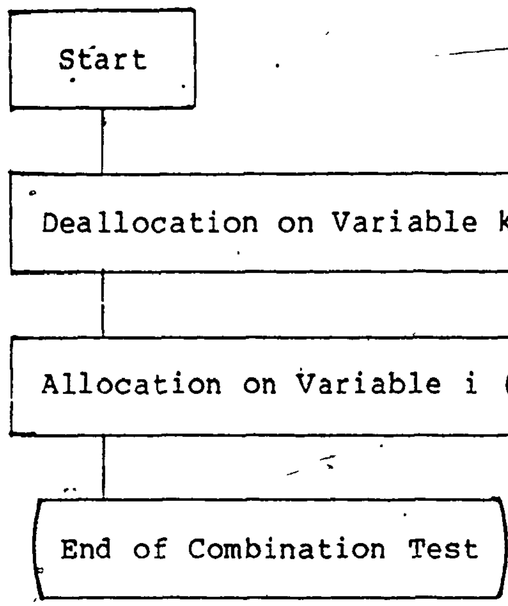
Some choices will have to be made with respect to which initial conditions to use. Also, different initial conditions will in most cases lead to different solutions. This will require some means of evaluating the different solutions in order to exploit them.

However, these considerations are less of a theoretical nature and relate more to operational implementation. Therefore, the discussion of initial conditions and different solutions is deferred to the next section.

3.3. Operational Implementation

The two components outlined above, the initial values of m_j coupled with the test for necessary conditions, plus

Exhibit 3-8: Quasi-Sufficiency Test for a
Combination Change - Deallocation on k
- Allocation on i



the repeated application of a quasi-sufficiency test, will be combined to form a phase. Each phase may use different quasi-sufficiency tests. The phase with a single change involving both an allocation and a deallocation is shown in Exhibit 3-9. The phase involving the combination change is shown in Exhibit 3-10. Combinations of different phases will make up a part of the new algorithm. Finally, since the different combinations of phases will generate different solutions, a comparison and evaluation of the different solutions will have to be made.

Exhibit 3-9: A Phase - Quasi-Sufficiency Test for a Single Change

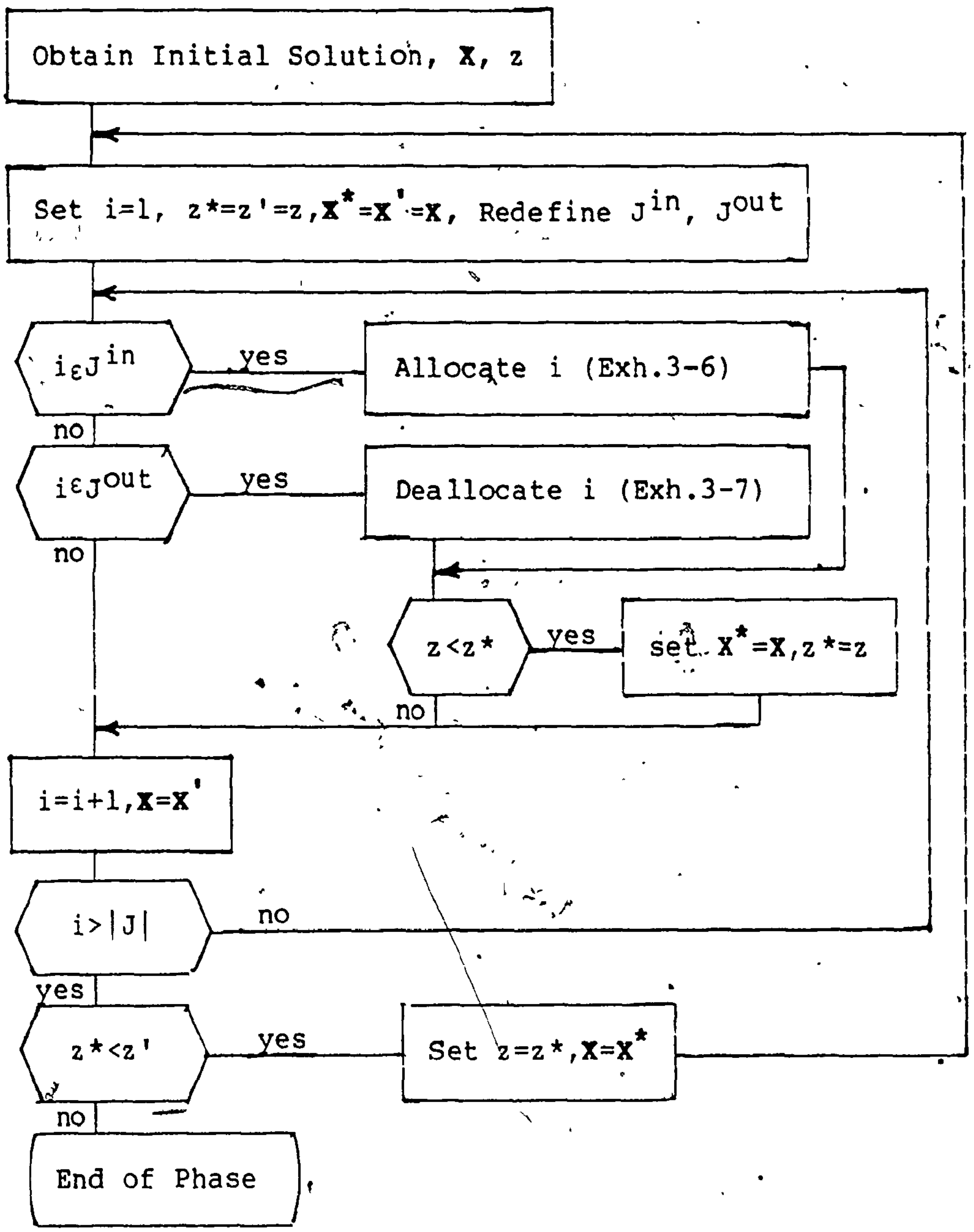
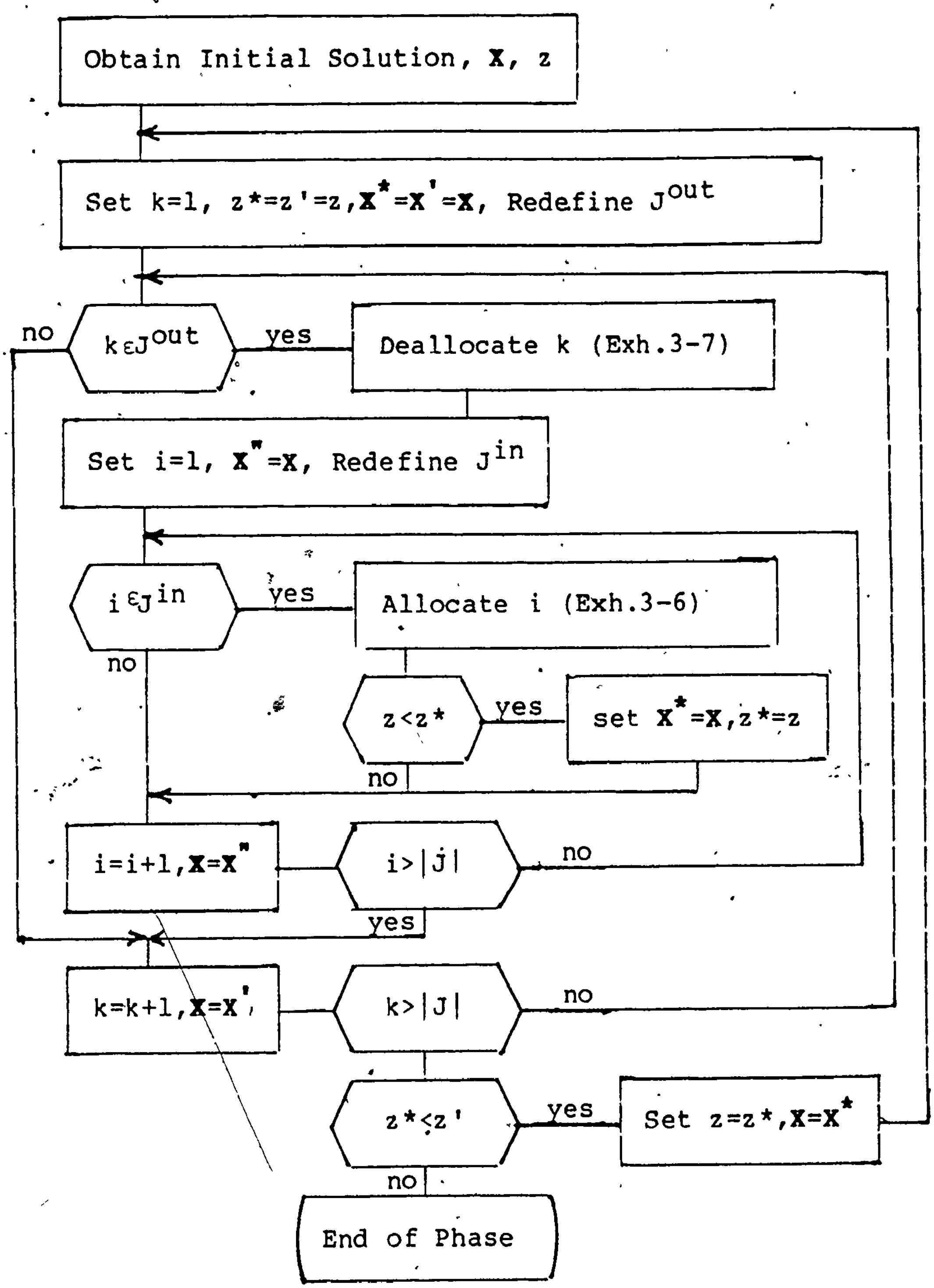


Exhibit 3-10: A Phase - Quasi-Sufficiency Test for a Combination Change



Each phase begins with an assumption with respect to m_j for all j developed either exogenously or from a previous phase. Then, through a process of allocation and deallocation within the quasi-sufficiency tests, improved solutions will be found. When the quasi-sufficiency conditions are met, no new improved solutions can be found. At this point, the algorithm terminates or new assumptions can be made with respect to m_j for all j and the process repeated.

The different initial conditions, which will be introduced in the following section, represent solutions dominated by either the continuous costs or the fixed charges. Other initial conditions will include corrections to these two.

3.3.1. Initial Conditions

Two methods are used to obtain initial conditions. The first method exogenously selects values for all m_j which represent suitable values. The second method takes a solution from a preceding phase and adjusts the value of some of the m_j and continues.

3.3.1.1. Initial Conditions Dominated by the Continuous Costs

The initial default value for all m_j is set at zero. Of course, this simply solves the associated linear programming problem (ALP) using the continuous objective function. By definition, this solution meets the necessary conditions for optimality in Exhibit 3-3. These initial conditions provide a promising area for search for problems in which the continuos costs are more significant.

3.3.1.2. Initial Conditions
Dominated by the Fixed Charges

The initial conditions in the previous section are dominated by the continuous objective function. In this section, initial conditions dominated by the fixed charges will be developed by quickly incorporating the fixed charges into various m_j after initially solving (ALP). The fixed charges are allocated to all positive variables using equation (3.5) and (ALPM) is solved.

$$m_j' = \begin{cases} \max\{ m_j, f_j/X_j \} & \text{if } X_j>0 \\ m_j & \text{if } X_j=0 \end{cases} \quad \forall j \tag{3.5}$$

The process is repeated with Equation (3.5) applied to each solution and (ALPM) solved until the solution stabilizes. Although the process will usually require two or three iterations, an arbitrary limit is imposed to prevent excessive looping which will not accomplish significant improvements. At this point, the solution will violate the necessary conditions for optimality as a number of variables will have $X_j>0$ and $m_j>0$. In order to meet the necessary conditions, the value for m_j for these variables will be set to 0 and (ALPM) resolved. This procedure is outlined in Exhibit 3-11.

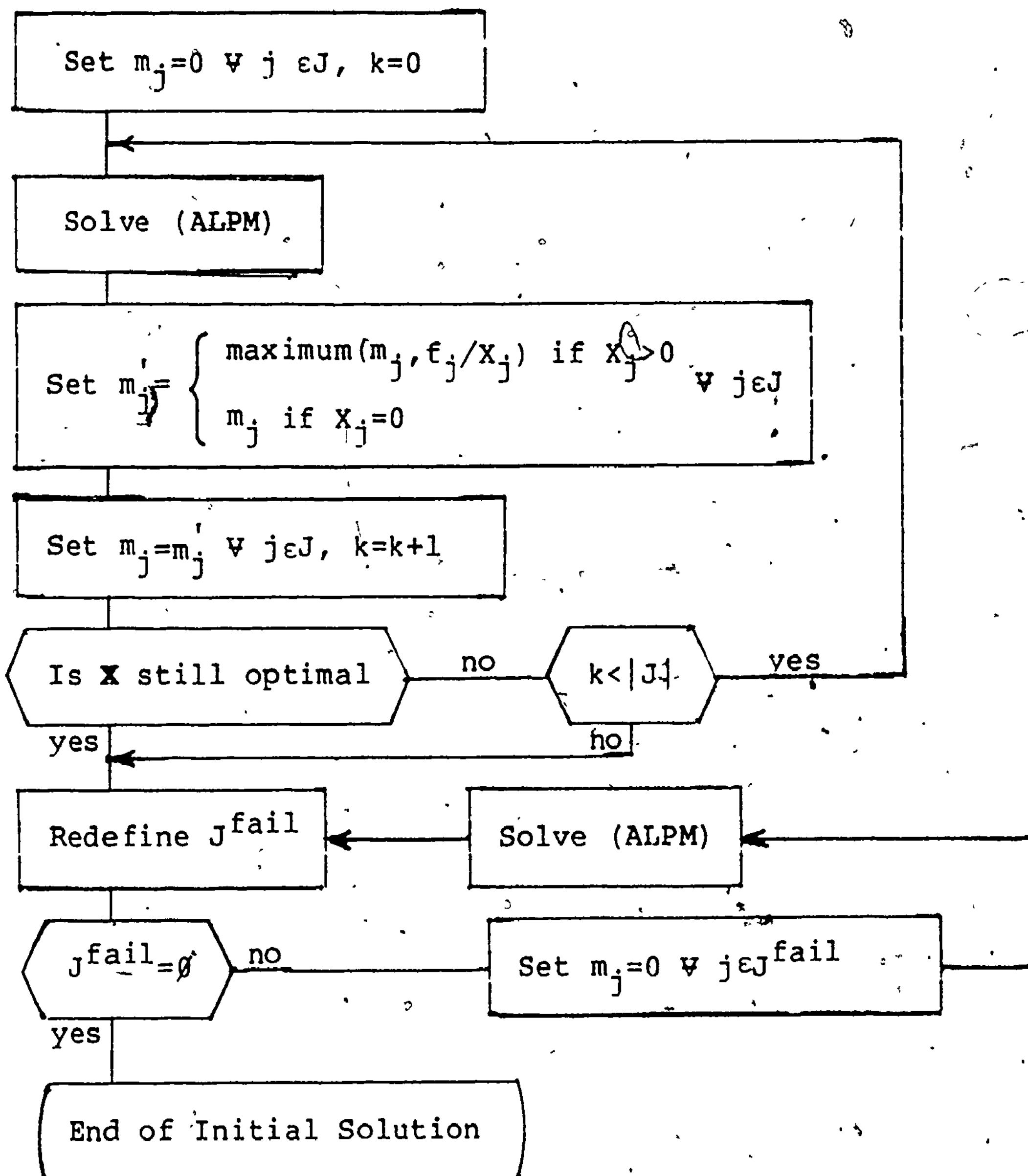
3.3.1.3. An Adjustment to the Initial Conditions

At the end of any phase starting with one of the initial conditions previously given, the fixed charge variables can take on three possible states depending on the values for X_j and m_j (Exhibit 3-3).

The first condition, set J^{in}, is for variables in the solution. If $X_j>0$, m_j must be zero. The second condition, set J^{out}, results from variables which were in the solution at one point but were driven out. Since $m_j>0$, an allocation of the fixed charge must have been made which forced the variable out of the solution. The third condition, set J^{free}, with both $X_j=0$ and $m_j=0$, represents variables which have never been in the solution.

Variables in the third condition quite often have small continuous costs with large fixed charges. When these variables come into the solution, they are rejected because of the large fixed charge. However, because of their small continuous cost, they may be keeping other variables from coming into the solution in their place. The initial allocation of the fixed charge ($m_j=0$) is a poor estimate and consequently will be increased to infinity. Therefore, the initial condition for this phase will modify the m_j from a

Exhibit 3-11: Initial Solution Dominated by Fixed Charges



previous phase by:

$$\text{Set } m_j = \text{infinity } \forall\ j \in \{j \mid X_j = 0, m_j = 0\} \qquad (3.6)$$

The necessary conditions for the current solution are by definition met and testing for the quasi-sufficient conditions can begin.

3.3.2. Multiple Changes from Different Solutions

The different initial conditions outlined lead to basically two different solutions to (MIFCP). The initial conditions dominated by continuous costs produces one solution while the initial conditions dominated by the fixed charges produces another. The third tvpe of initial condition is only a modification to the initial assumptions made by the previous two methods. The fact that there are two different solutions to (MIFCP) requires some resolution of which solution to use and what to do with the other. Picking the best solution and discarding the other would be one option. Alternatively, information from both solutions could be used in order to improve the best solution. Consequentlv a procedure is developed to compare these two solutions and attempt to produce multiple changes which will improve the best solution found so far.

As a result of the different initial solutions, there will be two solutions to (MIFCP): the best solution found so far, designated by the vector X^2, and an alternative solution designated by the vector X^1. In order to improve on the solution given by the X^2, multiple changes to the solution must be made since the quasi-sufficiency conditions have already been met. Therefore, the problem becomes how to heuristically make selective multiple changes to the best solution found so far that will lead to further improvement. This heuristic will make use of existing information encompassed in the two solutions represented by X^1 and X^2.

The heuristic begins by developing a set of variables which have a different status in the two solutions (Exhibit 3-12). The status in a solution relates to a variable being strictly greater than zero (and incurring the fixed charge). If the set of variables with different status is empty, the solutions are the same. If not, the set represents a multiple change to the best solution found so far.

Exhibit 3-12: Set of Variables with Different Status in Two Solutions

$$\text{set } S = \{\, j \mid X_j^1>0,\ X_j^2=0 \,\} \cup \{\, j \mid X_j^1=0,\ X_j^2>0 \,\}$$

The multiple change above, represented by the X^1, leads to a "good" solution as opposed to a multiple change picked at random which may lead to a bad solution. Since this multiple change does not lead to an improved solution, a new multiple change will be generated leading to a a third solution which will be represented by the vector X'. The values for X' are created by forcing a restricted single change on the solution represented by X^1. The single change will be accomplished by either an allocation (Exhibit 3-6) or a deallocation (Exhibit 3-7) on the set of variables defined in Exhibit 3-12 which are different in the two solutions represented by X^1, a "good" solution, and X^2, the best solution found so far. The single change will generate the solution which has the best objective function value given that a single change must be made.

This new solution represented by X' may have an objective value that is less than the objective function value of the solution represented by the X^1. However, it represents a solution which has more similarity to the solution represented by X^2. The solution represented by X' replaces the solution represented by X^1. The process is repeated with the two solutions, determining the best and making a single restricted change to the other. Although the process initially produces inferior solutions, occasionally better solutions are found in subsequent iterations. Eventually, the two solutions become identical and the heuristic terminates.

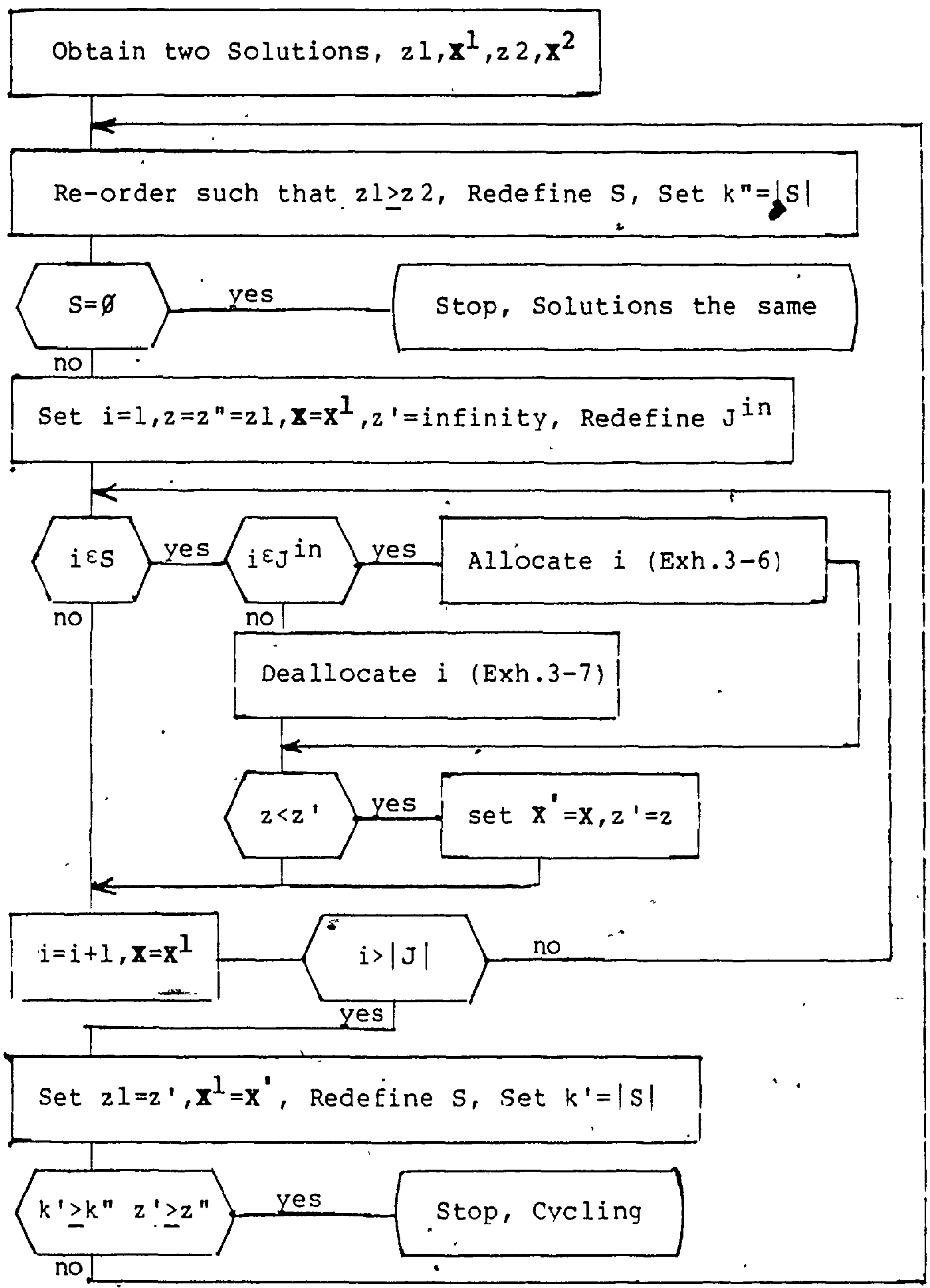
The process of comparing two solutions will occasionally start cycling as the heuristic attempts to move back to the solution represented by the original X_j. To prevent cycling, two numbers for each solution are determined. The objective function values of the solutions represented by x^1 and x', represented by z1 and z' respectively, are determined. Each of the two solutions is compared to the best solution found so far (represented by x^2) to determine the set of variables with a different status (Exhibit 3-12). A count of the number of variables with a different status is made for each solution and designated by k1 and k'. If $z1 \le z'$ (an inferior solution) and $k1 \le k'$, the heuristic may be cycling and is terminated with the solution represented by x^2 chosen as the best solution to the problem.

The heuristic is summarized in Exhibit 3-13. Although it does not always produce better solutions, it is relatively fast and occasionally produces good results.

3.4. The Cost Allocation Algorithms

The conceptual foundation and various components of the new cost allocation algorithms have been presented. It is now required to present the overall description of the algorithms. The basic cost allocation algorithm, COAL-b, will use the quasi-sufficiency test for a single change, two initial solutions and a comparison of the two results with a multiple change. This basic algorithm will be extended with COAL-x to include the quasi-sufficiency test for a combination change. In addition, two algorithms, COAL-c and

Exhibit 3-13: Multiple Changes from Two Solutions



COAL-f, are derived from the basic algorithm COAL-b and included for problems which exhibit special conditions.

3.4.1. The Basic Cost Allocation Algorithm - COAL-b

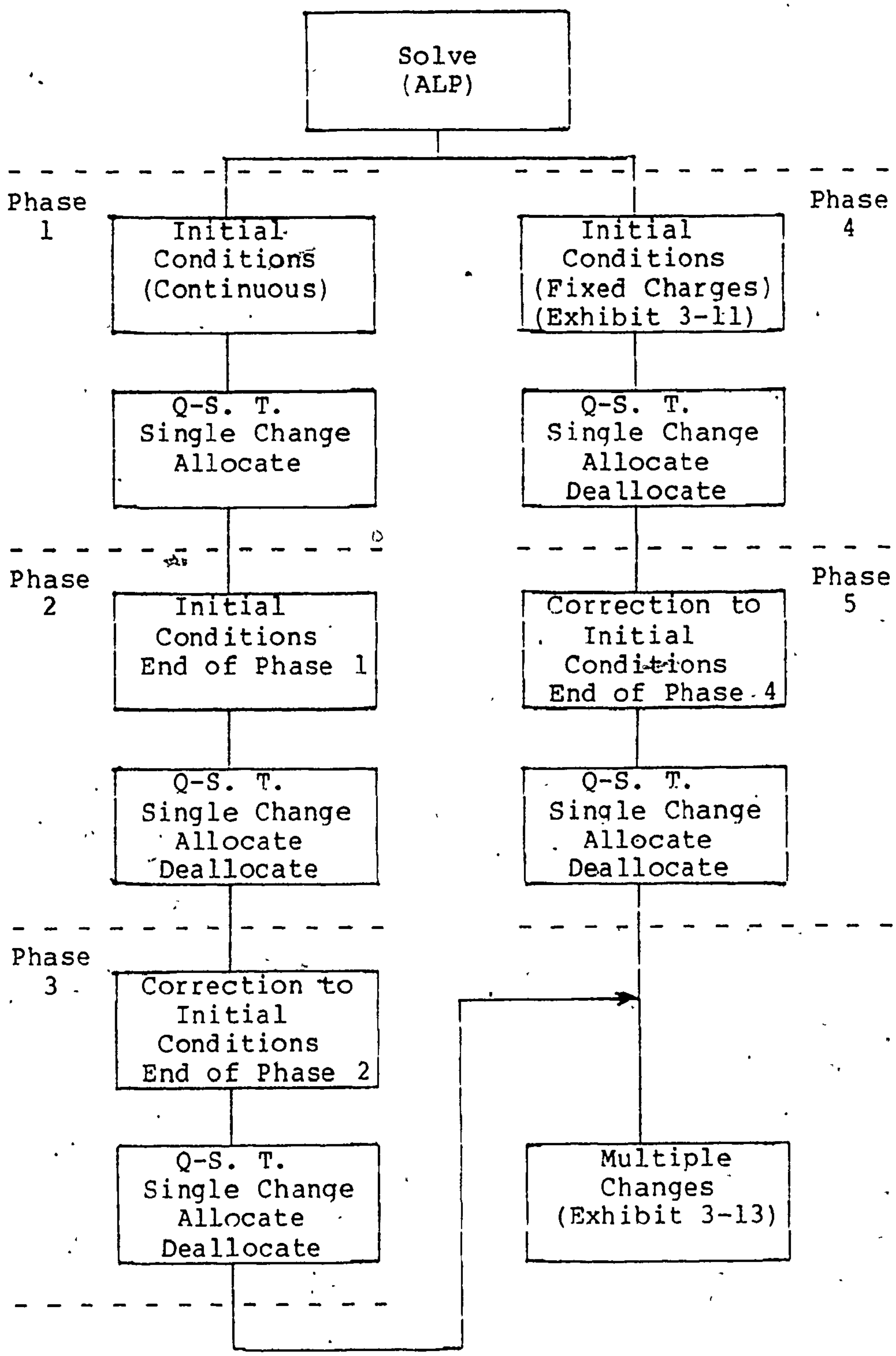
The Basic COst ALlocation algorithm, COAL-b, consists of two different initial starting solutions representing the continuous cost dominated solution and the fixed charge dominated solution, five different phases to improve the initial starting solutions and a comparison of the two final solutions. This is outlined in Exhibit 3-14.

3.4.1.1. An Initial Solution Dominated by the Continuous Costs - Phase 1

Phase 1 begins by selecting initial conditions with $m_j=0$ for all j. This essentially solves (ALP) and is an initial solution dominated by the continuous costs. Since there are no fixed charges allocated, the first quasi-sufficiency test with a single change looks at the allocation test only. The allocation process continues until further improvements can not be made.

At the end of Phase 1, the current best solution will have a number of fixed charge variables which are not in the solution and have $m_j>0$. The fact that $m_j>0$ indicates that at one point, this variable was in the solution and has been removed. The allocation in m_j is keeping the variable out of the solution. However, this is not an absolute

Exhibit 3-14: COAL-b - Basic Cost Allocation Algorithm



restriction. The requirement to meet the necessary conditions would allow a variable to come into the solution provided the additional allocation can be overcome.

3.4.1.2. Continued Search - Phase 2

The initial conditions for Phase 2 follow from the end of Phase 1 with no change. Since there are now several variables with $m_j>0$, it uses both the allocation and deallocation procedure as part of the quasi-sufficiency test for a single change. Again this continues until further improvements can not be made.

3.4.1.3. Correction to Initial Conditions - Phase 3

In Phase 1, the default allocation for all fixed charges was set at zero. As a result, at the end of Phase 3, many fixed charge variables which remain out of the best solution will still have an $m_j=0$. Variables with a relatively low continuous objective function coefficient and relatively high fixed charge will come into the solution as a result of another variable being forced out through an allocation. However, the solution appears poor because of the large fixed charge. These variables may prevent another variable from coming in and providing an improved solution.

The initial allocation for this phase identifies those fixed charge variables which have never been in the solution and sets the default value for m_j to infinity. The allocation and deallocation procedure is then repeated until further improvements can not be made.

After the quasi-sufficiency condition for a single change is met, the solution is saved in order to be used later. This solution represents a "good" solution when starting with the assumption that the continuous costs are dominant.

3.4.1.4. An Initial Solution Dominated by the Fixed Charges - Phase 4

Phase 4 restarts the algorithm with an initial solution that is dominated by the fixed charges. The three steps of the initial solution (Exhibit 3-11) incorporate the fixed charges into the m_j for a number of variables very quickly. In contrast, phases 1 through 3 build up the allocations of the fixed charges slowly, one variable at a time.

The initial solution of Phase 4 has a number of fixed charge variables with $m_j>0$. Consequently, Phase 4 continues with an allocation and deallocation procedure for the quasi-sufficiency test until further improvements can not be found.

3.4.1.5. Correction of Initial Conditions - Phase 5

This phase is a repeat of Phase 3. The initial conditions for this phase identifies those fixed charge variables which have never been in the solution and sets the default value for m_j to infinity. The allocation and deallocation procedure of the quasi-sufficiency test for a single change is repeated until further improvements can not be made.

This solution is then saved in order to be used later. This solution represents a "good" solution when starting with the assumption that the fixed charges are dominant. Typically, the solution at the end of Phase 5 is different from the solution at the end of Phase 3.

3.4.1.6. Compare Solutions

Finally, the solution generated by the Phases 1 to 3 is compared with the solution from Phases 4 and 5. Using the procedure outlined in Exhibit 3-13, a search is made for a multiple change which will improve the best solution found. This phase while not always obtaining improved solutions is always relatively fast.

3.4.1.7. Summary of COAL-b

This completes the description of the basic cost allocation algorithm. It consists of two approaches to solving a fixed charge problem with a method for synthesizing the two final solutions to look for a better solution. The first approach assumes that the continuous costs dominate and the second assumes the fixed charges dominate.

3.4.2. The Extended Cost Allocation Algorithm - COAL-x

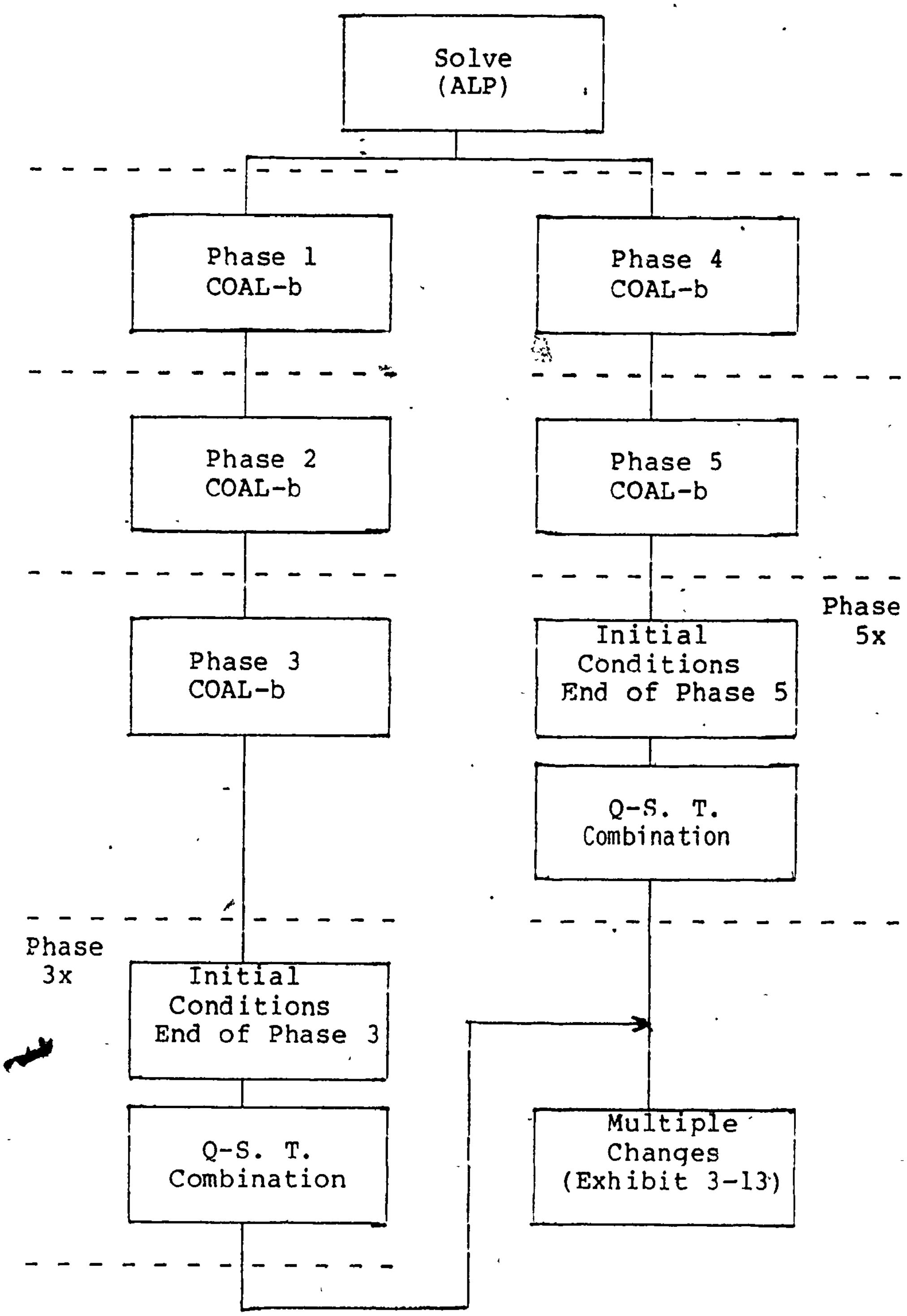
The previous algorithm changes one m_j at a time in a localized search for improved solutions. With the quasi-sufficiency test for a combination change, this is extended with two simultaneous changes in the m_j (Exhibit 3-8). An allocation on a variable in the solution is combined with the deallocation on a variable which currently has an allocation. This heuristic follows a similar format to COAL-b with a Phase 3x and a Phase 5x added (Exhibit 3-15).

The quasi-sufficiency test for a combination change will result in a large number of pairs of variables being tested. As a result, COAL-x will take considerably longer than COAL-b. Since COAL-x does take so long, testing of further combinations of variables is not carried out. Although the extended cost allocation algorithm, COAL-x, does take significantly longer than COAL-b, it consistently generates good solutions which are always as good as and usually better than solutions from COAL-b.

3.4.3. Cost Allocation Algorithms for Special Cases

Both COAL-b and COAL-x generate two solutions independent of each other. With problems of a particular type, one of these solutions may be consistently better. The second solution may add little benefit to the overall

Exhibit 3-15: COAL-x - Extended Cost Allocation Algorithm



solution. In these cases, it may be appropriate to use only part of the complete heuristic.

3.4.3.1. A Cost Allocation Algorithm Dominated by the Continuous Costs - COAL-c

Since the first three phases of COAL-b start with an initial solution dominated by the continuous charges, they tend to generate better solutions for those problems which are dominated by the continuous costs. In these cases, a algorithm consisting of these three phases would be appropriate providing good solutions faster than COAL-b. (Exhibit 3-16).

3.4.3.2. A Cost Allocation Algorithm Dominated by the Fixed Charges - COAL-f

In a similar fashion to COAL-c and the continuous costs, Phases 4 and 5 of COAL-b start with a solution which is dominated by the fixed charges. Therefore an algorithm consisting of these two phase would be appropriate for problems which are dominated by the fixed charges (Exhibit 3-17).

3.5. Computational Aspects of the New Heuristics

In the preceding discussion, the algorithm for solving (ALPM) was not specified. Since the cost allocation algorithms work through modifying the objective function,

Exhibit 3-16: COAL-c - Cost Allocation Algorithm Dominated by Continuous Costs

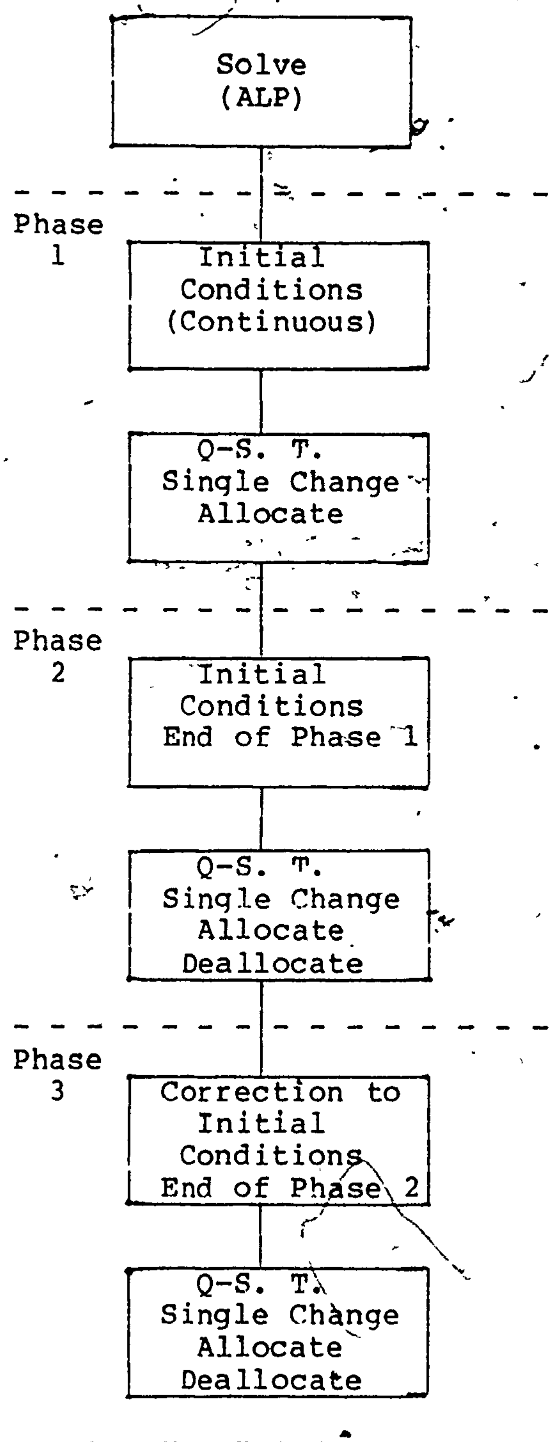
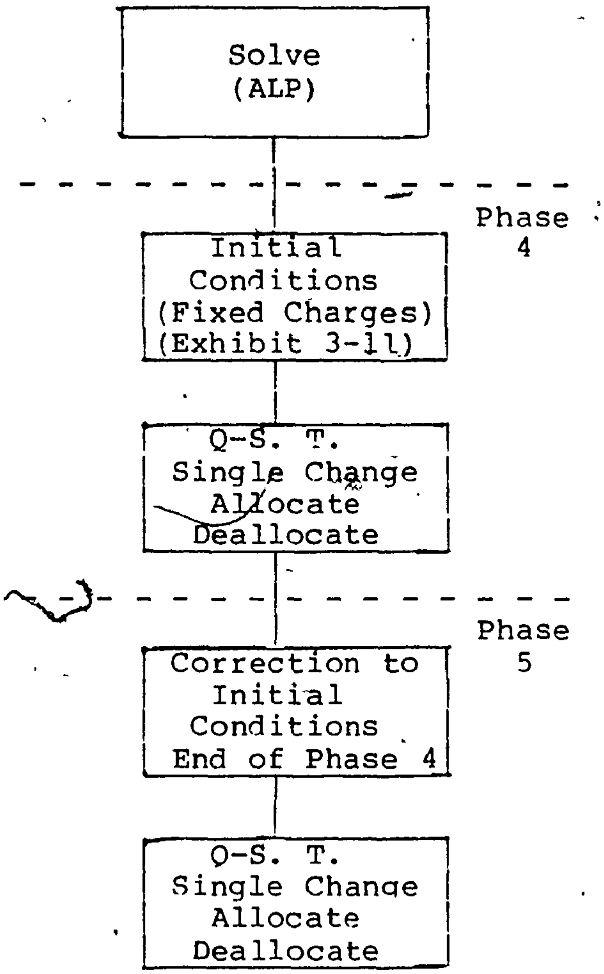


Exhibit 3-17: COAL-f - Cost Allocation Algorithm Dominated by Fixed Charges



any algorithm which solves (ALPM) will work. However, the new cost allocation algorithms are intended for large general fixed charge problems which require the flexibility of a linear programming formulation. Therefore, the appropriate tool for solving (ALPM) will be an algorithm suitable for large linear programming problems.

3.5.1. Solution of Large Linear Programming Problems

Large linear programming problems are typically very sparse with the number of non-zero coefficients usually less than 5%. The obvious choice for these problems is to use the revised simplex algorithm with the product form of the inverse.

The revised simplex algorithm format allows the storage of the original constraints in a very compact form. The product form of the inverse constructs the inverse from a series of eta-vectors; one for each simplex pivot. These eta-vectors maintain the sparse nature of the original problem far better than the explicit inverse itself.

Since an eta-vector is required for each simplex pivot, the space used by the product form of the inverse would soon become enormous. However, after a certain number of iterations, the inverse is re-generated with a new set of eta-vectors representing only those variables currently in the basis. An efficient re-inversion routine (Lasdon [58]) will dramatically reduce storage requirements, maintain computational accuracy and reduce the number of computations required.

3.5.2. Interface with the Cost Allocation Heuristic

The cost allocation algorithms will exploit some of the features of the product form of the inverse in order to efficiently solve a problem. In order for the cost allocation algorithms to determine the best operation at each step, a change must be made to (ALPM), the problem solved and evaluated. Then the original solution must be restored. The inverse of the original solution is stored with a series of eta-vectors. The solution from modifying the objective function can be obtained by adding a few eta-vectors. The inverse from the original solution is restored by resetting pointers with out requiring any calculations.

Periodically, some solutions are saved. However, only the solution itself is required and not the inverse or the simplex tableau. The re-inversion routine can be used to obtain the inverse if needed. Therefore, the memory requirements are increased by 4*m+2*f over linear programming (m=number of equations and f=number of fixed charges).

In the development of the new algorithms, no mention is made of ordinary continuous variables (i.e. variables with zero fixed charge). These variables are essentially ignored by the cost allocation algorithms and their values are determined by the more efficient linear programming algorithm. As such, problems with large numbers of ordinary continuous variables become similar to linear programming problems with large numbers of such variables. In the case of linear programming, although it certainly makes a problem more difficult, it is not considered a major difficulty. It would be quite feasible to include a column generation technique or a decomposition method in the cost allocation algorithm as part of the procedure for solving (ALPM).

The integration of the revised simplex method using the product form of the inverse results in very efficient algorithms for solving the fixed charge problem. The algorithms are able to solve problems quickly with little increase in memory requirements over ordinary linear programming.

3.6. Summary

The algorithm used by Cooper and Drebes [13] makes an allocation of the fixed charges to all j where $X_j>0$. The Balinski [4] method allocates the fixed charges to all variables on the basis of their upper bounds. By comparison, the new cost allocation algorithms essentially allocate a fixed charge when the variable is zero. This is consistent with the necessary conditions required for a solution to (ALPM) to be optimal.

The new cost allocation algorithms test only specific fixed charge variables. Other positive fixed charge variables are allowed to increase with no additional penalty. If, for example, a warehouse was closed and its customers were handled by other warehouses already open, there would be no increase in their fixed charges.

The cost allocation algorithms provide a means of solving the fixed charge problem which is consistent with the nature of the problem. The algorithm will work well for mixed fixed charge problems as it leaves the task of calculating the values for ordinary variables to linear programming. The algorithm is intended to be used with the revised simplex linear programming algorithm with the product form of the inverse. This is suited to large problems which are typically very sparse. There is minimal increase in the storage requirements of the new cost allocation algorithm over the requirement of the linear programming algorithm. As a result, the new cost allocation algorithms are very efficient for solving large general fixed charge problems.

CHAPTER 4

RESULTS

4.1. Overview

Within the different areas of application defined in Chapter 2, fixed charge problems can be classified into a number of types with varying degrees of specialized structures. Considerable success has been achieved at solving problems which can be described completely by a specialized structure. However, many problems include additional features which require a general formulation and solution technique. While a number of techniques are available for solving general fixed charge problems, the applications of the general solution techniques are limited to small problems. The new cost allocation algorithms (COAL), are developed specifically for large and general fixed charge problems where there has been a lack of successful applications.

In order to demonstrate the success of an algorithm in coping with problem structures, sample problems will be taken from different application areas. The different types of problems within an application area have similar structures. These structures are best exemplified by the specialized formulations such as the capacitated warehouse location problem, (CWLP), or the fixed charge lot size problem, (FCLSP). However, problems requiring a general formulation due to additional complexities retain the nature of the specialized formulations.

The evaluation of an algorithm should examine the impact of other features which have been reported in the literature to cause difficulties. These include the magnitude of the fixed charge component, the number of fixed charge variables, the tightness of capacity constraints, as well as the size of the problem. The ability to consistently generate good solutions to problems allowing for variations in these features with different structures will demonstrate the robustness of a particular algorithm.

The main objective of this chapter is to evaluate the new cost allocation algorithms for solving large general fixed charge problems. Also, the evaluation will compare the different methods used for solving general fixed charge problems outlined in Chapter 2 and the cost allocation algorithms presented in Chapter 3. The alternative approximate algorithms are implemented as accurately and efficiently as possible on the same computer as the as the new COAL algorithms. The testing of the alternative approximate algorithms available for solving large general fixed charge problems has not been carried out for a large variety of problems.

The selection of the test problems is of critical importance to the evaluation in order to avoid any biases. The test results can be used to evaluate the robustness of an algorithm across the different problem types. Further, the evaluation should assess the conditions when an algorithm will perform well.

4.2. Selection of Test Problems

The new cost allocation algorithms are intended for large and general fixed charge problems. However, the area described by large general fixed charge problems includes a wide variety of problem types with a number of different structures. A particular structure often makes a problem more difficult for a general purpose algorithm. Problems which have other features as well as a special structure will retain the difficulties inherent in the special structure. Therefore, testing of general purpose algorithms must not be restricted to a particular type of problem but should investigate as many types as is practical. Any biases which consistently favour a particular algorithm should not be allowed.

Following the above guidelines, the test problems used are summarized in Exhibit 4-1. Due to the overlap between the different areas, a categorization of the problems to different areas can not be precise. However, the problems are classified in the following areas: random, facility

Exhibit 4-1: Test Problems - Basic Structures

Problem	Source	Structure
Random	Cooper & Drebes [4]	Random
Facility Location		
-Waste Disposal	Walker et. al. [94]	Transshipment
-Warehouse Location	Rousseau [78]	(CWLP)
-Routing Problem	Rousseau [78]	(FCTP)
-Power Station	Dutton et. al. [21]	Transportation
Production		
-Hierarchical Production Planning	Graves [36] Hax & Golovin [42]	(FCLSP) (FCLSP)
Manpower		
-Variable Workforce	Hax & Golovin [42] & Mangiameli and Krajewski [60]	(FCLSP) & Manpower Balance
-Sales Force Planning	Haehling von Lanzenauer et.al.[38]	Manpower Balance

location, production planning and manpower planning. While a detailed description of each problem is presented in the appropriate section, an overview relating the problems to each other follows. The problems are selected from areas where the fixed charge problem has received the most attention. Applications in other areas, while potentially as interesting, are demonstrations of the use of integer programming concepts in new areas and quite small.

The standard fixed charge problems used for testing the different algorithms are the random problems generated by Cooper and Drebes [13] in 1967. These problems have no structure and the coefficients are selected randomly with a 50% density. The random problems are very small and not real. Nevertheless, they are used as sample problems due to their historical use. Cooper and Drebes method of concatenating the small problems to create larger random problems is also used (Exhibit 2-15).

Fixed charge problems occur in the literature most frequently with facility location and production planning. However, problems from these areas are generally not used for test purposes for evaluating algorithms for general fixed charge problems. Since these problems have wide applicability, they will be used as test problems.

Facility location problems include two actual applications in waste disposal and power station location. To further investigate the impact of different parameters and structures, a number of (CWLP) and (FCTP) problems are included.

Two problems from production planning are included. These problems use an inventory balance and capacity constraints as their basic structure thus differentiating them from facility location.

In addition, two manpower planning problems are tested. One problem involving a production-manpower planning problem includes constraints for the manpower balance in addition to the inventory balance and capacity constraints. The second problem has multiple manpower levels which creates different structural properties.

The impact of other factors in addition to the different structure from various problem types is also investigated. Problem characteristics which have an impact on the difficulty of a fixed charge problem are evaluated as to their impact on the performance of different algorithms. Factors which are related to the particular problem include size, capacity and fixed charge component. A final factor, which relates to differences between problem types, is the significance of a simplex pivot with respect to the fixed charges.

The size of a problem is the most important factor in making a problem difficult to solve. Size can be measured by such factors as the number of equations, fixed charge variables and ordinary continuos variables. Generally, increasing the number of fixed charge variables makes a problem much more difficult to solve. In contrast, increasing the number of ordinary continuos variables (with out a fixed charge) should not have the same degree of impact on the difficulty of solving a problem.

Care must be taken to insure that solution methods that appear to be practical with small sample problems remain so when considerably larger problems are encountered. It is important to avoid an exponential growth in solution times as the problem size increases (Zanakis and Evans[96], Haessler[39]).

To measure the impact of size by itself, changes in the number of equations or variables should not be due to other factors such as a change in the structure of a problem or capacity constraints. For example, problem size is changed in the waste disposal problem and the warehouse location problem by creating additional demand centers and facilities. In addition, the fraction of feasible arcs or demand center-supply center combinations is varied. Both the number of fixed charge variables and ordinary continuous variables are varied and the impact measured. However, it is important that the utilization of capacity remains the same for the different sizes.

Frances et. al. [31] discuss the impact of excess capacity on the difficulty of solving facility location problems. When there is little excess capacity in a system, a fixed charge problem is actually easier to solve. Increasing capacity leads to more flexibility in the problem and it becomes more difficult to find the optimum solution. However, as capacity becomes excessive, the problem becomes easier again and approaches the uncapacitated facility location problem or the uncapacitated lot size problems.

The impact of capacity is examined by varying the production capacity with the same demand in the hierarchical production planning problems and the size of demand with the same capacity in the power generating problem. The ratio of the fixed charge portion to the variable cost portion will be changed which will affect the difficulty of solving. However, this impact is expected to be minor.

The next dimension of difficulty is described by the size of the fixed charge component. Both Kennington [51] and McGinnis [64] indicate that problems with large fixed charge components are more difficult to solve. However, Frances et. al. [31] state that as the fixed charge component gets increasingly large, eventually, the problem becomes easier to solve. Changing the fixed charge component is accomplished by increasing the size of the fixed charges while keeping the other factors constant. The fixed charge component is varied in the production planning and manpower planning problems. The impact of a large fixed charge component can also be observed in the random problems.

The significance of a simplex pivot with respect to the fixed charges is a structural feature which often has a major impact on the performance of different algorithms. For example, a simplex pivot in (FCTP) is usually very significant as it exchanges one fixed charge variable for another. However, in the very similar (CWLP), several simplex pivots may be required to close a warehouse and shift demand to another warehouse. The impact of changing the significance of a simplex pivot has to be evaluated across problem types.

These different dimensions of difficulty are summarized Exhibit 4-2 including the problem sets where the impact of the particular aspect is tested. To be effective, a solution technique for large general fixed charge problems should perform well across these different classifications. The test problems which have been selected will enable an evaluation of an algorithm along the different dimensions of difficulty.

The different factors will have an impact on the performance of different algorithms. Other factors which contribute to performance such as computer speed, programming language or methods of manipulating the equations should be controlled when evaluating different algorithms in order to isolate the impact of such factors as size, capacity utilization, fixed charge component or structural differences.

4.3. Selection of Algorithms

Solving a general fixed charge problem involves a choice between techniques which can produce an optimal solution or techniques which produce a good but not necessarily optimal solution.

Since all the algorithms are based on linear programming, results will be reported for a linear

Exhibit 4-2: Dimensions of Difficulty

Problem Characteristic	Measurement	Evaluation (Problem Set)
Size	Number of -Fixed Charges -Equations -Variables	-Random -Waste Disposal -Warehouse Location
Capacity	Ratio of Demand to Supply	-Power Station -Production Planning
Fixed Charge Component	Size of Fixed Charges	-Random -Production Planning -Manpower Planning
Significance of Simplex Pivot	Problem Structure	-(FCTP) vs. (CWLP) vs. Waste Disposal -Production Planning vs. Manpower Planning

programming algorithm which solves the associated linear programming problem, (ALP), and can be used as a guide for the evaluation of other algorithms. The linear programming algorithm will use the revised simplex method with the product form of the inverse which will be incorporated into most of the algorithms used.

Solution techniques developed specifically for fixed charge problems and capable of generating an optimal solution include branch and bound, vertex generation and cutting planes. In addition, the various techniques for mixed-integer programming can obtain the optimal solution. However, the standard approach for obtaining an optimal solution to general fixed charge problems is to use a branch and bound mixed-integer programming algorithm. Therefore, the branch and bound mixed integer programming algorithm (BBMIP) from the Multi-Purpose Optimization System, MPOS [15], is used for a number of problems to generate the optimal solution. This algorithm is selected due to its availability and used to measure the performance of the most widely used optimizing technique for general fixed charge problems.

The other option for solving fixed charge problems is to use an approximate method including the Balinski [4] approximation or one of the adjacent extreme point techniques developed by Cooper and Drebes [13], Denzler [19], Hiraki [44], Steinberg [82] or Walker [93]. The Balinski approximation is suited to those problems which have a good estimate for the upper limits. Since problems with good estimates for upper bounds are specialized fixed charge problems, the Balinski approximation is not evaluated. The approximate methods are selected from the adjacent extreme point algorithms.

The first phase of the majority of the adjacent extreme point methods involves obtaining the initial fixed charge local optimum (F.C.L.O.). Starting with the solution to associated linear programming problem (ALP), F.C.L.O. examines all adjacent extreme points (within one simplex pivot) and selects the best. This process is repeated until no further improvement can be made. F.C.L.O. is included to demonstrate the performance of the first phase of the adjacent extreme point methods.

Steinberg [83] recommends two of his own algorithms, Heuristic 1 and 2, and two of Walker's [93] algorithms, Swift 1 and 2, as providing the best methods for solving large fixed charge problems. The four algorithms will be evaluated. The Walker, Steinberg and Hiraki [44] algorithms are also the most recent. However, Hiraki's approach appears to be less suitable for large problems than the two similar Walker algorithms and is excluded. The two Steinberg algorithms are given in Appendices A and B and the two Walker heuristics are given in Appendices C and D.

Of course, the four different variations of the new cost allocation technique, the basic COAL-b, the extended COAL-x, and the special cases COAL-c and COAL-f, are included. Each algorithm represents a particular combination of components from the overall cost allocation technique. Results from these four methods will allow an evaluation of the different components in the cost allocation technique.

The algorithms which will be used for solving the different test problems are summarized in Exhibit 4-3. The algorithms are programmed in Fortran-77 to run on a Control Data Cyber 170/173. Fortran is the obvious choice for implementing a mathematical algorithm. With the exception of BBMIP, all the algorithms use the same routines for various matrix operations based on the revised simplex method with the product form of the inverse. The revised simplex using the product form of the inverse is the standard procedure for solving large linear programming problems. Every effort is made to program the different algorithms as efficiently as possible. However, the fact that the different algorithms use the same basic routines for matrix manipulation as well as the same programming language and computer makes a more valid comparison.

However, measurement of the performance of an algorithm is not straight forward. Performance of approximate and optimization techniques involves a trade off between the quality of a solution and the resources required to obtain the solution. This trade-off requires the development of a performance criteria.

4.4. Performance Criteria

The purpose behind any solution technique is to obtain a good solution to a problem with a reasonable expenditure

Exhibit 4-3: Algorithms for Solving Test Problems

Linear Programming .... L.P.

Optimal Solution

Branch and Bound
Mixed Integer Programming .... BBMIP

Approximate Solution Techniques

Adjacent Extreme Point Algorithms

Initial Fixed Charge
Local Optimum .... F.C.L.O.

Steinberg -- Heuristic 1 .... DIS H 1
Steinberg -- Heuristic 2 .... DIS H 2

Walker -- Swift 1 .... Swift 1
Walker -- Swift 2 .... Swift 2

Cost Allocation Algorithms

Basic Cost Allocation .... COAL-b
Extended Cost Allocation .... COAL-x
Special Case -- Continuos Cost ... COAL-c
Special Case -- Fixed Charges .... COAL-f

of resources. Typically, a trade off must be made between improving the quality of the solution and the effort required. Clearlv, the optimization techniques develop the best quality for their solutions. However, the resources required for large problems may rule optimization techniques out as an effective means of solving the problem. On the other hand, solving (ALP) will provide a solution to a fixed charge problem while using a minimum of resources but the quality of the solution is typically poor.

The evaluation process is complicated by different problem structures and other factors which have an impact on the performance of a solution technique. The evaluation process should identify which techniques work best for the different problem types.

4.4.1. Quality

Quality of a solution refers to the value of the objective function obtained by an algorithm relative to the optimal value of the objective function. The comparison is typically measured as a per cent deviation from the optimum. Of course, the per cent deviation from the optimum can not be used to develop an absolute criteria to define "good" which must be evaluated on a problem by problem basis.

The optimal solution is used, provided it is available. If not, either a lower bound on the objective function or the best solution obtained will have to suffice. While the optimization techniques generate a lower bound on the objective function, the approximate methods do not. Since a lower bound will not be available for all problems, the best solution available will be used for those problems for which the optimal solution is not available.

Another measure of the quality of the solution would be the relative frequency of obtaining the optimal solution. However, both these measures would tend to rank the various algorithms the in a similar fashion. Good performance on one measure usually implies good performance on the other measure. Therefore, only the per cent deviation from the objective function value of the optimal solution will be used.

Algorithms should be relatively consistent within the problem types for which they are intended. Ideally, there should not be some problems where good quality solutions are obtained and others where poor solutions are obtained. One method for evaluating the consistency within problem types is to examine the maximum deviations for problems within a type. Normally, algorithms which perform well when using an average over a number of problems will have a low maximum deviation as well. However, an evaluation of an approximate algorithm should provide some indication of the maximum deviation likely to be encountered.

4.4.2. Resource Requirements

The second factor for evaluating the performance of an algorithm involves the resources required to obtain a solution. Two components are involved in determining computer resources required: the first involves central processor time and the second involves memory requirements. The first component, central processor time, is more significant. Normally, the methods which produce approximate solutions are much faster than the methods which produce an optimal solution. The approximate methods provide satisfactory solutions to large problems while minimizing resource requirements. Unless stated otherwise, all times reported are on a Control Data Cyber 170/173.

The second component involved in computer resources relates to the memory required to store the problem. This is relevant when the memory requirements effectively lengthen the solution times to impractical levels or exceed available space. The different methods selected for obtaining approximate solutions have memory requirements which are slightly larger than the memory required by linear programming and do not need to be evaluated separately. However, some of the optimizing algorithms and BBMIP, in particular, can require an enormous amount of memory. Although much of this memory can be on a peripheral device, this may result in dramatically lengthening the elapsed time to get a solution.

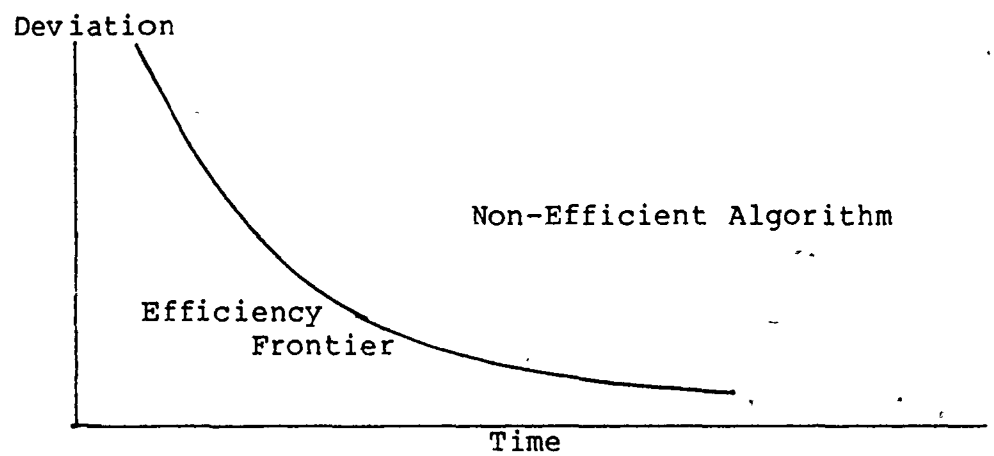
4.4.3. Efficiency Frontier

In summary, two factors will be used to measure the performance of the different algorithms. The first factor, representing the quality of the solution, will be the deviation from the optimal solution. The second factor, representing resource requirements will be the central processing time required for each algorithm. There will be a trade off between obtaining a better quality solution versus the computer resources required.

For each problem set, the algorithms which produce a better solution using less computer time will be referred to as efficient. This will define an efficiency frontier for a particular problem set (Exhibit 4-4). Algorithms on the efficiency frontier would be appropriate for solving the particular type of problem given a desired limit on the expenditure of computer resources. However, the non-efficient algorithms would not be appropriate as other algorithm will produce an equal or better quality solution and require less computer time. Particular algorithms which produce solutions quickly with lower quality will be on the upper left part of the efficiency frontier. Other algorithms will require more time to obtain better quality and will be on the lower right portion of the efficiency frontier. Non-efficient algorithms will be to the right and above the efficiency frontier.

The results for each problem set will consist of the quality of the solutions and the resource requirements. One representative efficiency frontier will be constructed using the quality and resource requirements. The efficient algorithms across the problem set will also be identified. An evaluation with in each problem type can be made indicating where particular algorithms are efficient by the positioning with respect to the efficiency frontier. Finally, an overall evaluation of the consistency of the

Exhibit 4-4: Efficiency Frontier



algorithms across the different problem types will be made.

4.5. Random Problems

Cooper and Drebes [13] use a number of small randomly generated fixed charge problems with five equations and ten fixed charge variables. The problems have no other structure. The equation coefficients are randomly selected in a range of $\pm$ 20 with a 50% density. These problems are also used by Steinberg [82] and Walker [93] to test their algorithms as well as Denzler [19], Murty [70] and McKeown [65, 66, 67] for their algorithms. Fifteen of these problems are used as sample problems and are given in Appendix E. Cooper and Drebes create larger problems by concantenating smaller problems as illustrated in Exhibit 2-15. Three larger problems are created each consisting of five sub-problems. Also, three problems are created each with ten sub-problems and one problem is created with all fifteen sub-problems.

Each problem is solved with greater than constraints (">"), equality constraints ("=") and less than constraints ("<"). The problems with ">" and "=" constraints are used by others. However, the problems with "<" constraints which are maximized have not been used by others and adds a new dimension to the random problems.

BBMIP is not used on these problems. The optimal solution for the small problems with five equations is obtained by inspecting all possible solutions. The optimal solutions for the larger problems is easily derived from the solutions to the small problems. Consequently, the results for BBMIP are denoted as -na- or not available.

4.5.1. Random Problems with ">" Constraints

The quality of the solutions for the ">" problems is given in Exhibit 4-5 which shows the average per cent deviation from the optimal solution plus the maximum deviation from the optimal solution for the four different sizes of problems. All the algorithms obtain the same solution for each small problem in the different sizes. The small variation is caused by the averaging process resulting from the concantenating of the sub-problems. With the fifteen problems in the 5x15 size, the average deviation is calculated by summing the per cent deviation for each problem and dividing by fifteen. For the one 75x225 size problem, the average deviation is effectively calculated by summing the fifteen objective function values of the solutions obtained by the algorithm and dividing by the sum of the fifteen objective function value of the optimal solutions. The averaging process also accounts for the reduction in maximum deviation with increasing problem size. The averaging procedure has problems with this problem set. Since it is the standard procedure, it will still be used. These problems are not encountered in the other problems in facility location, production planning or manpower planning.

COAL-x is the only algorithm to successfully solve all the ">" problems. The other approximate methods are close to the optimum while the linear programming solution is poor. The same relative standing for the different algorithms on quality of solution is obtained by the average deviation and the maximum deviation.

The average solution times required by the different algorithms for the ">" problems are given in Exhibit 4-6. As expected, COAL-x takes considerably longer than COAL-b which, in turn, takes longer than COAL-c and COAL-f. L.P. and the initial fixed charge local optimum, F.C.L.O., are respectively the fastest and second fastest. The increase in solution time is roughly proportional to the square of the size.

A representative efficiency frontier using the 50x150 size, 10 concantenated sub-problems, is constructed in Exhibit 4-7 using the average deviation from Exhibit 4-5 and the average solution times from Exhibit 4-6. The efficient algorithms include L.P., F.C.L.O., COAL-f, COAL-c and COAL-x. Although COAL-b combines both COAL-f and COAL-c, it is not efficient on its own. COAL-c usually produces the better solution in each case and COAL-b does not improve the solution for this problem set. Although L.P. is not shown on the graph, it is always "efficient" far in the upper left as it takes less time than any other algorithm.

Efficiency frontiers for the other sizes would be similar and are therefore not presented. However, a table of efficient algorithms for each size is given in Exhibit 4-8 with similar results for each size. For the small problems, COAL-c is faster than COAL-f. Thus COAL-f is not on the efficiency frontier.

The new cost allocation algorithms clearly dominate the Steinberg and Walker algorithms for this problem set. The initial Fixed Charge Local Optimum, however, does produce an efficient solution in the upper left segment of poor quality. COAL-f and COAL-c have a large increase in quality with a increase in solution time with respect to F.C.L.O. COAL-x has a smaller increase in quality with a large

Exhibit 4-5: Random Problems with ">" Constraints
-- Quality of Solution

Average Deviation from Optimum (%)

Size	5x15	25x75	50x150	75x225
L.P.	41.37	39.85	39.75	39.72
BBMIP	-na-	-na-	-na-	-na-
F.C.L.O.	2.10	2.28	2.23	2.21
DIS H 1	1.66	2.28	2.23	2.21
DIS H 2	1.62	1.34	1.31	1.30
Swift 1	1.66	1.82	1.78	1.77
Swift 2	1.66	1.82	1.78	1.77
COAL-b	.53	.56	.55	.54
COAL-x	.00	.00	.00	.00
COAL-c	.53	.56	.55	.54
COAL-f	1.68	1.76	1.78	1.79

Maximum Deviation from Optimum (%)

Size	5x15	25x75	50x150	75x225
L.P.	130.01	47.748	42.42	39.72
BBMIP	-na-	-na-	-na-	-na-
F.C.L.O.	14.64	4.47	3.42	2.21
DIS H 1	14.64	4.47	3.42	2.21
DIS H 2	14.17	2.36	2.01	1.30
Swift 1	14.64	3.10	2.73	1.77
Swift 2	14.64	3.10	2.73	1.77
COAL-b	7.89	1.67	.84	.54
COAL-x	.00	.00	.00	.00
COAL-c	7.89	1.67	.84	.54
COAL-f	14.22	2.69	2.60	1.79

Problems	15	3	3	1
Size-Equations	5	25	50	75
-Variables	15	75	150	225
-Fixed Charges	10	50	100	150

Exhibit 4-6: Random Problems with ">" Constraints
-- Resource Requirements

	Solution Times (cpu sec.)			
Size	5x15	25x75	50x150	75x225
L.P.	.06	.81	4.17	5.08
BBMIP	-na-	-na-	-na-	-na-
F.C.L.O.	.13	2.35	24.05	35.54
DIS H 1	.23	7.68	73.04	107.40
DIS H 2	.69	38.09	559.33	898.39
Swift 1	.40	18.94	303.66	493.04
Swift 2	.42	21.72	304.73	492.14
COAL-b	.39	12.00	143.44	214.75
COAL-x	.56	30.52	551.55	905.86
COAL-c	.20	6.04	78.52	119.08
COAL-f	.22	5.90	63.90	93.73
Problems	15	3	3	1
Size-Equations	5	25	50	75
-Variables	15	75	150	225
-Fixed Charges	10	50	100	150

Exhibit 4-7: Random Problems with ">" Constraints - Size 50 x 150 -- Efficiency Frontier

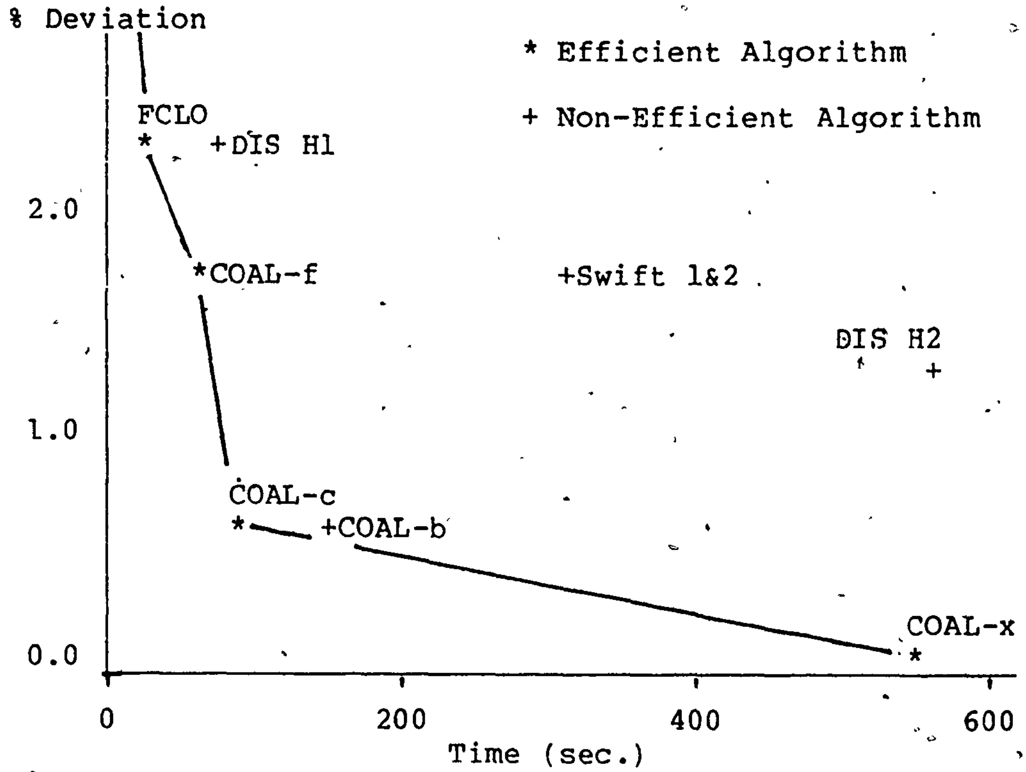


Exhibit 4-8: Random Problems with ">" Constraints
-- Efficient Algorithms using Averages

Size	5x15	25x75	50x150	75x225
L.P.	*	*	*	*
BBMIP	-na-	-na-	-na-	-na-
F.C.L.O.	*	*	*	*
DIS H 1	-	-	-	-
DIS H 2	-	-	-	-
Swift 1	-	-	-	-
Swift 2	-	-	-	-
COAL-b	-	-	-	-
COAL-x	*	*	*	*
COAL-c	*	*	*	*
COAL-f	-	*	*	*

* Efficient Algorithm (Best Quality for Solution Time)

- Non-efficient Algorithm (Alternative algorithm obtains a better quality solution in less time)

increase in solution time with respect to COAL-c.

4.5.2. Random Problems with "=" Constraints

A similar analysis is carried out for the "=" problems with the quality of solutions reported in Exhibit 4-9. Both Walker algorithms, one Steinberg algorithm and COAL-x obtain the optimum solution for all problems while the other approximate methods are relatively close. As in the ">" problems, the same solutions are found for all sizes of problems. of the approximate algorithms, COAL-c has the largest deviation from the optimum. The initial search procedure for COAL-c which examines one fixed charge variable at a time often leads it to a poor solution. COAL-f, with a more global perspective initially, often avoids such solutions.

The resource requirements measured in cpu seconds is given in Exhibit 4-10. COAL-x takes considerably longer than other algorithms. Steinberg's Heuristic 2 and the Walker algorithms take much less time for the "=" problems than the ">" problems. In order to handle the slack variables, the ">" problems have 50% more variables than the "=" problems. Since the adjacent extreme point algorithms do not differentiate between continuous and fixed charge variables, the solution times are affected significantly by the additional slack variables for the ">" equations. COAL-x is hampered by requiring an extra effort to improve the solution developed by COAL-c. The quasi-sufficiency test for a combination change is considerably slower than the single change test and is required to do extra improvements to the solution developed by COAL-c.

A representative efficiency frontier is shown in Exhibit 4-11 for the 50x100 size with 3 problems, each generated from 10 sub-problems. The efficient algorithms are L.P., F.C.L.O., COAL-f and DIS H 2. Of the cost allocation algorithms, only COAL-f manages to be efficient and is only marginally better than F.C.L.O. A table of efficient algorithms is presented for the different sizes in Exhibit 4-12. The Swift algorithms, which are quite close to Steinberg's Heuristic 2, are on the efficiency frontier for the smaller sizes.

4.5:3. Random Problems with "<" Constraints

The quality of solutions for the final set of random problems using the "<" equations is given in Exhibit 4-13. Only the two Walker algorithms obtain the optimal solution for all problems. The other algorithms are relatively close with COAL-c being the furthest from the optimum. Again, the improvement in quality of solution as size increases is an illusion. With the fifteen small problems, there are solutions with a small absolute value for the objective function. Hence, deviations are very large and when arithmetically averaged, severally degrade the performance

Exhibit 4-9: Random Problems with "=" Constraints
-- Quality of Solution

Average Deviation from Optimum*(%)

Size	5x10	25x50	50x100	75x150
L.P.	19.52	19.33	19.35	19.35
BBMIP	-na-	-na-	-na-	-na-
F.C.L.O.	.37	.42	.42	.42
DIS H 1	.37	.42	.42	.42
DIS H 2	.00	.00	.00	.00
Swift 1	.00	.00	.00	.00
Swift 2	.00	.00	.00	.00
COAL-b	.39	.38	.38	.38
COAL-x	.00	.00	.00	.00
COAL-c	1.23	1.27	1.26	1.26
COAL-f	.39	.38	.38	.38

Maximum Deviation from Optimum (%)

Size	5x10	25x50	50x100	75x150
L.P.	57.97	24.48	21.40	19.35
BBMIP	-na-	-na-	-na-	-na-
F.C.L.O.	5.62	1.26	.63	.42
DIS H 1	5.62	1.26	.63	.42
DIS H 2	.00	.00	.00	.00
Swift 1	.00	.00	.00	.00
Swift 2	.00	.00	.00	.00
COAL-b	1.25	.92	.57	.38
COAL-x	.00	.00	.00	.00
COAL-c	8.41	1.77	1.73	1.26
COAL-f	4.53	.92	.57	.38
Problems	15	3	3	1
Size-Equations	5	25	50	75
-Variables	10	50	100	150
-Fixed Charges	10	50	100	150

Exhibit 4-10: Random Problems with "=" Constraints
-- Resource Requirements

Size	Solution Times (cpu sec.) 5x10	25x50	50x100	75x150
L.P.	.06	.67	3.44	4.21
BBMIP	-na-	-na-	-na-	-na-
F.C.L.O.	.12	1.86	18.06	26.40
DIS H 1	.20	4.49	49.85	72.05
DIS H 2	.40	11.53	133.61	202.35
Swift 1	.27	9.21	146.81	239.31
Swift 2	.30	9.93	146.86	238.65
COAL-b	.32	10.60	128.90	196.22
COAL-x	.49	35.48	806.01	1397.54
COAL-c	.16	4.66	57.20	86.87
COAL-f	.20	5.97	68.27	102.28
Problems	15	3	3	1
Size-Equations	5	25	50	75
-Variables	10	50	100	150
-Fixed Charges	10	50	100	150

Exhibit 4-11: Random Problems with "=" Constraints
- Size 50x100 -- Efficiency Frontier

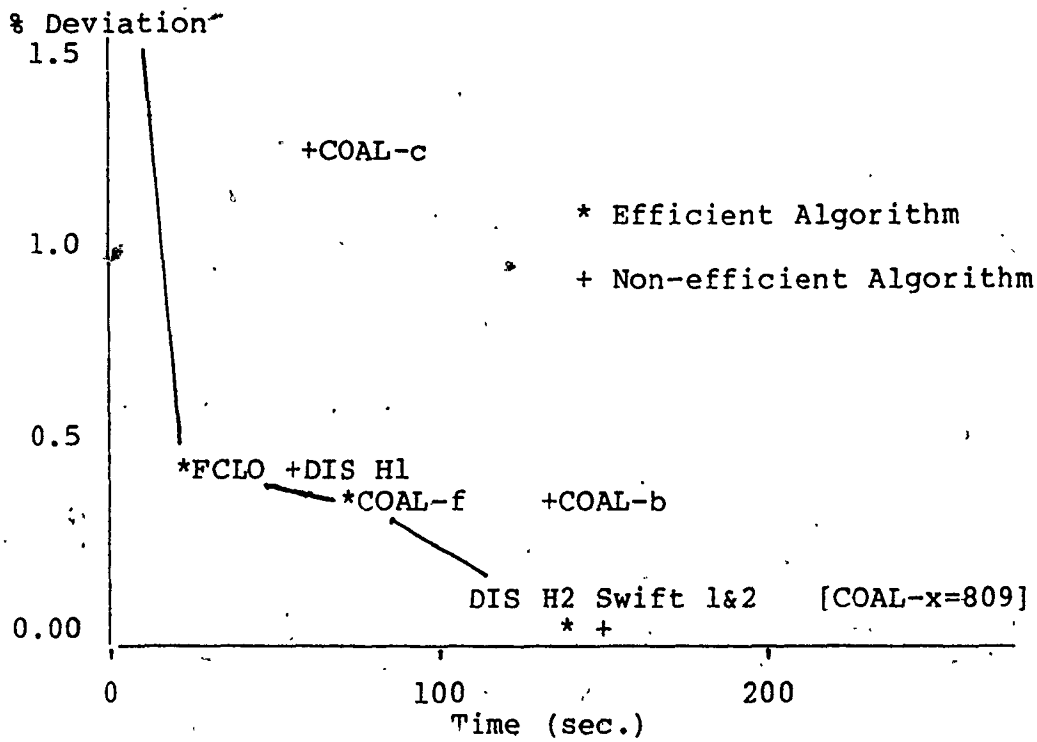


Exhibit 4-12: Random Problems with "=" Constraints
-- Efficient Algorithms using Averages

Size	5x10	25x50	50x100	75x150
L.P.	*	*	*	*
BBMIP	-na-	-na-	-na-	-na-
F.C.L.O.	*	*	*	*
DIS H 1	-	-	-	-
DIS H 2	-	-	*	*
Swift 1	*	*	-	-
Swift 2	-	-	-	-
COAL-b	-	-	-	-
COAL-x	-	-	-	-
COAL-c	-	-	-	-
COAL-f	-	*	*	*

* Efficient Algorithm (Best Quality for Solution Time)

\- Non-efficient Algorithm (Alternative algorithm obtains a better quality solution in less time)

on very small problems. With the larger problems, the objective function values are added before the deviations are calculated. Those problems with large per cent deviations have quite small absolute deviations. Hence, the apparent improvement.

The solution times required for the "<" problems are given in Exhibit 4-14. Steinberg's Heuristic 2 and Walker's two algorithms require more time with the "<" problems than with the ">" problems. However, the cost allocation algorithms require less time. Even COAL-x is considerably faster than DIS H 2, Swift 1 or Swift 2. The fixed charge component for the "<" problems is quite small with respect to the continuos component. While the adjacent extreme point algorithms essentially ignore the impact of changes in size of the fixed charges, the cost allocation algorithms implicitly exploit low values for the fixed charges.

The efficiency frontier for the 50x150 size problems is plotted in Exhibit 4-15. The efficient algorithms are L.P., F.C.L.O., COAL-f and Swift 1. Steinberg's Heuristics are dominated by others. COAL-c, although not requiring much computer time, gets a poor solution. COAL-b and COAL-x do not improve on COAL-f. Similar results are obtained for efficient algorithms for the other sizes of "=" problems (Exhibit 4-16).

Exhibit 4-13: Random Problems with "<" Constraints -- Quality of Solution

Average Deviation from Optimum (%)

Size	5x15	25x75	50x150	75x225
L.P.	185.09	6.19	6.05	6.05
BBMIP	-na-	-na-	-na-	-na-
F.C.L.O.	.95	.75	.50	.45
DIS H 1	.56	.75	.50	.45
DIS H 2	.95	.39	.26	.23
Swift 1	.00	.00	.00	.00
Swift 2	.00	.00	.00	.00
COAL-b	.38	.29	.19	.17
COAL-x	.38	.29	.19	.17
COAL-c	1.32	1.18	1.33	1.40
COAL-f	.38	.29	.19	.17

Maximum Deviation from Optimum (%)

Size	5x15	25x75	50x150	75x225
L.P.	2576.30	7.40	6.71	6.01
BBMIP	-na-	-na-	-na-	-na-
F.C.L.O.	8.45	2.26	.83	.45
DIS H 1	8.45	2.26	.83	.45
DIS H 2	8.45	1.17	.43	.23
Swift 1	.00	.00	.00	.00
Swift 2	.00	.00	.00	.00
COAL-b	5.76	.87	.32	.17
COAL-x	5.76	.87	.32	.17
COAL-c	14.09	2.66	2.12	1.40
COAL-f	5.76	.87	.32	.17
Problems	15	3	3	1
Size-Equations	5	25	50	75
-Variables	15	75	150	225
-Fixed Charges	10	50	100	150

Exhibit 4-14: Random Problems with "<" Constraints
-- Resource Requirements

Solution Times (cpu sec.)

Size	5x15	25x75	50x150	75x225
L.P.	.07	.63	2.90	3.51
BBMIP	-na-	-na-	-na-	-na-
F.C.L.O.	.12	1.59	13.21	18.81
DIS H 1	.21	4.66	49.19	69.78
DIS H 2	.88	63.77	1386.59	2389.46
Swift 1	.47	25.31	408.75	669.33
Swift 2	.58	41.18	452.93	692.03
COAL-b	.24	4.68	51.51	78.38
COAL-x	.32	11.97	147.70	223.52
COAL-c	.12	1.93	23.69	37.52
COAL-f	.15	2.87	28.09	40.98
Problems	15	3	3	1
Size-Equations	5	25	50	75
-Variables	15	75	150	225
-Fixed Charges	10	50	100	150

Exhibit 4-15: Random Problems with "<" Constraints
- Size 50x150 -- Efficiency Frontier

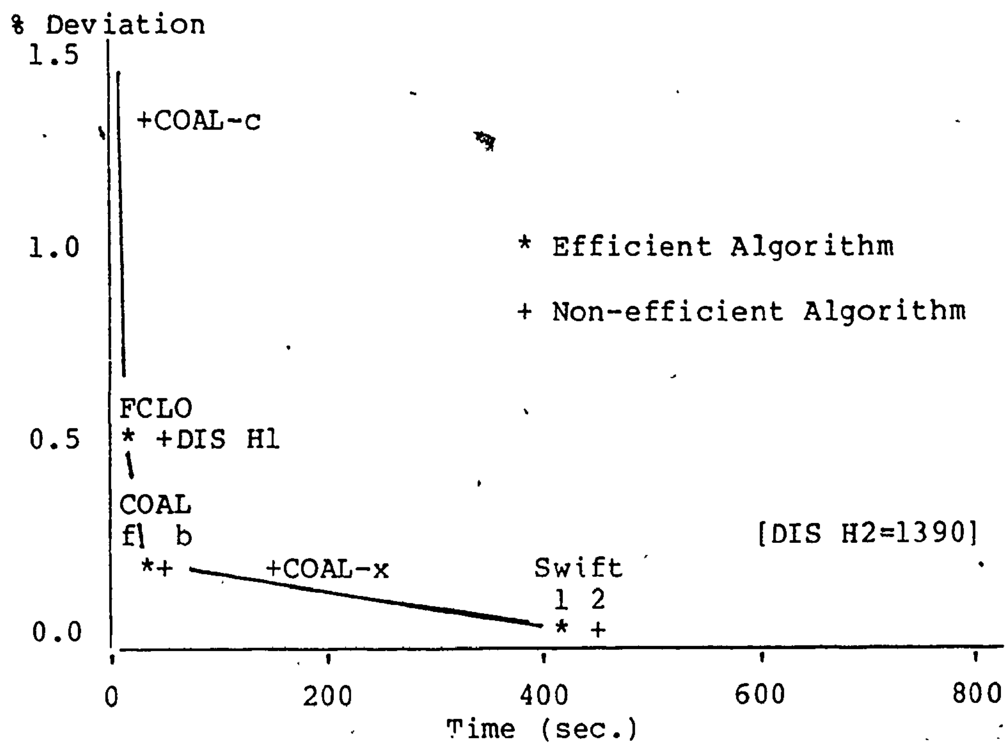


Exhibit 4-16: Random Problems with "<" Constraints
-- Efficient Algorithms using Averages

Size	5x15	25x75	50x150	75x225
L.P.	*	*	*	*
BBMIP	-na-	-na-	-na-	-na-
F.C.L.O.	*	*	*	*
DIS H 1	-	-	-	-
DIS H 2	-	-	-	-
Swift 1	*	*	[illegible]	*
Swift 2	-	-	-	-
COAL-b	-	-	-	-
COAL-x	-	-	-	-
COAL-c	-	-	-	-
COAL-f	*	*	*	*

* Efficient Algorithm (Best Quality for Solution Time)

- Non-efficient Algorithm (Alternative algorithm obtains a better quality solution in less time)

4.5.4. Evaluation - Random Problems

A summary of the efficient algorithms for the three types of random problems given in Exhibit 4-17. L.P., F.C.L.O., and COAL-f are efficient for all types. L.P. and F.C.L.O. obtain solutions quickly but of low quality. COAL-f obtains relatively good solutions for the three types of random problems. Steinberg's Heuristic 2 is efficient for the "=" problems with the two Swift algorithms very close. The Swift algorithms are efficient for the "<" problems but take considerably more time than the cost allocation algorithms.

The cost allocation algorithms clearly dominate Steinberg's Heuristic 2 and the two Walker algorithms for the ">" equations. For the "=" equations, the cost allocation algorithms are quite close in performance to DIS H 2 and Swift 1 and 2. For the problems with the "<" equations, the cost allocation algorithms are in the center of the efficiency frontier. The Walker algorithms obtain a small increase in quality over the cost allocation algorithms with a large increase in time.

The number of continuous variables and the size of the fixed charge component as well as the size of the problem have an impact on the performance of the different algorithms. Further analysis of the impact of these factors is deferred to the next section. While an analysis could be carried out for the random problems, the process of generating larger problems would raise doubts about the validity of any results produced.

The cost allocation algorithms perform on or close to the efficiency frontier for all the different random problem types. While the random problems are useful for evaluating the performance of different algorithms, they are not actual applications. In the following sections, various problems derived from actual applications will be used to insure that the algorithms are not inhibited by the structure inherent in a applied problem.

Exhibit 4-17: Random Problems -- Summary
-- Efficient Algorithms

	Random Problems ">"	"="	"<"
L.P.	*	*	*
BBMIP	-na-	-na-	-na-
F.C.L.O.	*	*	*
DIS H 1	-	-	-
DIS H 2	-	*	-
Swift 1	-	-	*
Swift 2	-	-	-
COAL-b	-	-	-
COAL-x	*	-	-
COAL-c	*	-	-
COAL-f	*	*	*

4.6. Facility Location

Facility location problems are common applications of fixed charge problems. A waste disposal problem from Walker et. al.[94] is used as an example of such a problem and compared with a capacitated warehouse location problem, (CWLP), and a fixed cost transportation problem, (FCTP). In addition, an application in the location of power generating stations is examined.

4.6.1. Waste Disposal Problem

The waste disposal problem (WDP) from Walker et. al. [94] is given in Exhibit 4-18. In order to insure that the results are not due to the particular parameters of the problem, similar problems are generated with the same cost, demands and capacities. However, the locations of the demand centers and the treatment centers are varied within the same size area. Larger problems are generated by doubling (Size 2) and tripling (Size 3) both the number of waste generating centers and intermediate and final treatment centers within the same size area. The number of variables in the problems is changed by varying the number of feasible generating center-treatment center combinations from 40% to 100% of the possible combinations (Appendix G).

The quality of the solutions is given in Exhibit 4-19. The results are organized into three columns corresponding

Exhibit 4-18: Waste Disposal Problem

(WDP) minimize $z = \sum_{ij} c_{ij} X_{ij} + \sum_{kl} b_{kl} T_{kl} + \sum_{j} f_j Y_j$

subject to:

$$\sum_j X_{ij} = d_i \qquad \forall \; i$$

$$\sum_i X_{ik} \leq q_k Y_k \qquad \forall \; k$$

$$\sum_i X_{ik} = \sum_l a_k T_{kl} \qquad \forall \; k$$

$$\sum_i X_{il} + \sum_k T_{kl} \leq q_l Y_l \qquad \forall \; l$$

$$X_{ij}, T_{kl} \geq 0 \qquad \forall \; i,j,k,l$$

$$Y_j = 0,1 \qquad \forall \; j$$

where:

i = index of a waste generating center
j = index of a treatment center
k = index of an intermediate treatment center
l = index of a final treatment center

X_{ij} = amount of waste from center i shipped to treatment center j.

T_{kl} = amount of treated waste from intermediate center k shipped to final treatment center j.

Y_j = 0-1 variable indicating if treatment center j is operating.

d_i = waste generated at center i.

q_j = capacity of treatment center j.

a_k = fraction of treated waste sent to a final treatment site from intermediate treatment center k.

c_{ij} = variable cost of shipping and treating waste from i to treatment center j.

b_{kl} = variable cost of shipping and treating waste from intermediate treatment center k at final treatment center l.

f_j = fixed cost of treatment center j.

---

to the number of times the demand centers and treatment centers are repeated. Steinberg's Heuristic 2 requires extremely long computer runs to solve these problems and is not used for the Size 2 and Size 3 problems. In order to reduce execution time for the Walker algorithms, the Size 3 problems have an arc density of only 40%.

BBMIP, of course, has the best quality with no deviation from the optimum. However, the cost allocation algorithms provide better quality solutions than the adjacent extreme point algorithms. As problem size increases, the deviations from the optimum of the solutions of the approximate algorithms become larger.

The average solution times are given in Exhibit 4-20. The cost allocation algorithms are considerably faster than the two Walker algorithms and Steinberg's Heuristic 2. The small increase in execution time for the Walker algorithms between Size 2 and Size 3 can be attributed to the low density of the Size 3 problems and the small increase in the number of variables (304 to 329, respectively). BBMIP solves (WDP) in less time than DIS H 2, Swift 1 or 2.

As a result, the efficiency frontier for the Size 3 problems is dominated by cost allocation algorithms (Exhibit 4-21). Again, L.P. and F.C.L.O. are on the upper left corner using little time but obtaining poor solutions. BBMIP is on the lower right always obtaining the optimum but requiring more time. Similar results for the efficient algorithms for Size 1 and Size 2 are also observed (Exhibit 4-22). COAL-c generates better solutions than COAL-f while COAL-b does not improve on COAL-c.

In the preceding discussion, changes in execution times for different algorithms have been attributed to changes in the size of the problem as well as to the number of variables. The relationship between the two factors and execution time can be analysed by developing equations to relate the solution times to the number of equations and the number of variables. (Note that the number of equations, the number of fixed charges and the size are all directly proportional). The equations are presented in Exhibit 4-23.

The Walker algorithms are dramatically affected by the number of ordinary variables in a problem with an exponent of over 2 while the exponent for the number of equations (representing equations and fixed charges) is than 1. Thus, the solution times required by the Walker algorithms are very dependent upon the number of variables (with or with out a fixed charge) in the problem. The cost allocation algorithms show no relationship with the number of variables

Exhibit 4-19: Waste Disposal Problem
-- Quality of Solution

Average Deviation from Optimum (%)

	Size 1	Size 2	Size 3
L.P.	4.19	9.97	7.90
BBMIP	0.00	0.00	0.00
F.C.L.O.	1.08	3.36	2.75
DIS H 1	.92	3.36	2.75
DIS H 2	.77	-na-	-na-
Swift 1	.01	1.41	1.63
Swift 2	.01	1.41	1.63
COAL-b	.00	.62	.95
COAL-x	.00	.62	.71
COAL-c	.01	.62	.95
COAL-f	.00	1.31	1.34

Maximum Deviation from Optimum (%)

	Size 1	Size 2	Size 3
L.P.	9.90	11.77	10.68
BBMIP	0.00	0.00	0.00
F.C.L.O.	9.15	4.99	4.53
DIS H 1	9.15	4.99	4.53
DIS H 2	9.15	-na-	-na-
Swift 1	.07	1.70	3.03
Swift 2	.07	1.70	3.03
COAL-b	.00	1.48	2.14
COAL-x	.00	1.48	2.14
COAL-c	.07	1.48	2.14
COAL-f	.00	3.07	2.14
Problem	12	7	3
Size-Equations	30	60	90
-Variables	88	304	329
-Fixed Charges	7	14	21

Exhibit 4-20: Waste Disposal Problem
-- Resource Requirements

	Solution Times (cpu sec.)		
	Size 1	Size 2	Size 3
L.P.	1.07	7.42	17.73
BBMIP	1.82	40.00	720.47
F.C.L.O.	.68	5.43	12.95
DIS H 1	3.99	37.42	69.88
DIS H 2	45.90	-na-	-na-
Swift 1	23.13	838.36	1172.01
Swift 2	23.80	713.73	1154.88
COAL-b	1.24	10.59	55.29
COAL-x	1.92	22.58	155.20
COAL-c	.71	5.81	26.13
COAL-f	.51	4.73	24.51
Problem	12	7	3
Size-Equations	30	60	90
-Variables	88	304	329
-Fixed Charges	7	14	21

Exhibit 4-21: Waste Disposal Problem - Size 3
-- Efficiency Frontier

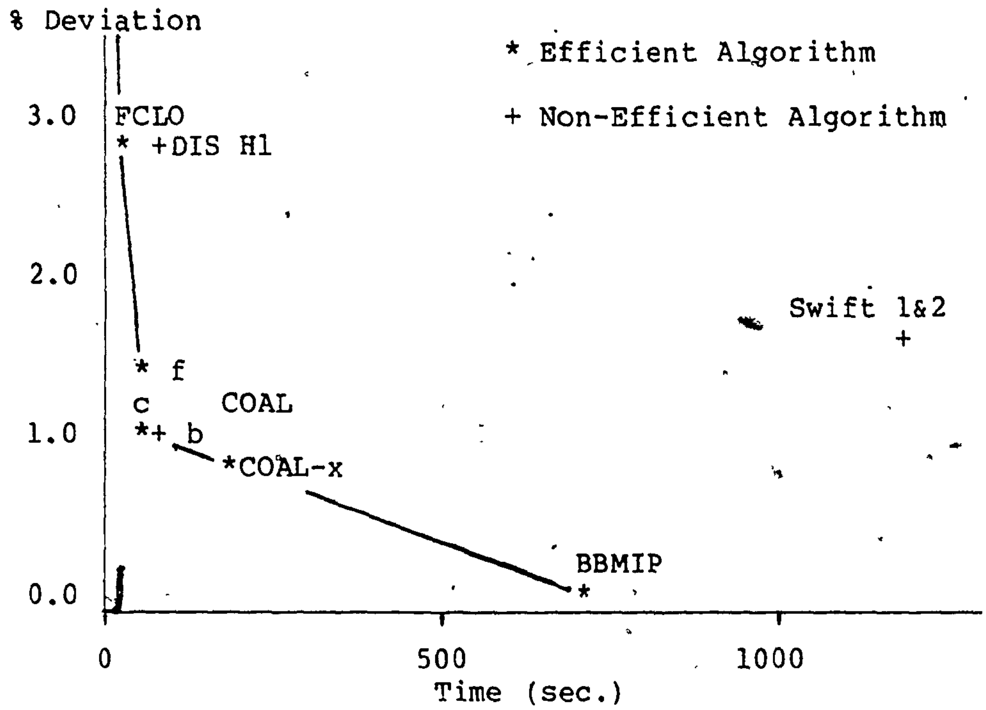


Exhibit 4-22: Waste Disposal Problem
-- Efficient Algorithms using Averages

	Size 1	Size 2	Size 3
L.P.	*	*	*
BBMIP	-	*	*
F.C.L.O.	-	-	*
DIS H 1	-	-	-
DIS H 2	-	-na-	-na-
Swift 1	-	-	-
Swift 2	-	-	-
COAL-b	-	-	-
COAL-x	-	-	*
COAL-c	-	*	*
COAL-f	*	*	*

* Efficient Algorithm (Best Quality for Solution Time)

- Non-efficient Algorithm (Alternative algorithm obtains a better quality solution in less time)

Exhibit 4-23: Waste Disposal Problem -- Relationship Between Size and Solution Time

Algorithm	Equation	r^2 Logged Data	r^2 Original Data
Linear Programming	$t=0.000167\ n^{.639}\ m^{1.73}$	.994	.888
BBMIP	$t=0.0971\ e^{.0193n}\ e^{.0905m}$	.938	.906
Initial Fixed Charge Local Optimum	$t=0.000686\ n^{.750}\ m^{1.71}$	.980	.799
Steinberg Heuristic 1	$t=0.000311\ n^{1.16}\ m^{1.22}$	.977	.955
Heuristic 2	-na-		
Walker Swift 1	$t=0.0000449\ n^{2.46}\ m^{.630}$	.978	.828
Swift 2	$t=0.0000638\ n^{2.23}\ m^{.823}$	.967	.702
COAL-b	$t=0.0000127\ m^{3.32}$	.974	.656
COAL-c	$t=0.0000119\ m^{3.21}$	.975	.768
COAL-f	$t=0.0000040\ m^{3.41}$	.973	.638
COAL-x	$t=0.0000032\ m^{3.85}$	.976	.659

t = time(seconds) m = number of equations
n = number of variables(fixed charge + regular)

r^2 = coefficient of determination

and the number of equations have an exponent of over 3. The marked increase in computer time required by the Walker algorithms can also be attributed to the number of variables being proportional to the square of the size.

All the algorithms with the exception of BBMIP have solution times as a polynomial function of size. By comparison, BBMIP has an exponential relationship indicating that solution times will increase dramatically with size for larger problems.

(WDP) has of course a particular structure. Any conclusions made with respect to the performance of different algorithms do not necessarily apply to other problem types with a different structure. In order to insure that the results observed with (WDP) apply with other problems with different structures, other problem types are analysed.

4.6.2. Capacitated Warehouse Location Problem

The basic structure for many facility location problems is represented by the capacitated warehouse location problem (CWLP). Although there are preferable means for solving, the capacitated warehouse location problems are as difficult to solve by the algorithms for the general fixed problem as a facility location problem which requires a linear programming formulation. Since there are fewer variables in this type of problem than the waste disposal transshipment problem (WDP), the Walker and, hopefully, the Steinberg algorithms will have less difficulty in solving these problems allowing further analysis. A number of capacitated warehouse location problems are generated from a problem from Rousseau [78] (Appendix G). Larger problems are created in the same fashion as in the waste disposal problems for Size 2, Size 3 and Size 4 while the arc density is varied from 30% to 100%.

The quality of the solutions (Exhibit 4-24) are similar to the results from (WDP). The cost allocation algorithms do significantly better than the adjacent extreme point algorithms. For these problems, COAL-f has a lower deviation than COAL-c. COAL-b obtains solutions which are an improvement over both COAL-f and COAL-c.

The solution times (Exhibit 4-25) also have results similar to (WDP) although the adjacent extreme point algorithms use slightly less time. However, the efficiency frontier (Exhibit 4-26) is again dominated by the cost allocation algorithms with L.P. and F.C.L.O., on the upper right and BBMIP on the lower left. As the problem size decreases, BBMIP becomes dominant as it always obtains the optimum and its solution time becomes less than the cost allocation algorithms for small problems.

A similar analysis is made to relate solution times to size and the number of variables allowing further comparisons of the algorithms. The equations are given in

Exhibit 4-24: Capacitated Warehouse Location Problem -- Quality of Solution

Average Deviation from Optimum (%)

	Size 1	Size 2	Size 3	Size 4
L.P.	13.62	19.30	19.46	21.36
BBMIP	0.00	0.00	0.00	0.00
F.C.L.O.	7.23	8.30	10.55	11.41
DIS H 1	3.82	6.02	9.38	10.00
DIS H 2	3.09	[illegible].89	4.40	7.05
Swift 1	2.87	4.56	4.87	6.31
Swift 2	3.15	3.70	4.89	5.88
COAL-b	.00	1.44	1.47	1.21
COAL-x	.00	1.17	1.01	.50
COAL-c	1.13	2.22	3.07	4.12
COAL-f	1.48	3.61	2.33	1.75

Maximum Deviation from Optimum (%)

	Size 1	Size 2	Size 3	Size 4
L.P.	29.54	32.84	30.78	36.02
BBMIP	0.00	0.00	0.00	0.00
F.C.L.O.	16.75	13.61	20.50	18.40
DIS H 1	15.72	12.05	20.00	15.59
DIS H 2	13.42	11.28	9.94	12.58
Swift 1	10.88	8.49	10.72	9.27
Swift 2	13.42	8.49	14.04	8.84
COAL-b	.00	4.42	6.92	3.90
COAL-x	.00	4.42	4.08	1.47
COAL-c	10.88	6.67	8.04	8.74
COAL-f	8.09	7.38	8.71	3.90

Problem Size-Equations	12 / 18	18 / 36	18 / 54	9 / 72
-Variables	39	117	225	373
-Supply Centers (Fixed Charges)	4	8	12	16

Exhibit 4-25: Capacitated Warehouse Location Problem
-- Resource Requirements

	Solution Times (cpu sec.)			
	Size 1	Size 2	Size 3	Size 4
L.P.	.30	1.16	2.95	6.38
BBMIP	.55	10.48	62.28	791.96
F.C.L.O.	.36	1.86	4.98	13.37
DIS H 1	1.72	13.64	45.72	153.67
DIS H 2	3.84	76.83	436.84	1887.37
Swift 1	4.12	69.22	588.87	1[illegible]77.86
Swift 2	3.41	73.07	536.92	2183.51
COAL-b	1.77	13.58	52.05	136.09
COAL-x	2.68	30.72	172.81	673.61
COAL-c	.58	5.36	15.84	48.49
COAL-f	1.04	6.68	29.41	64.70
Problem	12	18	18	9
Size-Equations	18	36	54	72
-Variables	39	117	225	373
-Supply Centers (Fixed Charges)	4	8	12	16

Exhibit 4-26: Capacitated Warehouse Location Problem
- Size 4 -- Efficiency Frontier

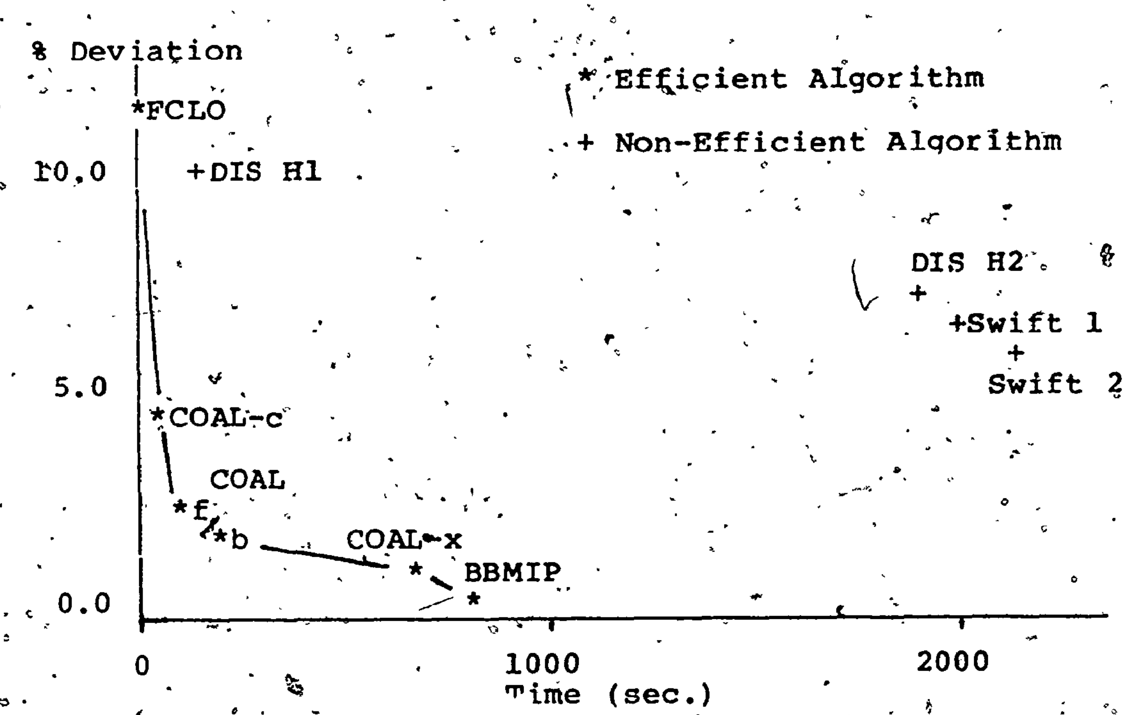


Exhibit 4-27: Capacitated Warehouse Location Problem
-- Efficient Algorithms using Averages

	Size 1	Size 2	Size 3	Size 4
L.P.	*	*	*	*
BBMIP	*	*	*	*
F.C.L.O.	-	*	*	*
DIS H 1	-	-	-	-
DIS H 2	-	-	-	-
Swift 1	-	-	-	-
Swift 2	-	-	-	-
COAL-b	-	-	*	*
COAL-x	-	-	-	*
COAL-c	-	*	*	*
COAL-f	-	-	*	*

* Efficient Algorithm (Best Quality for Solution Time)

\- Non-efficient Algorithm (Alternative algorithm obtains a better quality solution in less time)

Exhibit 4-28. As in the waste disposal problems, the solution times of the Walker and Steinberg algorithms are very dependent on the number of variables. These solution times are displayed graphically on a logarithmic scale assuming an 80% arc density (Exhibit 4-29). In particular, the graph demonstrates the impact of an exponential relationship on the solution time for BBMIP.

With the capacitated warehouse location problems and the waste disposal problems, the relationships between solution time, number of equations and number of variables for the initial fixed charge local optimum (F.C.L.O.) are very similar to the relationships for linear programming but very different from the relationships for Steinberg's Heuristic 2 and the two Walker algorithms. The adjacent extreme point search maintains its efficiency while using the revised simplex method. The marked increases in solution time of the Steinberg and Walker algorithms is due to the nature of these algorithms.

The results for (WDP) and (CWLP) are very similar. Since the basic structure of the two problems is similar, it is not surprising that they produce similar results. In order to evaluate the algorithms with different structures, problems from (FCTP) are analysed.

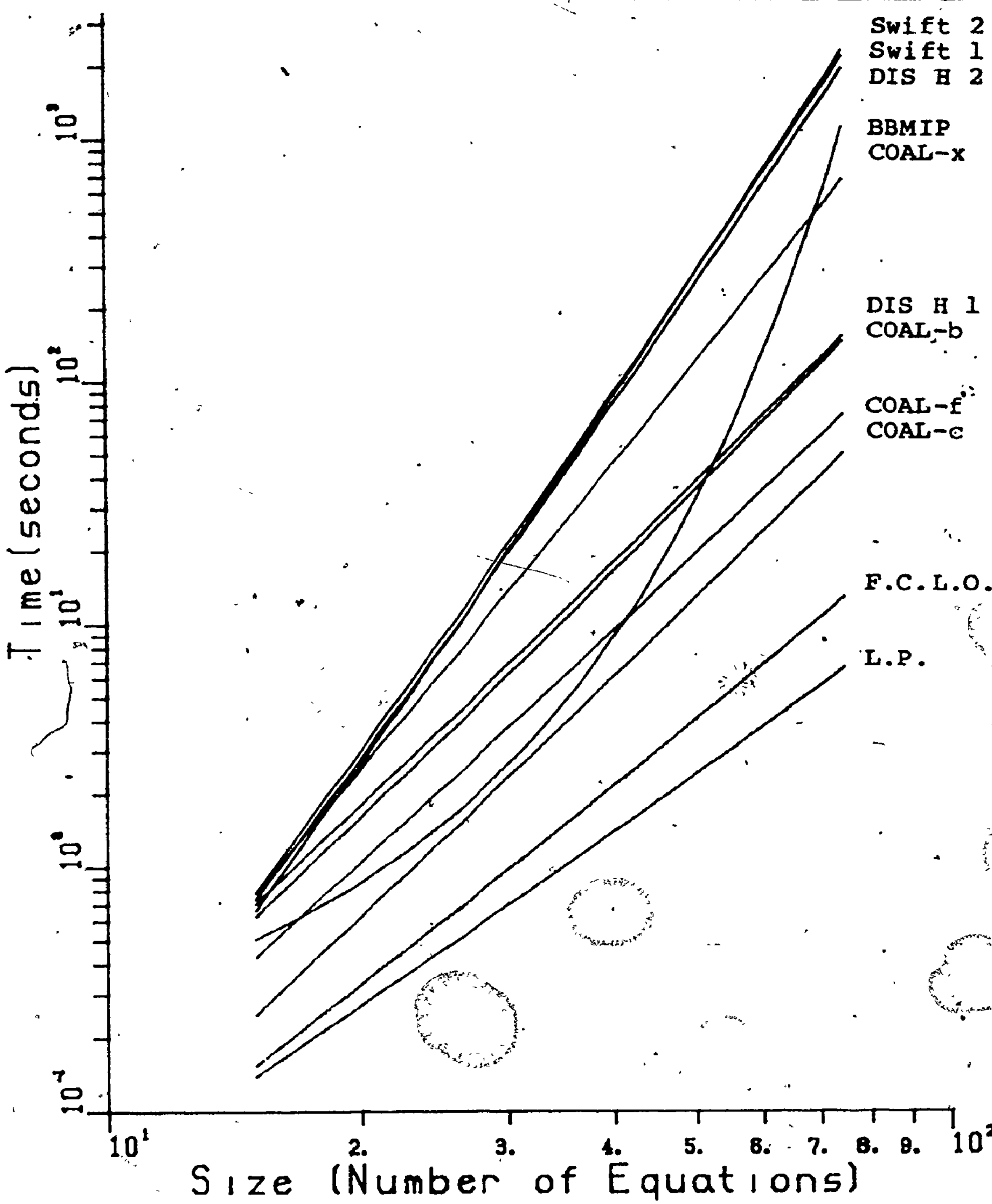
Exhibit 4-28: Capacitated Warehouse Location Problem -- Relationship Between Size and Solution Time

Algorithm	Equation	r^2 Logged	r^2 Original
Linear Programming	$t=0.000142\ n^{.989}\ m^{.576}$	.995	.976
BBMIP	$t=0.1126\ e^{.0926n}e^{.00538m}$	.938	.935
Initial Fixed Charge Local Optimum	$t=0.000841\ n^{1.16}\ m^{.616}$	.976	.858
Steinberg Heuristic 1	$t=0.000677\ n^{1.15}\ m^{1.21}$	.970	.828
Heuristic 2	$t=0.000248\ n^{2.71}\ m^{-.102}$	.990	.944
Walker Swift 1	$t=0.000166\ n^{2.51}\ m^{.285}$	.975	.784
Swift 2	$t=0.000837\ n^{2.44}\ m^{.556}$	.981	.787
COAL-b	$t=0.000486\ n^{.911}\ m^{1.66}$	.986	.957
COAL-c	$t=0.000125\ n^{.704}\ m^{2.01}$	.946	.870
COAL-f	$t=0.000301\ n^{.719}\ m^{1.87}$	.955	.797
COAL-x	$t=0.000099\ n^{1.23}\ m^{1.92}$	.980	.857

t = time(seconds) m = number of equations
n = number of variables(fixed charge + regular)

r^2 = coefficient of determination

Exhibit 4-29: Solution Times - C.W.L.P. - 80% Arc Density



4.6.3. Fixed Cost Transportation Problem

The fixed cost transportation problem represents a structure which is different from the capacitated warehouse location problem. (FCTP) has fixed charges associated with the arcs while (CWLP) has the fixed charges associated with the facilities. As in (CWLP), the fixed cost transportation problem can be solved more effectively by special purpose algorithms. However, the authors of the other approximate algorithms use (FCTP) for test purposes (taken primarily from Gray [37]) although they do not do as well as the specialized algorithms. In order to compare the impact of the two different structures, two fixed cost transportation problems are taken from Rousseau [78]. As in the previous sections, additional problems are generated with similar properties. The fixed cost transportation problems have the same number of variables as the Size 1 capacitated warehouse location problems generated previously with an arc density of 80%. The two set of problems are roughly the same size although (FCTP) has four times as many fixed charges. An attempt was made to solve each of the (FCTP) problems with BBMIP. However, this proved to be quite difficult and BBMIP was terminated before proving the final solution optimal. However, BBMIP did not find a solution that is better than the best solution obtained by at least one of the heuristics.

The quality of the solutions for (FCTP) is much better for the adjacent extreme point algorithms than for (CWLP) (Exhibit 4-30). Although COAL-x still obtains the best solutions, the other approximate methods are relatively close. The solution times (Exhibit 4-31) are similar between (FCTP) and (CWLP). COAL-x has the largest increase due to the substantial increase in the number of fixed charge variables. In spite of the improvement in performance, none of the Steinberg or Walker algorithms appear on the efficiency frontier (Exhibit 4-32 and 4-33).

The notion of an efficiency frontier can be used as a guide only when the results of different algorithms are close. Although one algorithm may appear better on an average basis, results may be different for individual problems. Swift 2 did obtain the best solution for some of the problems although, on average, it did not perform as well as the cost allocation algorithms.

The performance of the adjacent extreme point heuristics improves as they implicitly take advantage of the structure inherent in the fixed cost transportation problem. In (FCTP), each decision variable has a fixed charge associated with it. One simplex pivot typically involves exchanging the fixed charges when one variable enters the basis and removes another. With the (CWLP) problems, the fixed charges are associated with a group of decision variables. One simplex pivot typically involves shifting

Exhibit 4-30: Comparison of
Fixed Cost Transportation Problem &
Capacitated Warehouse Location Problem
-- Quality of Solution

Average Deviation from Optimum (%)

	FCTP	CWLP
L.P.	1.06	13.70
BBMIP	-na-	0.00
F.C.L.O.	.12	7.16
DIS H 1	.12	4.29
DIS H 2	.10	3.48
Swift 1	.10	3.00
Swift 2	.10	3.00
COAL-b	.10	.00
COAL-x	.02	.00
COAL-c	.35	1.81
COAL-f	.10	1.91

Maximum Deviation from Optimum (%)

	FCTP	CWLP
L.P.	2.46	26.21
BBMIP	-na-	0.00
F.C.L.O.	.59	16.75
DIS H 1	.59	15.72
DIS H 2	.59	10.88
Swift 1	.59	10.88
Swift 2	.59	10.88
COAL-b	.50	.00
COAL-x	.11	.00
COAL-c	.92	10.88
COAL-f	.50	8.09

Problem	6	6
Size-Equations	12	18
-Variables	28	29
-Fixed Charges	28	7
-Demand Centers	5	4

Exhibit 4-31: Comparison of
Fixed Cost Transportation Problem &
Capacitated Warehouse Location Problem
-- Resource Requirements

Solution Times (cpu sec.)

	FCTP	CWLP
L.P.	.21	.27
BBMIP	-na-	2.40
F.C.L.O.	.48	.34
DIS H 1	.96	1.50
DIS H 2	10.26	3.01
Swift 1	3.44	2.88
Swift 2	3.52	2.40
COAL-b	1.82	1.67
COAL-x	4.30	2.45
COAL-c	.60	.56
COAL-f	1.29	.93
Problem	6	6
Size-Equations	12	18
-Variables	28	29
-Fixed Charges	28	7
-Demand Centers	5	4

Exhibit 4-32: Fixed Cost Transportation Problem
-- Efficiency Frontier

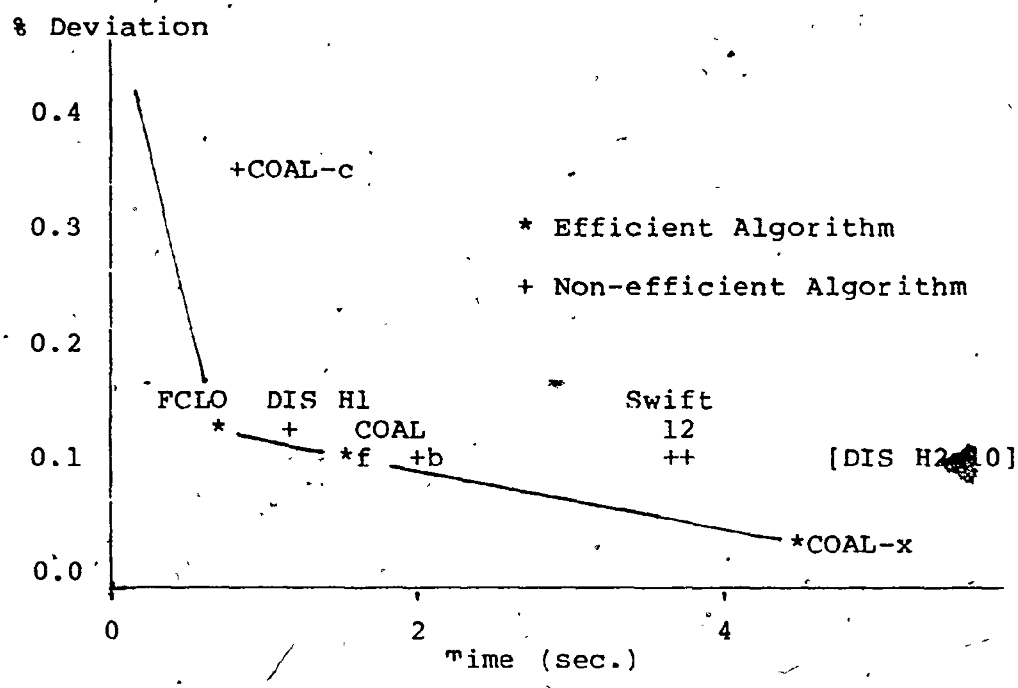


Exhibit 4-33: Comparison of
Fixed Cost Transportation Problem &
Capacitated Warehouse Location Problem
-- Efficient Algorithms using Averages

	FCTP	CWLP
L.P.	*	*
BBMIP	-na-	-na-
F.C.L.O.	*	*
DIS H 1	-	-
DIS H 2	-	-
Swift 1	-	-
Swift 2	-	-
COAL-b	-	*
COAL-x	*	-
COAL-c	-	*
COAL-f	*	-

* Efficient Algorithm (Best Quality for Solution Time)

- Non-efficient Algorithm (Alternative algorithm obtains a better quality solution in less time)

demand from one facility to another and does not always involve exchanging fixed charges. To shift from one facility to another will usually require several simplex iterations.

In spite of a structure which is favorable to the adjacent extreme point algorithms, the cost allocation algorithms outperform Steinberg's Heuristic 2 and the Walker algorithms on both dimensions of quality of solution and solution time. The cost allocation algorithms maintain a good position on the efficiency frontier obtaining "good" solutions with relatively modest requirements for cpu time.

To further confirm the results for facility location problems, particularly with large problems, an application in the location of power generating stations is analyzed.

4.6.4. Power Station Location Problem

An additional problem involving the location of nuclear power generating stations, (PSLP), is taken from Dutton et. al. [21] (Exhibit 4-34). (PSLP) is too large to be solved by BBMIP. However, the optimal solution is given by Dutton et. al. who decompose (PSLP) into transportation sub-problems which are much easier to solve. Since Swift 1 and Swift 2 are very similar, only Swift 2 is used. As observed in (WDP) and (CWLP), DIS H 2 would require lengthy computer runs to solve the Power Station Location problem and also is not used.

Exhibit 4-34: Power Station Location Problem

(PSLP) minimize $z = \sum_{i}\sum_{j} (c_{ij}X_{ij} + d_{ij}Z_{ij}) + \sum_{j} f_jY_j$

subject to:

$$\sum_{j} X_{ij} = q_i \qquad \forall\ i$$

$$\sum_{j} Z_{ij} = p_i \qquad \forall\ i$$

$$\sum_{i} X_{ij} \leq s_jY_j \qquad \forall\ j$$

$$\sum_{i} Z_{ij} \leq r_jY_j \qquad \forall\ j$$

$$X_{ij} \leq a\ Z_{ij} \qquad \forall\ i,j$$

$$X_{ij} \geq B\ Z_{ij} \qquad \forall\ i,j$$

$$X_{ij}, Z_{ij} \geq 0 \qquad \forall\ i,j$$

$$Y_j = 0,1 \qquad \forall\ j$$

where:

i = index of a demand center
j = index of a power generating center

X_{ij} = amount of annual power generated at generating station j for demand center i (kilowatt-hours).

Z_{ij} = amount of peak power generated at generating station j for demand center i (kilowatts).

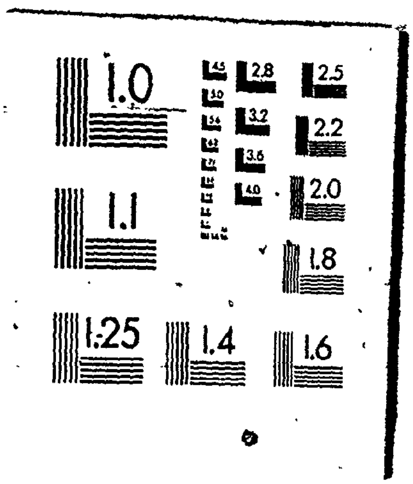
Y_j = 0-1 variable indicating if power station j is operating.

q_i = demand for annual power at center i.

p_i = demand for peak power at center i.

s_j = annual generating capacity of power station j.

3 OF/DE 3



t_j = peak generating capacity of power station j.

a = minimum amount of annual power in kilowatt-hours which must be used for each kilowatt of peak power used.

b = maximum amount of annual power in kilowatt-hours which must be used for each kilowatt of peak power used.

c_{ij} = variable cost of generating and transmitting annual power from generating station j to demand center i.

d_{ij} = variable cost of generating and transmitting peak power from generating station j to demand center i.

f_j = fixed cost of operating generating station j.

The problem is solved assuming a 4%, 5% and 6% growth in demand. Only COAL-x achieves the optimal solution in all three problems (Exhibit 4-35). Since only one problem is used, maximum deviations are not included. The other cost allocation algorithms are close while the adjacent extreme point algorithms have larger deviations. Steinberg's Heuristic 1 improves the solution found by F.C.L.O. for the 4% and 5% growth and obtains a better quality solution than Swift 2. The solution times given in Exhibit 4-36 again demonstrate the difficulty the adjacent extreme point algorithms have with a large number of variables. L.P., F.C.L.O., COAL-c, COAL-f and COAL-x are on the efficiency frontier for the 4% growth (Exhibit 4-37). For the 5% growth, COAL-c and COAL-x are on the efficiency frontier. For 6% growth, COAL-c finds the optimal solution with the smallest solution time and is on the efficiency frontier. (Exhibit 4-38).

The problem with a 6% growth rate has tight capacity constraints and is relatively easy to solve. Only one of the possible power stations is not built. Of course, the 5% and 4% growth rates have more excess capacity and are more difficult to solve. As a result, the solution time for COAL-x doubles as it improves the solutions found by COAL-b. The solution time for DIS H 1 also increases as it finds improved solutions from F.C.L.O. However, the time required for COAL-b, COAL-c, COAL-b or Swift 2 are not increased significantly by the effective increase in capacity. However, the quality of these solutions deteriorates slightly.

4.6.5. Evaluation - Facility Location

The results for the facility location problems are summarized by the efficient algorithms for each type in Exhibit 4-39. The cost allocation algorithms are efficient for all problem types in facility location. L.P., F.C.L.O. and BBMIP are also efficient for the different problem types. The Steinberg and Walker algorithms are not efficient for any problem type.

Exhibit 4-36: Power Station Location Problem
-- Resource Requirements

	Solution Times (cpu sec.)		
	4% Growth	5% Growth	6% Growth
L.P.	44.77	45.46	54.59
BBMIP	-na-	-na-	-na-
F.C.L.O.	40.37	26.91	27.25
DIS H 1	2026.28	1098.24	814.50
DIS H 2	-na-	-na-	-na-
Swift 1	-na-	-na-	-na-
Swift 2	2371.01	2758.75	2410.30
COAL-b	392.47	344.66	338.54
COAL-x	756.89	677.79	354.02
COAL-c	86.51	80.72	157.47
COAL-f	266.62	262.36	178.47

Size-Equations	263
-Fixed Charges	13
-Variables	471

Exhibit 4-36: Power Station Location Problem
-- Resource Requirements

	Solution Times (cpu sec.)		
	4% Growth	5% Growth	6% Growth
L.P.	44.77	45.46	54.59
BBMIP	-na-	-na-	-na-
F.C.L.O.	40.37	26.91	27.25
DIS H 1	2026.28	1098.24	814.50
DIS H 2	-na-	-na-	-na-
Swift 1	-na-	-na-	-na-
Swift 2	2371.01	2758.75	2410.30
COAL-b	392.47	344.66	338.54
COAL-x	756.89	677.79	354.02
COAL-c	86.51	80.72	157.47
COAL-f	266.62	262.36	178.47

Size-Equations	263
-Fixed Charges	13
-Variables	471

Exhibit 4-37: Power Station Location Problem - 4% Growth
-- Efficiency Frontier

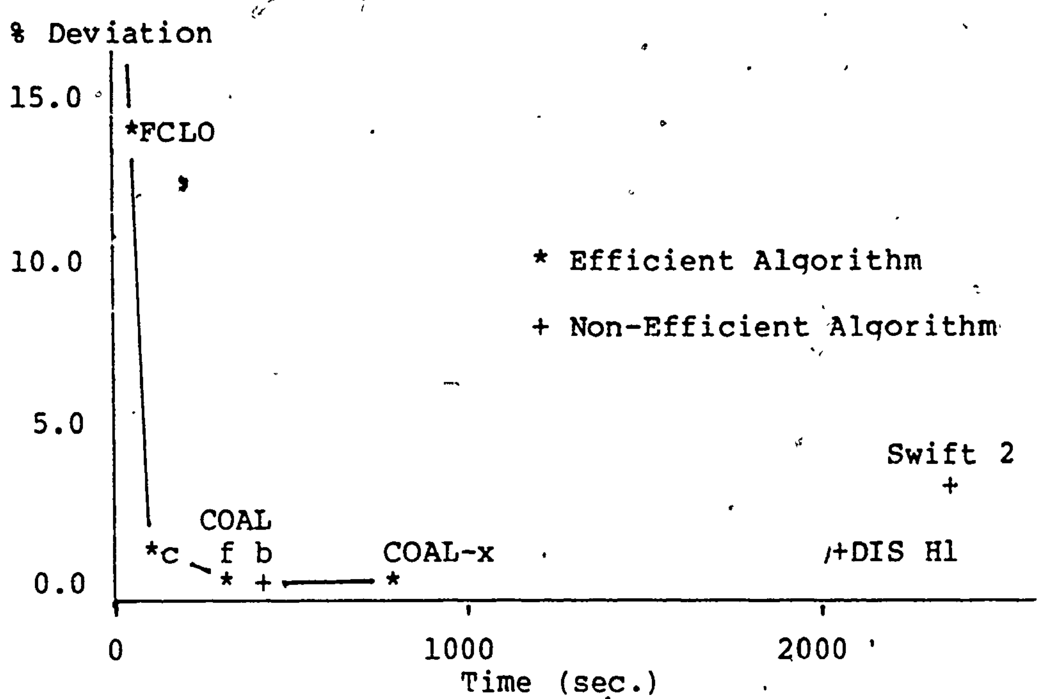


Exhibit 4-38: Power Station Location Problem
-- Efficient Algorithms

	4% Growth	5% Growth	6% Growth
L.P.	*	*	*
BBMIP	-na-	-na-	-na-
F.C.L.O.	*	-	-
DIS H 1		-	-
DIS H 2	-na-	-na-	-na-
Swift 1	-na-	-na-	-na-
Swift 2	-	-	-
COAL-b	-	-	-
COAL-x	*	*	-
COAL-c	*	*	*
COAL-f	*	-	-

* Efficient Algorithm (Best Quality for Solution Time)

\- Non-efficient Algorithm (Alternative algorithm obtains a better quality solution in less time)

The general relationship between quality of solution and resource requirement is demonstrated by an efficiency frontier for facility location problems in general (Exhibit 4-40). L.P. and F.C.L.O., although fast, provide relatively poor solutions. The different cost allocation algorithms are in the center providing greatest improvement in quality for the smallest increase in solution time. BBMIP is on the lower right providing a small increase in quality of solution at the expense of a large increase in execution time.

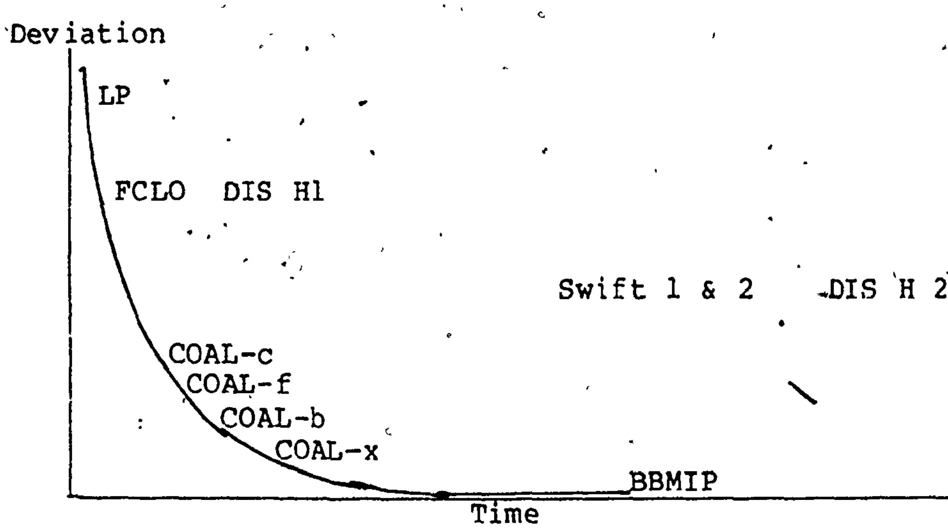
(WDP), (CWLP) and (PSLP) have similar structures with fixed charges on facilities and many continuous variables without a fixed charge. The Walker and Steinberg algorithms handle all variables as if they were fixed charge variables. Consequently, they run into difficulty as the number of continuous variables with no fixed charge is often quite large. In the analysis relating size, number of variables and solution times, the cost allocation algorithms have a higher exponent on the size component with a lower exponent on the number of continuos variables. The Walker and Steinberg algorithms have the reverse. In contrast, the Walker and Steinberg algorithms provide better solutions to the fixed cost transportation problem (FCTP), in which most of the variables have fixed charges. However, the adjacent extreme point algorithms are still not efficient.

Exhibit 4-39: Facility Location Problems -- Summary
-- Efficient Algorithms

	(WDP)	(CWLP)	(PSLP)	(FCTP)
L.P.	*	*	*	*
BBMIP	*	*	-na-	-na-
F.C.L.O.	*	*	*	*
DIS H 1	-	-	-	-
DIS H 2	-na-	-	-	-
Swift 1	-	-	-	-
Swift 2	-	-	-	-
COAL-b	-	*	-	-
COAL-x	*	*	*	*
COAL-c	*	*	*	*
COAL-f	*	*	*	*

* Efficient Algorithm

Exhibit 4-40: Efficiency Frontier - Facility Location



Execution time is a polynomial function of time for all the approximate algorithms and linear programming. Only BBMIP has an exponential growth in execution time as problem size increases. As problems become large, BBMIP will soon have difficulty with lengthy computer runs.

The new cost allocation algorithms consistently generate good solutions to facility location problems which are real, large and general fixed charge problems. The cpu requirements of the new cost allocation algorithms are significantly less than BBMIP, Steinberg's Heuristic 2 or the Walker algorithms. Both, the quality of solution and solution times are considerably better than the solution obtained by the Walker and Steinberg algorithms.

4.7. Production Planning

Production planning problems with set up costs also occur frequently in the literature. These problems are classified as the fixed charge lot size problem (Exhibit 2-12). Two problems are used as samples of production planning problems: one problem from Graves [36], and a smaller problem from Hax and Golovin [42].

4.7.1. Hierarchical Production Planning-Graves

Graves [36] uses a hierarchical production planning problem which is a fixed charge linear programming problem similar to (FCLSP) except there are no restrictions on the amount of overtime. A problem of the size used by Graves requires 240 fixed charge variables (or 240 binary variables) plus 264 continuous variables and 252 constraints. Following Graves, one problem has little seasonality in demand (Set 1), moderate seasonality (Set 2) and a high degree of seasonality (Set 3). The capacity is increased by 20% for Set 1 to generate a fourth problem (Loose Capacity). The final problem divides the fixed charges for Set 1 by 5 (Set Up * 0.2).

Swift 2 is chosen to represent the Walker algorithms. The Walker algorithms require lengthy computer runs before they terminate. Since Swift 1 produces results similar to Swift 2, it is not used. These problems are too time consuming to be reasonably solved by Steinberg's slow Heuristic 2 or COAL-x and much too large to be attempted by BBMIP. While COAL-b runs are very lengthy as well, the additional time beyond COAL-f and COAL-c is minimal.

Since the optimal solution is not known, the deviation from the best solution obtained by the algorithms tested is used for the measurement of quality (Exhibit 4-41). Swift 2 obtains the best solution for 3 out of 5 problems while COAL-f (and COAL-b) obtains the best solution for the other two problems.

Of the algorithms tested, Swift 2 requires the longest execution times (Exhibit 4-42). The cost allocation algorithms require less time while F.C.L.O. and L.P. require even less. The cost allocation algorithms exploit the smaller fixed charges for the "Set Up * 0.2" to significantly reduce their execution times.

The efficiency frontier for Set 2 demonstrates the extra time required by Swift 2 to obtain a better solution (Exhibit 4-43). The efficient algorithms for the different problems given in Exhibit 4-44 indicate that all the algorithms tested, from time to time, are efficient.[1]

The increasing seasonality generally makes the problem easier to solve as it in some sense makes it smaller. The out of season demand for these problems is set to zero which in effect reduces the size of the problem and therefore the time required to solve for the cost allocation algorithms.

The "Loose Capacity" relaxes the constraint by increasing the capacity. Since there is already a fair degree of excess capacity particularly with no restriction on overtime, the problem becomes easier to solve.

---

1. By comparison, Graves requires an average of 236 cpu seconds for the five problems on a Prime 400 to come within 5% of a lower bound. Graves states that a PRIME 400 is 3 to 8 times slower than a 370/168. A Cyber 173 would also be slower than a 370/168. Although comparisons are difficult, it would appear that the Graves algorithm would be faster for this particular type of problem and below the efficiency frontier for the algorithms tested. Of course, the fixed charge linear programming formulation is more flexible than the Graves algorithm would allow.

Exhibit 4-41: Production Planning Problem-Graves
-- Quality of Solution

Deviation from Best (%)

	Set 1	Set 2	Set 3	Loose Capacity (Set 1)	Set Up * 0.2 (Set 1)
L.P.	27.37	25.80	22.23	28.93	2.68
BBMIP	-na-	-na-	-na-	-na-	-na-
F.C.L.O.	1.89	8.30	1.72	.40	.57
DIS H 1	1.75	3.58	1.71	.40	.57
DIS H 2	-na-	-na-	-na-	-na-	-na-
Swift 1	-na-	-na-	-na-	-na-	-na-
Swift 2	.21	.00	.00	.00	.37
COAL-b	.00	.42	.20	.10	.00
COAL-x	-na-	-na-	-na-	-na-	-na-
COAL-c	3.98	2.86	.37	.40	.17
COAL-f	.00	.42	.57	.10	.00

Size-Equations 252
-Fixed Charges 240
-Variables 504

Exhibit 4-42: Production Planning Problem-Graves -- Resource Requirements

Solution Times (sec.)

	Set 1	Set 2	Set 3	Loose Capacity (Set 1)	Set Up * 0.2 (Set 1)
L.P.	25.0	29.0	35.1	23.2	24.1
BBMIP	-na-	-na-	-na-	-na-	-na-
E.C.L.O.	617.0	294.2	209.4	716.0	71.5
DIS H 1	1503.8	1858.7	1590.2	1453.3	1079.0
DIS H 2	-na-	-na-	-na-	-na-	-na-
Swift 1	-na-	-na-	-na-	-na-	-na-
Swift 2	9136.3	13140.7	5590.0	4841.8	4831.5
COAL-b	7381.8	4809.8	3251.1	4640.3	604.0
COAL-x	-na-	-na-	-na-	-na-	-na-
COAL-c	3240.3	2273.6	1550.5	2101.5	247.6
COAL-f	3755.4	2356.7	1635.2	2644.4	330.3

Size-Equations 252
-Fixed Charges 240
-Variables 504

Exhibit 4-43: Production Planning Problem-Graves-Set 2
-- Efficiency Frontier

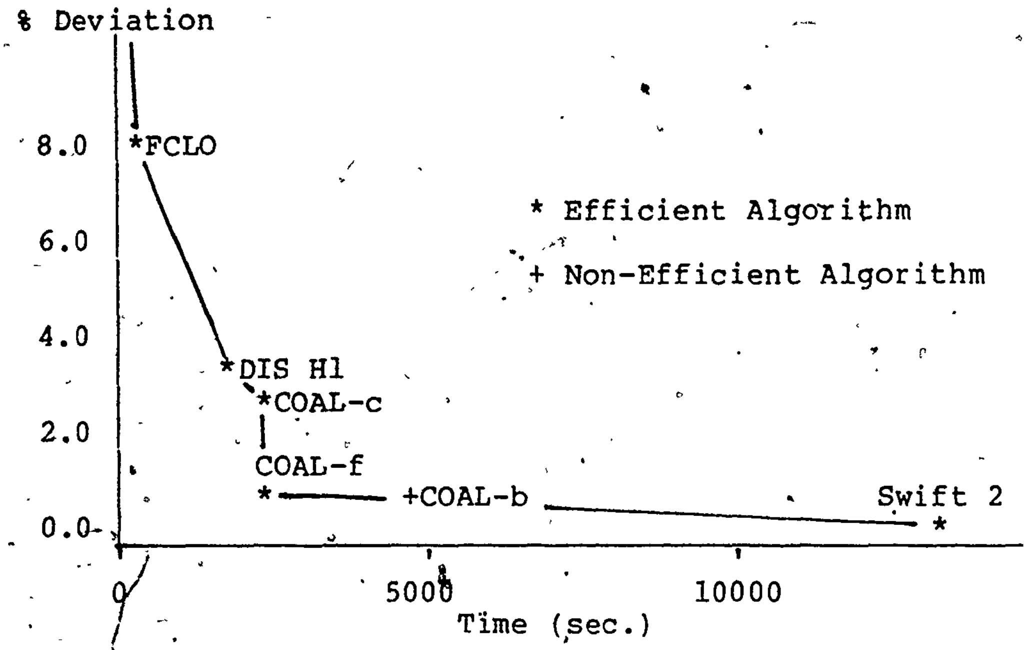


Exhibit 4-44: Production Planning Problem-Graves -- Efficient Algorithms

	Set 1	*Set 2	Set 3	Loose Capacity (Set 1)	Set Up * 0.2 (Set 1)
L.P.	*	*	*	*	*
BBMIP	-na-	-na-	-na-	-na-	-na-
F.C.L.O.	*	*	*	*	*
DIS H 1	*	*	-	-	-
DIS H 2	-na-	-na-	-na-	-na-	-na-
Swift 1	-na-	-na-	-na-	-na-	-na-
Swift 2	-	*	*	*	-
COAL-b	-	-	*	-	-
COAL-x	-na-	-na-	-na-	-na-	-na-
COAL-c	-	*	*	-	*
COAL-f	*	*	-	*	*

* Efficient Algorithm (Best Quality for Solution Time)

\- Non-efficient Algorithm (Alternative algorithm obtains a better quality solution in less time)

These problems are dominated by the fixed charges. "Set Up * 0.2" divides the fixed charges by five and makes the problem considerably easier for the cost allocation algorithms to solve.

All the algorithms tested appear on the efficiency frontier for some of the problems. Swift 2 is on the far right taking much longer than the others with a slight improvement in quality over COAL-f. Steinberg's Heuristic 1 manages to improve the solution found by the initial Fixed Charge Local Optimum but is still on the upper left of the efficiency frontier. The cost allocation algorithms tested are in the middle showing good improvement in quality with small improvements in the time required.

4.7.2. Hierarchical Production Planning-Hax and Golovin

Due to the difficulty in solving the problem from Graves and thus making comparisons, a similar problem with the same structure as (FCLSP) including a limit on the amount of over time but only 65 binary variables is selected from Hax and Golovin [42].

The basic problem (Base Case) has a seasonality similar to Set 2 from Graves with an intermediate amount of seasonality. The fixed charges are similar in size to the "Set Up * 0.2" case from Graves. The size of the fixed charges is varied for two problems. "Set Up Case I" has fixed charges in between the Base Case and the problem from

Graves. The fixed charges in "Set Up Case II" are comparable in size to the Graves problem. The impact of capacity constraints is tested by two additional sensitivity analyses. "Tight Capacitv" represents a reduction in the production capacity available while "Loose Capacity" represents an increase in the production capacity. Hax and Golovin state that the parameters of the Base Case represent a production planning problem in tire manufacturing. These problems can be solved within reasonable time limits by all algorithms except BBMIP.

Since the optimal solutions for these problems are not known, the quality of a solution is measured with respect to the best solution found (Exhibit 4-45). COAL-x obtains the best solution most frequently followed by Steinberg's Heuristic 2, Swift 1 and COAL-b. However, all the approximate methods achieve relatively low deviations.

COAL-x and Steinberg's Heuristic 2 have long solution times for some problems (Exhibit 4-46). COAL-x requires lengthy runs in order to improve some of the solutions obtained by both COAL-f and COAL-c. The two Walker algorithms are generally faster than COAL-x or DIS H 2. COAL-c, COAL-f and COAL-b take less time than the Walker algorithms. The increase in set up costs slows down the cost allocation algorithms for "Set Up I" and "Set Up II".

The relative positioning of the different algorithms on the efficiency frontier is demonstrated in Exhibit 4-47 for the Base Case. As in Graves' problems, all of the algorithms are efficient for at least some of the problems (Exhibit 4-48).

An attempt was made to solve the "Base Case" and the "Set Up Case II" of the Hax and Golovin problems with BBMIP. The program ran for approximately 4 hours of CPU time before the program was stopped. In the "Base Case", the best solution is the same as given in Exhibit 4-45. With the "Set Up Case II", the best solution found by BBMIP is inferior to the solution in Exhibit 4-45. Although BBMIP was allowed to run for approximately 4 hours of CPU time for each problem, the elapsed time was 36 hours on a quiet Christmas weekend. The extra time is required to save and restore various tables on the disk. Since none of the approximate methods require disk storage, an equivalent amount of cpu time could be obtained during such low use periods in approximately 4.5 hours elapsed time. The large increase in both cpu time and elapsed time demonstrates the problems associated with branch and bound mixed integer programming as problem size increases.

4.7.3. Evaluation -- Production Planning

An assessment of the efficient algorithms for the two sets of problems with the sensitivity analyses are summarized in Exhibit 4-49. Essentially, all the algorithms tested are efficient. Only Swift 2, which is dominated by

Exhibit 4-45: Production Planning Problem-Hax & Golovin -- Quality of Solution

	Deviation from Best (%)				
	Base Case	Set Up Case I	Set Up Case II	Tight Capacity	Loose Capacity
L.P.	.43	8.84	20.65	.36	1.05
BBMIP	-na-	-na-	-na-	-na-	-na-
F.C.L.O.	.21	4.08	3.52	.15	.18
DIS H 1	.21	4.08	3.52	.15	.00
DIS H 2	.00	1.23	1.74	.00	.00
Swift 1	.00	2.18	.00	.07	.00
Swift 2	.05	2.18	1.74	.07	.00
COAL-b	.00	1.28	.71	.00	.16
COAL-x	.00	.00	.16	.00	.16
COAL-c	.21	1.28	.71	.01	.23
COAL-f	.11	1.82	1.51	.00	.23

Size-Equations 91
-Fixed Charges 65
-Variables 169

Exhibit 4-46: Production Planning Problem-Hax & Golovin -- Resource Requirements

Solution Times (cpu sec.)

	Base Case	Set Up Case I	Set Up Case II	Tight Capacity	Loose Capacity
L.P.	5.35	5.32	5.36	6.32	3.91
BBMIP	-na-	-na-	-na-	-na-	-na-
F.C.L.O.	5.98	10.43	16.95	7.05	7.92
DIS H 1	38.29	46.59	59.54	45.20	49.87
DIS H 2	636.48	500.34	707.28	2871.12	271.48
Swift 1	163.69	396.78	466.57	255.11	159.46
Swift 2	255.75	190.12	326.74	148.61	175.51
COAL-b	43.79	57.38	196.02	44.99	48.38
COAL-x	130.49	608.76	1111.99	170.77	114.89
COAL-c	13.33	23.29	100.03	17.71	20.11
COAL-f	23.25	29.15	98.48	21.39	18.58

Size-Equations	91
-Fixed Charges	65
-Variables	169

Exhibit 4-47: Production Planning Problem-Hax & Golovin
- Base Case -- Efficiency Frontier

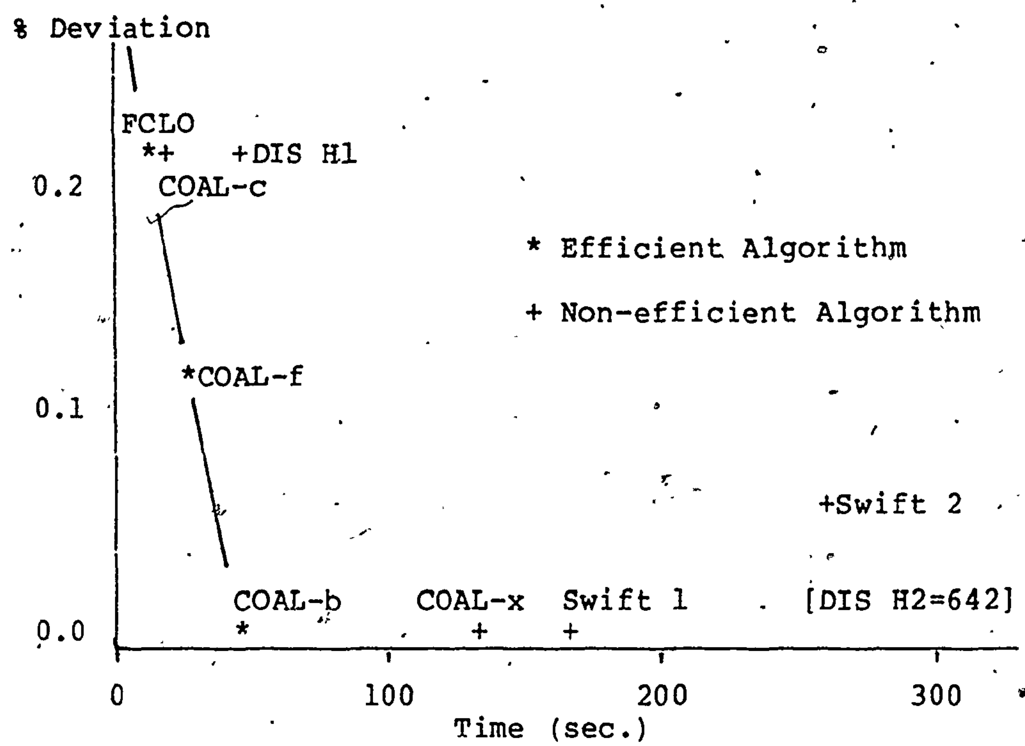


Exhibit 4-48: Production Planning Problem-Hax & Golovin
-- Efficient Algorithms

	Base Case	Set Up Case I	Set Up Case II	Tight Capacity	Loose Capacity
L.P.	*	*	*	*	*
BBMIP	-na-	-na-	-na-	-na-	-na-
F.C.L.O.	*	*	*	*	*
DIS H 1	-	-	-	-	*
DIS H 2	-	*	-	-	-
Swift 1	-	-	*	-	-
Swift 2	-	-	-	-	-
COAL-b	*	-	-	-	*
COAL-x	-	*	-	-	-
COAL-c	-	*	*	*	-
COAL-f	*	-	*	*	-

* Efficient Algorithm (Best Quality for Solution Time)

- Non-efficient Algorithm (Alternative algorithm obtains a better quality solution in less time)

its close relative, Swift 1, is not efficient for one set. Since BBMIP is not tested, its efficiency can not be evaluated. While each algorithm is not efficient for every problem, each algorithm is efficient for some of the variations. Also, the algorithms are close to the efficiency frontier for each problem. Therefore, each algorithm is assessed as being efficient. The relative positions of each algorithm on the efficiency frontier is shown in Exhibit 4-50.

Again, L.P. and F.C.L.O. are on the upper left with fast solution times and poor quality. DIS H 1 obtains improved solutions with an increase in solution time. The cost allocation algorithms, COAL-c, COAL-f and COAL-b, are on a good position on the efficiency frontier obtaining better solutions with relatively modest increases in execution times. The other adjacent extreme point algorithms, Swift 1 and 2 and DIS H 2, sometimes obtain slightly better solutions but require lengthy computer runs. COAL-x, which also occasionally obtains the best solutions, requires more time than the Walker algorithms but less than DIS H 2. Steinberg's Heuristic 2 has the longest execution times but has little if any improvement in quality.

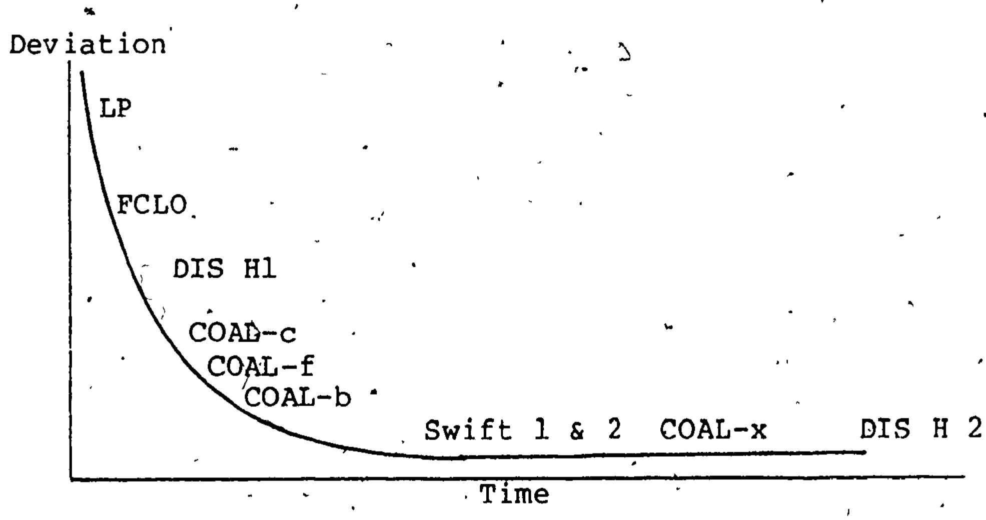
When the size of the fixed charges is decreased, the solution time of the cost allocation algorithms improves. However, the execution time of the adjacent extreme point algorithms is independent of the size of the fixed charges.

Exhibit 4-49: Production Planning Problems - Summary -- Efficient Algorithms

	Graves	Hax & Golovin
L.P.	*	*
BBMIP	-na-	-na-
F.C.L.O.	*	*
DIS H 1	*	*
DIS H 2	-na-	*
Swift 1	-na-	*
Swift 2	*	-
COAL-b	*	*
COAL-x	-na-	*
COAL-c	*	*
COAL-f	*	*

* Efficient Algorithm (Best Quality for Solution Time)

Exhibit 4-50: Efficiency Frontier - Production Planning



In (FCLSP), a simplex pivot is often quite significant. A simplex pivot can result in the production from one month being shifted to another month thus avoiding the setup costs. Consequently, the adjacent extreme point algorithms generate good solutions. Never the less, the cost allocation algorithms, particularly COAL-c, COAL-f and COAL-b, are very efficient in producing good solutions to the fixed charge lot size problems.

4.8. Manpower Planning

Two problems from manpower planning are selected to further investigate the performance of the different algorithms in solving problems with different structures. One example involves the integration of manpower planning into a production-planning problem by allowing the size of the work force to be a variable. This approach is used by Hax [41] with his variable work force problem and by Mangiameli and Krajewski [60] in their study of the effect of different work force strategies. These problems have a single variable representing the size of the work force. A second multi-period manpower planning problem in sales force management is used which has a number of variables representing different levels of experience and training thus adding additional complexity to the problem.

4.8.1. Variable Work Force Problem

The variable work force problem, (VWF), is an extension of production planning where the size of the work force is a decision variable as well as the decisions involving production level. Controlling the size of the work force involves equations for a manpower balance as well as decision variables for the number hired and fired in each time period. The formulation for (VWF) is given in Exhibit 4-51.

By definition, (VWF) and (FCLSP) are related. Although similar, (VWF) is more complicated than (FCLSP) involving the interaction of two sub-problems for the production level decisions and for the manpower level decisions. Since there is such a close relationship between (VWF) and (FCLSP), more insight into the performance of the various algorithms can be achieved by using the Hax and Golovin [42] problem as the basis for the variable work force problem. However, the structure for hiring, training and firing is taken from Mangiameli and Krajewski [60]. The hiring cost represents a two week training period while the firing cost represents two week severance pay. In addition, a sensitivity analysis is performed on a fixed charge assigned to training which is set at 0, $1,000 (two weeks pay for one instructor) and $5,000 (a large fixed charge).

The quality of solutions for the Base Case for the (VWF) problem (Exhibit 4-52) is very similar to the results

Exhibit 4-51: Variable Work Force Problem

(VWF) minimize $z = \sum_t \left(c_t R_t + d_t O_t + a_t H_t + b_t F_t + g_t Z_t + \sum_j (u_{jt} I_{jt} + s_{jt} Y_{jt})\right)$

subject to:

$$P_{jt} + I_{j,t-1} - I_{jt} = v_{jt} \qquad \forall\ j,t$$

$$\sum_j (w_j P_{jt}) - O_t \le R_t \qquad \forall\ t$$

$$R_t - H_t + F_t = R_{t-1} \qquad \forall\ t$$

$$O_t \le q_t R_t \qquad \forall\ t$$

$$H_t - n_t Z_t \le 0 \qquad \forall\ j,t$$

$$P_{jt} - m_{jt} Y_{jt} \le 0 \qquad \forall\ j,t$$

$$O_t, P_{jt}, I_{jt}, \ge 0 \qquad \forall\ j,t$$

$$Z_t, Y_{jt} = 0,1 \qquad \forall\ j,t$$

where:

j = product group
t = time periods

R_t = regular workforce in period t

O_t = overtime worked in period t

H_t = additional hours hired in period t

F_t = Hours laid off in period t

Z_t = binary variable for training in period t

I_{jt} = inventory, group j in t

P_{jt} = production, group j in t

Y_{jt} = Binary variable for production of group j in period t

c_t = regular payroll in period t

d_t = overtime payroll in period t

a_t = hiring cost/hour in period t

b_t = firing cost/hour in period t

g_t = fixed training cost in period t

u_{jt} = holding cost, group j period t

s_{jt} = set up cost, group j period t

q_t = over time limit, period t

w_j = production time required/unit, group j

v_{jt} = demand, group j in t

m_{jt} = maximum production, group j in t

n_t = maximum training, period t

---

obtained with the original Base Case from Hax and Golovin. Swift 1 and 2 obtain the best solution for all three problems. Steinberg's Heuristic 2 obtains the best solution in two problems and a large deviation in the other. COAL-f obtains the best solution in one and low deviations in the other two problems. COAL-b and COAL-x obtain the same solutions as COAL-f. COAL-c obtains one best solution, one low deviation and one high deviation.

The Swift algorithms and DIS H 2 take much longer with (VWF) than with (FCLSP) due to the increase in the number of variables (Exhibit 4-53). COAL-x takes considerably longer in one case. The long execution time for COAL-x results from the effort required to improve the solution obtained by COAL-c. COAL-x takes considerably longer when the solution from COAL-c (or COAL-f) is poor.

The efficiency frontier for the problem with $1,000 fixed training cost demonstrates the extra time required by the Walker algorithms to obtain an improved solution (Exhibit 4-54). As in the production planning problems, a number of algorithms are efficient for different problems (Exhibit 4-55).

The variable work force problem involves the interaction of two sub-problems: one for the production level decisions, and one for the work force level decisions. The fixed charges on the production set up are relatively small when compared with the fixed charges on the training costs, particularly with the $5,000 fixed charge. Consequently, the base run variable work force problem with fixed training costs is dominated by the work force level problem.

In order to make the production level decisions "more important", a second problem using "Set Up Case I" is used. The interaction of the two sub-problems with "Set Up Case I" makes the problem more difficult for the adjacent extreme point heuristics as shown by the poor quality of solution they obtain (Exhibit 4-56). The Walker algorithms have high deviations in all three problems. Steinberg's Heuristic 2 manages a low deviation in one problem. COAL-x obtains the best solution for all three problems. COAL-b obtains the best solution in two with a low deviation in the third. COAL-f obtains the best solution in one problem and low

Exhibit 4-52: Variable Work Force Problem - Base Case -- Quality of Solution

Deviation from Best (%)

	Fixed Training Costs ($) 0	1,000	5,000
L.P.	.110	.994	7.560
BBMIP	-na-	-na-	-na-
F.C.L.O.	.056	.083	3.027
DIS H 1	.056	.083	3.027
DIS H 2	.000	.000	1.495
Swift 1	.000	.000	.000
Swift 2	.000	.000	.000
COAL-b	.000	.027	.028
COAL-x	.000	.027	.028
COAL-c	.000	.027	2.406
COAL-f	.000	.027	.028
Size-Equations	104	104	104
-Variables	195	195	195
-Fixed Charges	65	78	78

Exhibit 4-53: Variable Work Force Problem - Base Case -- Resource Requirements

Solution Times (cpu sec.)

	Fixed Training Costs ($)		
	0	1,000	5,000
L.P.	10.07	9.74	10.00
BBMIP	-na-	-na-	-na-
F.C.L.O.	5.10	7.16	7.41
DIS H 1	62.22	64.31	64.09
DIS H 2	911.03	2157.44	749.23
Swift 1	402.09	331.61	323.46
Swift 2	291.28	352.09	423.63
COAL-b	33.05	42.95	88.75
COAL-x	117.82	187.32	1194.86
COAL-c	15.57	24.97	33.77
COAL-f	16.56	17.67	39.60
Size-Equations	104	104	104
-Variables	195	195	195
-Fixed Charges	65	78	78

Exhibit 4-54: Variable Work Force Problem - Base Case
- Fixed Training Costs $1,000
-- Efficiency Frontier

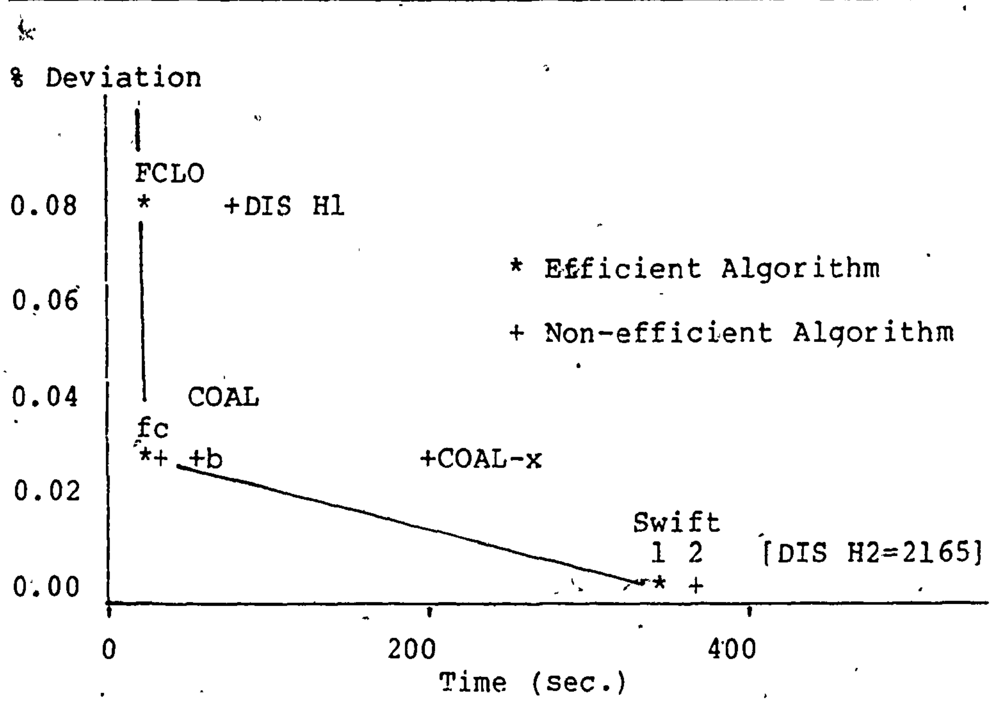


Exhibit 4-55: Variable Work Force Problem - Base Case -- Efficient Algorithms

	Fixed Training Costs ($)		
	0	1,000	5,000
L.P.	*	*	*
BBMIP	-na-	-na-	-na-
F.C.L.O.	*	*	*
DIS H 1	-	-	-
DIS H 2	-	-	-
Swift 1	-	*	*
Swift 2	-	-	-
COAL-b	-	-	-
COAL-x	-	-	-
COAL-c	*	-	*
COAL-f	-	*	*

* Efficient Algorithm (Best Quality for Solution Time)

\- Non-efficient Algorithm (Alternative algorithm obtains a better quality solution in less time)

deviations in the other two. COAL-c obtains one best, one low and one high deviation. This problem demonstrates the inability of the adjacent extreme point algorithms to handle situations which have relatively complex relationships between the fixed charge variables.

The solutions times with "Set Up Case I" are similar to the (VWF) problem using the Base Case (Exhibit 4-57). The two Walker algorithms and Steinberg's Heuristic 2 require a large amount of computer time. COAL-x again requires a lengthy computer run to improve the one poor solution from COAL-c.

Consequently, the cost allocation algorithms dominate the Walker and Steinberg algorithms on the efficiency frontier (Exhibit 4-58). Swift 1, 2 and DIS H 2 do not appear as efficient algorithms for any of the three problems (Exhibit 4-59).

The second variable work force problem illustrates the difficulty the adjacent extreme point algorithms have in coping with complex relationships between fixed charges. The cost allocation algorithms obtain good solutions in both problems with modest cpu time requirements. The difficulty the adjacent extreme point heuristics have evaluating complex models is further demonstrated in the next section suing a multi-period multi-level manpower planning in a sales force problem with fixed charges on groups of variables.

Exhibit 4-56: Variable Work Force Problem - Set Up Case I
-- Quality of Solution

Deviation from Best (%)

	Fixed Training Costs ($)		
	0	1,000	5,000
L.P.	3.394	4.277	10.091
BBMIP	-na-	-na-	-na-
F.C.L.O.	3.174	3.377	6.193
DIS H 1	.417	1.888	6.193
DIS H 2	.966	.160	1.774
Swift 1	1.629	1.763	3.383
Swift 2	1.628	1.763	3.383
COAL-b	.000	.000	.055
COAL-x	.000	.000	.000
COAL-c	.000	.252	2.139
COAL-f	.097	.000	.055
Size-Equations	104	104	104
-Variables	195	195	195
-Fixed Charges	65	78	78

Exhibit 4-57: Variable Work Force Problem - Set Up Case I -- Resource Requirements

Solution Times (cpu sec.)

	Fixed Training Costs ($)		
	0	1,000	5,000
L.P.	9.99	9.73	9.73
BBMIP	-na-	-na-	-na-
F.C.L.O.	7.27	8.20	8.20
DIS H 1	186.61	87.36	60.95
DIS H 2	718.20	1192.70	900.27
Swift 1	402.09	331.61	323.46
Swift 2	283.57	316.84	306.64
COAL-b	131.04	144.92	125.67
COAL-x	341.37	385.77	1579.07
COAL-c	70.40	82.75	57.93
COAL-f	62.57	53.56	47.73
Size-Equations	104	104	104
-Variables	195	195	195
-Fixed Charges	65	78	78

Exhibit 4-58: Variable Work Force Problem - Set Up Case I
- Fixed Training Costs $1,000
-- Efficiency Frontier

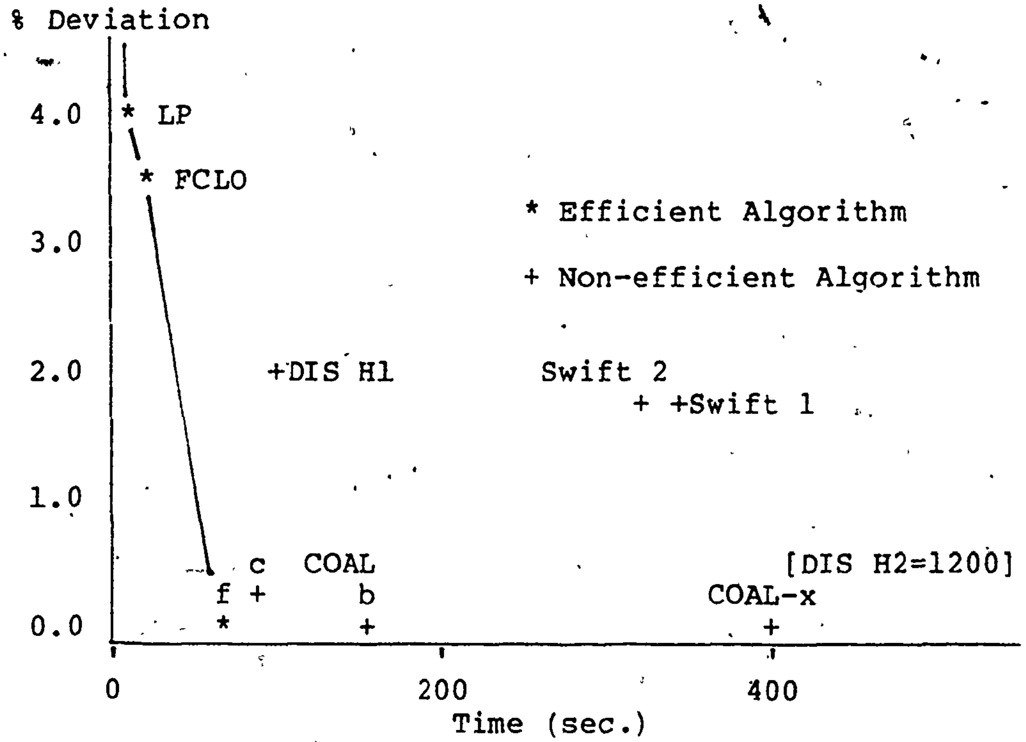


Exhibit 4-59: Variable Work Force Problem - Set Up Case I -- Efficient Algorithms

	Fixed Training Costs ($)		
	0	1,000	5,000
L.P.	*	*	*
BBMIP	-na-	-na-	-na-
F.C.L.O.	*	*	*
DIS H 1	-	-	-
DIS H 2	-	-	-
Swift 1	-	-	-
Swift 2	-	-	-
COAL-b	-	-	-
COAL-x	-	-	*
COAL-c	*	-	-
COAL-f	*	*	*

* Efficient Algorithm (Best Quality for Solution Time)

\- Non-efficient Algorithm (Alternative algorithm obtains a better quality solution in less time)

4.8.2. Strategic Manpower Planning

Haehling von Lanzenauer et. al. [38] have used a fixed charge formulation in a problem in the development of hiring and training policies for a sales force in a strategic manpower planning problem, (SMP) (Exhibit 4-60). The fixed charges are associated with hiring and training of employees. However, the employees eligible for certain training may have different experience levels. Thus, the fixed charges for this problem apply to groups of decision variables rather than a single variable as in (FCLSP).

A sensitivity analysis on the impact of the size of the fixed charges is included with (SMP). Results are presented with the fixed charge multiplied by 2, 3 and 4. The optimal solution is obtained for the Base Run and "Fixed Charges * 4" by all the cost allocation algorithms and Swift 1 (Exhibit 4-61). Swift 2 obtains the optimal solution for the Base Run. COAL-c obtains the best solution by an approximate method for Fixed Charge * 2 and Fixed Charges * 3. algorithms. As the fixed charges increase the problem becomes more difficult. However, further increases make the problem easier to solve.

The performance of the Walker and Steinberg algorithms with respect to solution time is relatively poor (Exhibit 4-62). The adjacent extreme point algorithms are hampered by the relatively large number of continuous variables which are handled as if they were fixed charge variables. In

Exhibit 4-60: Strategic Manpower Planning

$$\text{(SMP)} \quad \text{maximize} \quad z = \sum_i\sum_j\sum_t (r_{ij}X_{ijt} - c_{ij}T_{ijt}) + \sum_i\sum_t f_iY_{it}$$

subject to:

$$X_{i,j+1,t+1} = a_{ij}\,(X_{ijt} - T_{ijt} + T_{i-1,jt}) \qquad \forall\ ijt$$

$$\sum_i\sum_j X_{ijt} \leq m_t \qquad \forall\ t$$

$$\sum_j T_{ijt} - u\,Y_{it} \leq 0 \qquad \forall\ it$$

$$\sum_{j\varepsilon S(j)}\sum_i T_{ijt} \leq \sum_{i\varepsilon Q(i)}\sum_j s_{ij}X_{ijt} \qquad \forall\ t$$

$$X_{ijt}, T_{ijt} \geq 0 \qquad \forall\ ijt$$

$$Y_{it} = 0,1 \qquad \forall\ it$$

where:

i = index of training classification
j = index of experience classification
t = time period

X_{ijt} = number of salesmen of experience j training i in period t

T_{ijt} = number of salesmen of experience j training i in period t sent to a training course.

Y_{it} = A 0-1 variable indicating a training course for class i is given.

r_{ij} = revenue produced by a salesman in class j, experience i.

c_{ij} = variable cost of training a salesman in training class j and experience i.

f_i = fixed cost of training salesmen in training class i.

a_{ij} = attrition rate of salesmen in in training class j and experience i.

m_t = maximum number of salesmen in period t.

u = upper limit on salesmen in a course.

$j \varepsilon S(j)$ = set of experience classes requiring supervision.

$i \varepsilon Q(i)$ = set of training classes capable of supervision.

s_{ij} = number of salesmen that can be supervised by a supervisor.

---

contrast, the cost allocation algorithms are very fast. COAL-x is faster than DIS H 2, Swift 1 or 2.

Consequently, the Steinberg and Walker algorithms do not appear on the efficiency frontier for any problem. The cost allocation algorithms dominate the slower Steinberg algorithm and the two Walker algorithms. Since COAL-c and COAL-f perform will with respect to the deviation from the optimum, COAL-b and COAL-x do not improve on them and do not appear on the efficiency frontier (Exhibit 4-63 and 4-64).

The adjacent extreme point algorithms do not get good solutions when fixed charges are associated with groups of variables. In (FCLSP) and (VWF), the fixed charges are associated with a single variable. A simplex pivot is more significant in the context of fixed charges for (FCLSP) than for (SMP). The decision to shift production in (FCLSP) from one period to another and save a set-up is accomplished by

Exhibit 4-61: Strategic Manpower Planning
-- Quality of Solution

Deviation from Optimum (%)

	Base Run	Fixed Charges * 2	Fixed Charges * 3	Fixed Charges * 4
L.P.	.56	2.38	4.62	7.32
BBMIP	.00	.00	.00	.00
F.C.L.O.	.02	.57	1.46	2.77
DIS H 1	.02	.24	.69	1.54
DIS H 2	.02	.24	.47	.87
Swift 1	.00	.08	.06	.00
Swift 2	.00	.24	.55	.78
COAL-b	.00	.06	.03	.00
COAL-x	.00	.06	.03	.00
COAL-c	.00	.06	.03	.00
COAL-f	.02	.08	.25	.00

Size-Equations	120
-Fixed Charges	30
-Variables	240

Exhibit 4-62: Strategic Manpower Planning
--- Resource Requirements

Solution Times (cpu sec.)

	Base Run	Fixed Charges * 2	Fixed Charges * 3	Fixed Charges * 4
L.P.	8.14	8.15	8.45	8.20
BBMIP	2755.00	2270.03	2056.70	3278.60
F.C.L.O.	14.00	17.18	17.83	17.18
DIS H 1	151.36	231.87	235.55	210.66
DIS H 2	500.40	395.17	1878.26	972.33
Swift 1	615.70	614.99	873.75	867.07
Swift 2	447.92	294.04	498.06	602.92
COAL-b	33.27	51.50	55.99	69.34
COAL-x	98.03	162.80	270.60	133.22
COAL-c	17.35	25.73	34.75	40.70
COAL-f	16.30	22.56	19.69	29.27

Size-Equations	120
-Fixed Charges	30
-Variables	240

Exhibit 4-63: Strategic Manpower Planning
- Fixed Charges * 3 -- Efficiency Frontier

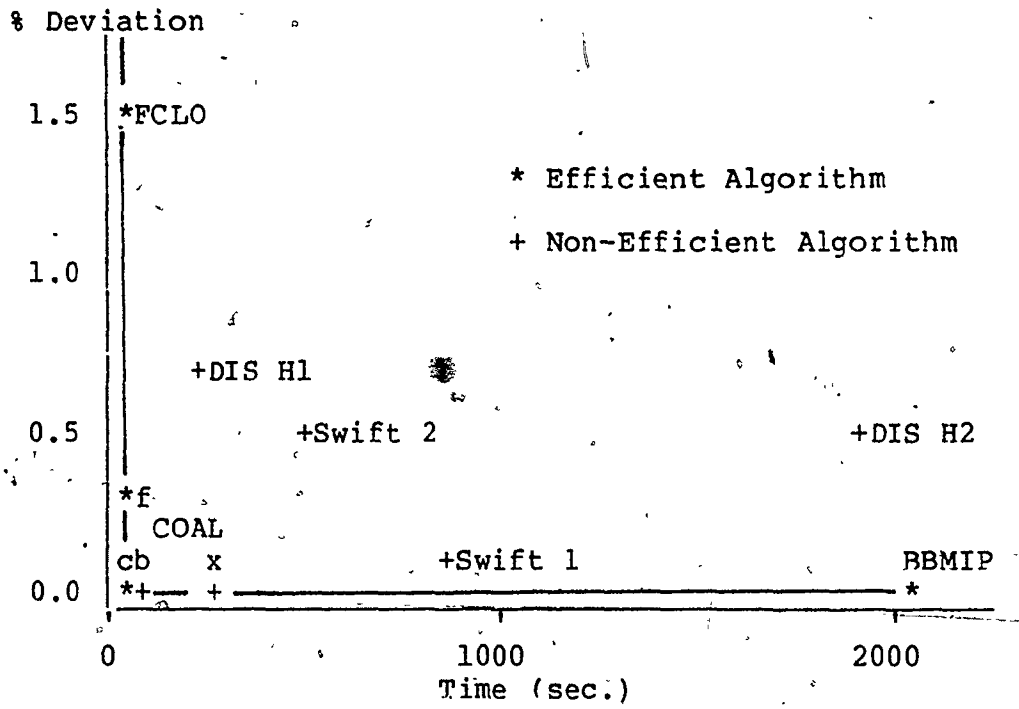


Exhibit 4-64: Strategic Manpower Planning
-- Efficient Algorithms

	Base Run	Fixed Charges * 2	Fixed Charges * 3	Fixed Charges * 4
L.P.	*	*	*	*
BBMIP	-	*	*	-
F.C.L.O.	*	*	*	*
DIS H 1	-	-	-	-
DIS H 2	-	-	-	-
Swift 1	-	-	-	-
Swift 2	-	-	-	-
COAL-b	-	-	-	-
COAL-x	-	-	-	-
COAL-c	*	*	*	-
COAL-f	-	*	*	*

* Efficient Algorithm (Best Quality for Solution Time)

- Non-efficient Algorithm (Alternative algorithm obtains a better quality solution in less time)

one simplex pivot. In (SMP), shifting training from one period to the next to save a set-up usually requires many simplex pivots.

As was observed in the results with the facility location problems, the Walker and Steinberg algorithms performance is not particularly good when the problem does not correspond well to a simplex operation. However, the new cost allocation algorithms do very well. The solution time is very short and the quality of the solution is very good.

4.8.3. Evaluation -- Manpower Planning

A summary of the efficient algorithms for the manpower planning problems (VWF) using the Base Run, (VWF) using "Set Up Case I" and (SMP) is given in Exhibit 4-65. The cost allocation algorithms are efficient for all three problems. L.P. and F.C.L.O. are also efficient for all three problems. Walker's Swift 1 is efficient for (VWF) using the Base Run. BBMIP is efficient for the one problem which is small enough to allow testing to be carried out.

The position of the algorithms on the efficiency frontier for (VWF) using the Base Run is very similar to (FCLSP) (Exhibit 4-49). However, the positioning of the algorithms for (VWF) based on "Set Up Case I" and (SMP) is shown in Exhibit 4-66 with L.P. and F.C.L.O. in the upper left, the cost allocation algorithms showing significant improvement with a small increase in time and BBMIP having a small improvement with a large increase in time. The adjacent extreme point heuristics, DIS H 1 and 2 and Swift 1 and 2, are up and to the left indicating a relatively poor performance. The performance of the adjacent extreme point algorithms is better with (VWF) than with (SMP). Of course (VWF) has more similarity to (FCLSP) where the simplex pivot will make a significant change to fixed charge variables which is exploited by the adjacent extreme point algorithms. However, as additional complexity is introduced in the form of another fixed charge sub-problem or groups of fixed charge variables, the performance of the adjacent extreme point algorithms deteriorates. The additional complexity which is handled by the linear programming algorithm has little impact on the performance on the cost allocation algorithms.

4.9. Summary

The efficient algorithms for all the different problem types are summarized in Exhibit 4-67. In order to be included in the efficient set, an algorithm should demonstrate the ability to generate good solutions faster than other algorithms for some, but not all, of the problems and being reasonably close to efficient algorithms for other problems in the same type.

Exhibit 4-65: Man Power Planning Problems -- Summary -- Efficient Algorithms

	(VWF) Base	(VWF) Set Up I	(SMP)
L.P.	*	*	*
BBMIP	-na-	-na-	*
F.C.L.O.	*	*	*
DIS H 1	–	–	–
DIS H 2	–	–	–
Swift 1	*	–	–
Swift 2	–	–	–
COAL-b	–	–	–
COAL-x	–	*	–
COAL-c	*	*	*
COAL-f	*	*	*

* Efficient Algorithm

Exhibit 4-66: Efficiency Frontier - Man Power Planning

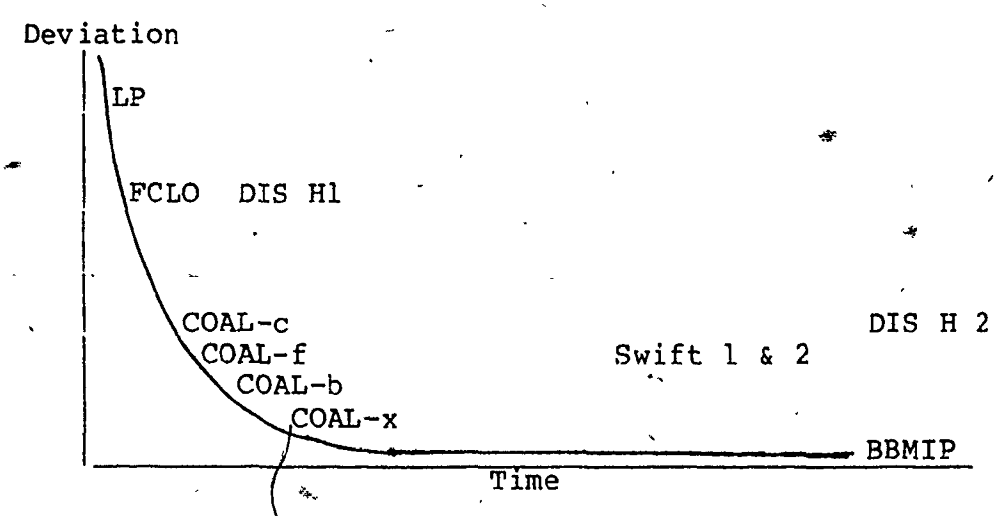


Exhibit 4-67: Summary -- Efficient Algorithms

	Random	Facility Location	Production Planning	Man Power Planning
L.P.	*	*	*	*
BBMIP	-na-	*	-na-	*
F.C.L.O.	*	*	*	*
DIS H 1	-	-	*	-
DIS H 2	*	-	*	-
Swift 1	*	-	*	-
Swift 2	-	-	-	-
COAL-b	-	*	*	-
COAL-x	*	*	*	-
COAL-c	*	*	*	*
COAL-f	*	*	*	*

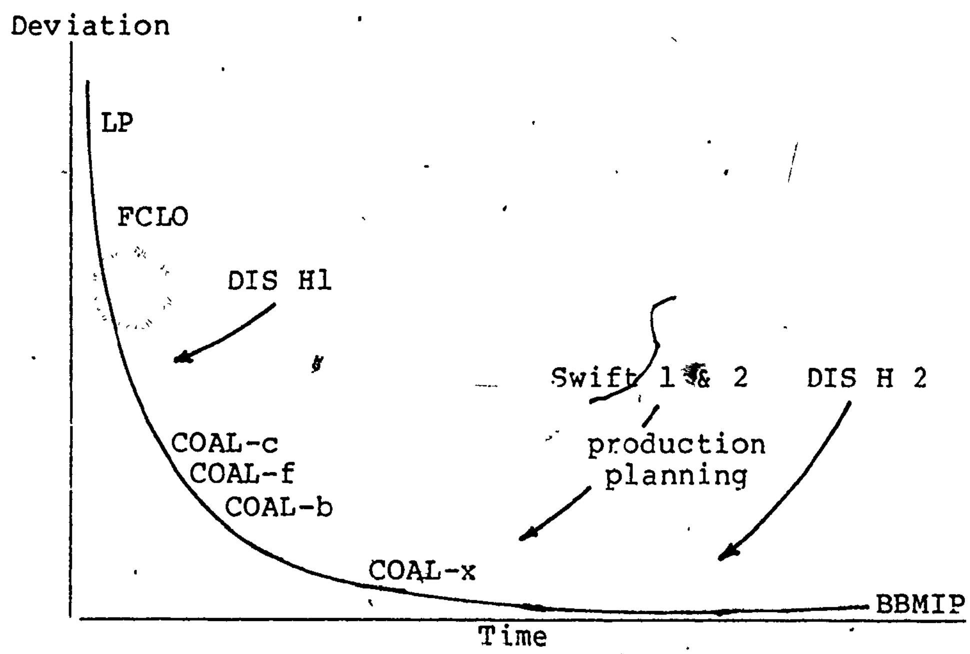
* Efficient Algorithm

L.P. and F.C.L.O. are efficient across all problem types. BBMIP is efficient for those problems for which are small enough for BBMIP to solve. There are efficient cost allocation algorithms for each of the problem types tested. However, the adjacent extreme point algorithms are efficient only for the production planning problems.

The position on the efficiency frontier for the different algorithms is relatively consistent across problem types (Exhibit 4-68). Linear Programming and the initial

Fixed Charge Local Optimum consistently obtain solutions with the lowest expenditure of cpu time. However, the quality of the solutions is low.

Exhibit 4-68: Efficiency Frontier - Fixed Charge Problem



The cost allocation algorithms, COAL-c, COAL-f and COAL-b, consistently obtain the greatest improvement in quality of solution with a small increase in solution time required to solve the problem. This consistency is observed across all different problem types. COAL-x, while improving the solutions from COAL-b, may require a large increase in computer time to solve the problem.

The adjacent extreme point heuristics, Steinberg's Heuristic 1 and 2 and Walker's Swift 1 and 2, are dominated by other algorithms which produce an equal or better quality solution using less cpu time for both facility location and manpower planning problems. However, the adjacent extreme point heuristics provide effective solutions for the production planning problem with solutions on the right showing some improvement over the cost allocation algorithms with a large increase in cpu time. The adjacent extreme point algorithms, DIS H 2, Swift 1 and 2, generally take much longer than the cost allocation algorithms. For facility location problems and manpower planning problems, the Walker algorithms and Steinberg's Heuristic 2 obtain poor solutions in spite of the lengthy computer runs. Only in production planning do these three algorithms obtain a small improvement over the cost allocation algorithms. Steinberg's Heuristic 1 usually takes a similar amount of time as COAL-c or COAL-f but usually does not improve the solution found by its initial phase represented by the initial Fixed Charge Local Optimum. Steinberg's Heuristic 1 generates significant improvement only in the production planning problems.

Of course, the branch and bound mixed-integer programming algorithms, BBMIP, consistently generates good solutions for the smaller problems. However, the cpu time require by BBMIP for the moderate size problems is prohibitive. Many of the problems tested are much to large for BBMIP to solve.

The quality of the solutions of the Steinberg or Walker algorithms, based on the adjacent extreme point search, is "good" only when the problem structure is such that a simplex iteration is significant as in the fixed cost transportation problem, the production planning problem or any small problem. When this is not the case, as in the capacitated warehouse location problem or the manpower planning problem, these algorithms generate poor solutions.

The solution times for the Walker and Steinberg algorithms are dramatically effected by an increase in continuous variables with out a fixed charge. As a result, DIS H 2, Swift 1 and 2 have execution times that are much longer than the cost allocation algorithms for all the applied problems which typically have many continuous variables with out a fixed charge.

The solution times for the Walker and Steinberg algorithms are relatively independent of the size of the fixed charges. The solution times for the cost allocation algorithms are consistent with the observations by Kennington [51], McGinnis [64] and Francis et. al. [31] that solution times increase with the size of the fixed charges. Or stated another way, the Steinberg and Walker algorithms do not take advantage of the small size of the fixed charges to improve their speed. The solution times decrease significantly for the cost allocation algorithms when the size of the fixed charges is decreased.

Solution time is a polynomial function of problem size for the cost allocation algorithms which avoids problems of exponential growth which limits the effectiveness of branch and bound methods. Memory requirements of the new algorithms are slightly larger than required by linear programming. COAL-c and COAL-f require storing one extra solution. COAL-b and COAL-f require two additional solutions. Therefore, memory requirements of the cost allocation algorithms do not pose a serious problem.

For the problems tested so far, COAL-c requires less time on average than COAL-f although the solutions generated by COAL-f are better on average than COAL-c. The maximum deviations by COAL-c are much larger than the maximum deviations with COAL-f which could account for the difference in average deviations. COAL-c with its single change can result in a poorer solution which is difficult to improve. Due to its initial phase which has a global perspective incorporating many fixed charges in one step, COAL-f avoids these solutions. However, COAL-f is not consistently better as COAL-c sometimes produces the better solution. The relative performance of COAL-c versus COAL-f is a function of the parameters of a problem as well as its structure. On the basis of the results to date, it is difficult to predict the conditions under which COAL-c will perform better than COAL-f or vice-a-versa. Since COAL-b is combination of COAL-c and COAL-f, it always produces solutions as good as and sometimes better than either at the expense of essentially solving the problem twice. COAL-x is similar to COAL-b with an extended local search which may find better solutions. However, this improvement can be expensive particularly when either (or both) COAL-c or COAL-f generate poor solutions. The problems inherent in the combinatorial nature of the search procedure in COAL-x become apparent.

For all the problem types tested, the cost allocation techniques produce an algorithm in the center of the efficiency frontier. One or more of the algorithms, COAL-c, COAL-f and COAL-b, is on the center portion of the efficiency frontier. Any of the three, COAL-c, COAL-f or COAL-b, which is not on the efficiency frontier is still relatively close. These algorithms produce "good" quality solutions while requiring modest amounts of computer time to obtain the solution. COAL-x produces better quality solutions but increases the execution time. This, of course, keeps COAL-x on the efficiency frontier but on the right hand side. COAL-x, on average for each problem type, obtains the best quality solutions of all the approximate methods. The cost allocation technique is very robust capable of efficiently solving a wide variety of large, general and applied fixed charge problems.

CHAPTER 5

CONCLUSION

5.1. Review

The fixed charge problem, originally defined by Hirsch and Dantzig [45], refers to a linear programming problem with one discontinuity in the objective function at an activity level of zero (Figure 2-1). Other discontinuities, such as economies of scale, price breaks, minimum threshold levels or fixed charges at different levels, can be modeled by reformulating the single fixed charge cost structure and thus adding relevance relevance to the fixed charge formulation.

The application areas for fixed charge problems discussed include facility location, production planning and manpower planning. Facility location problems involve the selection of facilities from a number of finite and predetermined possible sights. A number of constraints are imposed on the problem such as meeting demand and not exceeding capacity. Associated with the various facilities is a fixed charge which will be incurred if the facility is open and a cost which is a function of the volume processed. Facility location can be sub-divided into specialized fixed charge problems such as the capacitated warehouse location problem, the uncapacitated facility location problem and the fixed cost transportation problem. The fixed cost transportation problem has fixed charges associated with operating or using a particular route rather than a facility. While all facility location problems have these basic structures, many problems will have additional features or constraints and thus can not be classified as one of the specialized fixed charge problems.

Production planning decisions often require various set up procedures entailing a fixed charge which must be incurred before any quantity can be produced. Within production planning, specialized fixed charge problems are the fixed charge lot size problem, the single item lot size problem and the uncapacitated lot size problem. An overlap between production planning and manpower planning is created when the size of a work force becomes a decision variable. Hiring, training and firing of employees become decisions which may include fixed charges in their costs. The production/manpower planning problems typically have one variable representing the manpower level per period. However, many problems require different categories of manpower to represent different levels of experience and training. These features add additional complexity to standard manpower problems.

The fixed charge problem may be applied in other areas such as accounting, distribution planning and media selection in marketing or portfolio selection. The standard procedure is to apply a mixed-integer formulation to such problems and the fixed charge nature of the problem is not recognized or exploited.

A common procedure for solving a general fixed charge problem is to formulate as a mixed-integer problem and solve with a commercially available package. These packages employ branch and bound algorithms based on Land and Doig's [56] original work. Branch and bound algorithms have also been developed specifically for the general fixed charge problem. The algorithms involve enhancements to the basic Land and Doig method. While some improvement is noted, test results indicate that branch and bound algorithms for the fixed charge problem face the same difficulties as other branch and bound algorithms as problems become large. Optimal solutions to the general fixed charge problem have also been obtained by algorithms using cutting planes and vertex generation. However, limited success in solving larger problems has been reported.

Consequently, a number of approximate algorithms have been developed for solving general fixed charge problems. The basis of most of these algorithms is an adjacent extreme point search. The test results for the approximate algorithms, while promising, have been limited to a number of rather specialized problems.

In contrast to the lack of success for the general fixed charge problem, algorithms for solving specialized fixed charge problems, such as the capacitated warehouse location problem or the fixed charge lot-size problem, have met with a considerable success. Techniques which obtain the optimal solution are able to solve in reasonable time very large problems. Approximate algorithms for obtaining "good" solutions are also available for solving extremely large problems. A number of successful applications in industry are reported for both optimizing and approximate methods.

However many problems have features which make it impossible to use a specialized algorithm. While the methods available for solving a general fixed charge problem are adequate for small problems, difficulties arise as the problems become large. While size is strongly dependent on the number of fixed charges, other factors such as the number of equations and ordinary variables will have an impact on size and problem difficulty. There exists a need for a method of solving large general fixed charge problems particularly in light of the wide applicability of such a formulation.

The new COAL solution technique is developed as an approximate method for solving large general fixed charge problems. The COAL technique involves an allocation of the fixed charge to the continuous coefficient in the associated linear programming problem which can be solved by ordinary linear programming. A particular set of allocations will produce a solution to the associated linear programming problem and the fixed charge problem. A set of allocations and the resulting solution must meet necessary conditions for optimality.

While the necessary conditions define a set of possible solutions which could be optimal, sufficient conditions are required to prove a particular solution is optimal. Due to the combinatorial nature of the fixed charge problem, sufficient conditions which can be easily applied to large problems are difficult to develop. Therefore, a number of quasi-sufficient conditions are developed. These conditions, if met, will indicate that a particular solution, while not necessarily an optimal solution, is at least a good solution. The quasi-sufficient conditions, if not met, will indicate an improvement to be made to the current solution.

The necessary conditions and the quasi-sufficient conditions with a number of heuristics rules for calculating allocations are combined to form four different COst ALlocation (COAL) algorithms. Different aspects of fixed charge problems are incorporated into the design of each algorithm. As a result, each algorithm may generate different solutions with different computational requirements and will be useful in different circumstances.

An evaluation of the new COAL algorithms is carried out on a number of fixed charge problems. These include not only sample problems commonly used in the literature but, as well, a number of applied general and specialized fixed charge problems. A number of factors which cause difficulty in fixed charge problems, such as problem size, are varied to determine their impact on performance on the algorithms tested. A comparative analysis of the performance of the COAL algorithms with other algorithms used for solving large general fixed charge problems on the same computer is presented. All algorithms are implemented as accurately and efficiently as possible.

An efficiency frontier is developed for each of the problem areas to evaluate the various solution methods. The efficiency frontier displavs the trade off between the quality of the solution and the computational effort required. The new cost allocation technique falls in the center of the efficiency frontiers for all different application areas producing a substantial increase in quality with little additional requirement for resources.

5.2. Contribution

The methodology underlying the cost allocation technique is a significant departure from the current approximate methods of obtaining good solutions to general fixed charge problems. It is developed in a manner consistent with the nature of the fixed charge problem. The various steps involved in the new cost allocation technique apply only to the fixed charge variables. The value of any continuous variables without a fixed charge are determined by the more efficient linear programming algorithm. In addition, the standard method for solving large linear programming problems, the revised simplex with the product form of the inverse, is exploited in the actual implementation of the algorithms.

As well as evaluating the performance of the COAL techniques on its own, extensive testing of other approximate algorithms based on the adjacent extreme point search is carried out. Results for the adjacent extreme point algorithms on a variety of applied problems have not been reported in the literature. Thus, the appropriateness of these approximate methods for a number of application areas can be evaluated. This evaluation is made across a number of different problem areas which are large and applied.

The new cost allocation algorithms, COAL-b, COAL-c and COAL-f, consistently generate good solutions to the different problems tested. The four COAL algorithms are consistently on a very good position on the efficiency frontier. The algorithms obtain significant improvements in the quality of the solution for the increase in time.

The cost allocation algorithm, COAL-x, uses more cpu time to obtain a smaller incremental improvement in the quality of the solution. Since COAL-x is an extension of COAL-b, COAL-x always improves the solution from (or at least obtains as good as) COAL-b.

Both linear programming and the first phase of the adjacent extreme point heuristics which obtains the initial fixed charge local optimum are fast but the quality of the solution is poor. Branch and bound mixed integer programming (BBMIP), when it is capable of solving a problem, produces optimal solutions. The solution time for BBMIP is an exponential function of problem size. Therefore, as size increases, branch and bound techniques become impractical. For all other algorithms, solution time is a polynomial function of size.

For facility location problems, the cost allocation algorithms, COAL-b, COAL-c and COAL -f, generate high quality solutions for the cpu time required. The COAL algorithms dominate the Steinberg and Walker adjacent extreme point algorithms obtaining better quality solutions in less time. BBMIP, while very effective for small problems, has solution times which are an exponential function of size and becomes impractical for larger problems.

In production planning problems, the cost allocation algorithms are again in a central position on the efficiency frontier. The problems used for production planning are too large to be solved within practical limits by BBMIP. The adjacent extreme point algorithms obtain good quality solutions to production planning problems. However, the execution time required for a marginal improvement in quality over the cost allocation algorithms is very large.

Manpower planning problems display more complexity in their structure than production planning. Very good solutions are obtained with reasonable computational effort by COAL-c and COAL-f. As such, it is difficult for COAL-b and COAL-x to improve the quality in these particular problems. However, the COAL algorithms are again on the efficiency frontier. The additional complexity of the manpower planning problems create difficulties for the adjacent extreme point algorithms. Again, the cost allocation algorithms dominate the Steinberg and Walker adjacent extreme point algorithms. BBMIP, which obtains optimal solutions for the smaller problems requires very long solution times. The larger manpower planning problems could not be solved by BBMIP within practical limits.

The cost allocation algorithms are consistently on the efficiency frontier for all different problem areas. COAL-b, COAL-c and COAL-f obtain good solutions with reasonable execution time. COAL-x, while requiring more execution time, generates the best solutions by an approximate algorithm for nearly all the problems.

5.3. Further Research

Several heuristic rules are employed in the design of the new COAL algorithms. These heuristics rules involve choices which will affect the performance of an algorithm. The choices made keep in mind the intent of solving large and general fixed charge problems. However, these choices could be the subject of further research.

A modification in the design of COAL-x may result in an increase in efficiency. COAL-x applies the combination quasi-sufficiency test twice: once on the solution at the end of Phase 3 and again on the solution from the end of Phase 5. The very lengthy runs occasionally required by COAL-x arise when a poor solution from either Phase 3 or 5 must be improved with several iterations. However, the intent of COAL-x is to obtain the best possible solution and places a high priority on quality. In order to maximize the likelihood of obtaining an improved solution, COAL-x applies the combination quasi-sufficiency test twice. An algorithm using one combination quasi-sufficiency test on the solution from COAL-b only would take less time than COAL-x with a small decrease in the quality of solutions.

COAL-c uses an initial solution dominated by the continuos costs while COAL-f uses an initial solution dominated by the fixed charges. In spite of testing on a wide variety of problems, it is not shown conclusively that

COAL-c performs better on problems dominated by continuous costs or that COAL-f performs better on problems dominated by fixed charges. However, the comparison of COAL-c and COAL-f is not the subject of the testing. Further research could be productive in identifying the conditions when COAL-c (or COAL-f) would be the better choice.

In another heuristic rule, the largest improvement with each quasi-sufficiency test is used to generate an improved solution. The choice of the largest improvement at each iteration has considerable intuitive appeal and consequentlv is selected. The testing of one allocation in the new algorithms requires a considerable effort. The testing of all single or combination changes to find the largest improvement requires a major effort. An improvement in efficiency could be made by limiting this effort in a logical fashion. For example, quasi-sufficiency tests could be restricted to those changes which produced a positive improvement in a previous iteration. For the COAL-x algorithms, which mav be required to apply the quasi-sufficiency to many combinations of fixed charge variables, of fixed charge variables, the potential savings of such an approach could be substantial.

There are many possibilities for examining multiple changes. For example, the initial solution dominated by fixed charges (Exhibit 3-10) performs a series of multiple changes to the first solution from solving the associated linear programming problem. A similar allocation could be made during other phases simultaneously to all positive fixed charge variables. Since this process seems to be counter to the nature of the fixed charge problem as outlined in the necessary conditions, it is only used to obtain the initial solution dominated by fixed charges. However, several cost allocation iterations could be combined into one iteration with a possible gain in efficiency. The process could prove effective in achieving a multiple change.

Another possible area for further research deals with different initial conditions. For example, the Balinski approximation could be be used as a starting point for either Phase 1 or 3. Since the Balinski approximation works well only with good upper bounds, which are not available for most of the problems examined, it does not seem appropriate and is not used in the current algorithms. For appropriate problems, it could be very productive at little cost for computational requirements.

The initial fixed charge local optimum is always obtained relatively quickly and an adjacent extreme point iteration is faster than an iteration in one of the cost allocation algorithms. Thus, the possibility of incorporating the adjacent extreme point search into a cost allocation algorithm could be considered, particularly when the structure is suitable, e.g. the fixed charge lot size problem (FCLSP). The primary reason for not incorporating an adjacent extreme point search was to differentiate the cost allocation algorithms and evaluate them on their own. Incorporating an adjacent extreme point search into the cost allocation techniques has the potential for generating further improvement. However, this improvement will be more difficult than first appearances would indicate. The cost allocation algorithms require a positive allocation on many fixed charge variables in order to keep them out of the solution. The adjacent extreme point search does not produce such an allocation which will have to be created.

The solution of the associated linear programming problem with the cost allocations is currently being handled by the revised simplex method using the product form of the inverse. However, there is no specific requirement to always use a linear programming algorithm as a method of solving the associated linear programming problem. For example, a more efficient network algorithm could be used if the structure is appropriate. Other methods could also include non-linear solution methods such as quadratic programming. However, the actual implementation would vary from problem to problem in order to exploit the structure thus becoming a specialized algorithm.

5.4. Summary

The four cost allocation algorithms developed in this thesis are consistently on or close to the efficiency frontier for all the problem areas tested. The three algorithms, COAL-c, COAL-f and COAL-b, provide good solutions in al application areas. They significantly improve the quality of the solutions obtained by simple linear programming or the initial fixed charge local optimum with modest increases in execution times. The adjacent extreme point algorithms require much longer execution times but achieve a marginal improvement in quality for production planning problems only. Branch and bound mixed-integer programming, while achieving the optimum solution, required much longer execution times for the moderately large problems and proved impractical for the largest problems tested.

COAL-x, by design, consistently shows an improvement over the other three cost allocation algorithms. However, this is achieved at the expense of lengthy computer runs for some problems. Overall, COAL-x achieves the best quality solutions of the approximate algorithms tested and should be used when quality is of prime importance.

COAL-c is, on average, faster than COAL-f while COAL-f produces, on average, better solutions. COAL-c occasionally produces slightly larger deviations which result from its single change search procedure. However, COAL-f is not consistently better than COAL-c. While the results tend to indicate that COAL-c performs better with problems dominated by the continuous costs, this is not conclusive despite the extensive testing on a wide variety of problems. COAL-b combines both COAL-c and COAL-f and therefore has a quality of solution as good as or better than the best of COAL-c or COAL-f. However, COAL-b has approximately twice the computational requirements of COAL-c or COAL-f on their own.

The new cost allocation algorithms have demonstrated a robustness in solving a wide variety of large general fixed charge problems. The flexibility inherent in the fixed charge formulation gives the technique wide applicability. The cost allocation method is a significant departure from other approximate methods for solving general fixed charge problems. The four new COAL algorithms demonstrate a significant improvement over current methods for solving large general fixed charge problems.

Appendix A: Heuristic One - Steinberg

1. Obtain an initial solution of the associated linear programming problem (ALP). Set control parameters $a_0=m/2$, and $b_0=m/2$.
2. Find the fixed charge local optimum of the current solution. Call the resulting solution x_0, with objective function value z_0. Proceed to Step 3.
3. Find the non-basic variable j which will yield the smallest increase in the objective function z. Insert the corresponding x_j into the basis. Set $a_1=1$ and $b_1=1$. Go to Step 4.
4. Select any non-basic variable j which will decrease the objective function. Insert the corresponding x_j into the basis. Call the resulting solution x_1, with objective function value z_1. Proceed to Step 5.
5. Compare z_0 with z_1:
   a. If $z_0>z_1$, return to Step 2.
   b. If $z_0=z_1$ but $x_0\neq x_1$, set $a_1=a_1+1$. If $a_1<a_0$, return to Step 4; otherwise, terminate.
   c. If $z_0=z_1$ and $x_0=x_1$, set $a_1=a_1+1$ and $b_1=b_1+1$. If $a_1\leq a_0$ and $b_1\leq b_0$, proceed to Step 6; otherwise, terminate.
   d. If $z_0<z_1$, set $a_1=a_1+1$. If $a_1<a_0$, return to 4; otherwise, terminate.

Appendix A: Heuristic One – Steinberg (continued)

6. Perform b_1 consecutive iterations in each of which the variable which yields the largest increase in the objective function z is inserted into the basis. Return to Step 4.

Appendix B: Heuristic Two - Steinberg

1. Obtain an initial solution of the associated linear programming problem (ALP). Set control parameters $a_0=m/2$, and $b_0=n-m$.
2. Find the fixed charge local optimum of the current solution. Call the resulting solution x_0, with objective function value z_0. The entire simplex tableau corresponding to x_0 is saved. Set $b_1=1$. Proceed to Step 3.
3. Beginning with the tableau corresponding to x_0, find the variable j which will yield the (b_1)'th smallest increase in the objective function z. Insert the corresponding x_j into the basis. Set $b_1=b_1+1$. If $b_1 \leq b_0+1$, proceed to step 4; otherwise terminate.
4. Select any variable j which will decrease the objective function and insert the corresponding x_j into the basis. Call the resulting solution x_1 with objective function value z_1. Set $a_1=a_1+1$. If $a_1<a_0$, then proceed to step 5; otherwise, return to step 2.

Appendix B: Heuristic Two - Steinberg (continued)

5. Compare z_0 with z_1:
   a. If $z_0<z_1$, return to Step 4.
   b. If $z_0=z_1$ but $x_0 \neq x_1$, return to Step 4.
   c. If $z_0=z_1$ and $x_0 \neq x_1$, set $a_1=1$. Return to Step 3.
   d. If $z_0>z_1$, return to Step 2.

Appendix C: Swift 1 - Walker

1. Obtain an initial solution of the associated linear programming problem (ALP).
2. Find the fixed charge local optimum of the current solution. Call the resulting solution x_0, with objective function value z_0. Proceed to Step 3.
3. Force a currently non-basic variable, not yet tried, into the basis, yielding a new solution, x_1 with objective function value $z_1 \geq z_0$. If all non-basic variables in solution x_0 have been tried without an improvement, stop and call x_0 the (approximate) solution; otherwise, go to Step 4.
4. Iterate as in Step 2, until a fixed charge local optimum is found. Call this solution x_1.
   a. If $x_0=x_1$ (i.e., no iterating was possible), return to solution x_0. Return to Step 3.
   b. If $z_0>z_1$, a better solution has been found. Rename this solution x_0. Return to Step 3.
   c. If $z_0 \leq z_1$, return to Step 3.

Appendix D: Swift 2 - Walker

1. Obtain an initial solution of the associated linear programming problem (ALP).
2. Find the fixed charge local optimum of the current solution. Call the resulting solution x_0, with objective function value z_0. Proceed to Step 3.
3. Force a currently non-basic variable, not yet tried, into the basis, yielding a new solution, x_1 with objective function value $z_1 \geq z_0$. If all non-basic variables in solution x_0 have been tried without an improvement, stop and call x_0 the (approximate) solution; otherwise, go to Step 4.
4. Iterate as in Step 2, until a fixed charge local optimum is found. Call this solution x_1.
   a. If $x_0=x_1$ (i.e., no iterating was possible), return to solution x_0. Return to Step 3.
   b. If $z_0>z_1$, a better solution has been found. Rename this solution x_0. Return to Step 3.
   c. If $z_0 \leq z_1$, return to solution x_0 (the best solution so far). Go to Step 3.

Appendix E: Random 5 x 10 Problems

maximize $z = \sum_j (c_j X_j + f_j Y_j)$

subject to:

$$\sum_j a_{ij} X_j = b_i \qquad \forall\ i$$

$$Y_j = \begin{cases} 1 \text{ if } X_j > 0 \\ 0 \text{ if } X_j = 0 \end{cases} \qquad \forall\ j$$

Problem 1: Used in "=", ">" and "<" types

f_j	17	676	220	729	152	183	595	942	759	139
c_j	14	11	16	10	20	19	20	12	1	18
b_i	a_{ij}									
500	15		6	16	9		17	-13	5	20
280	3		5		-8	-18	13	11	2	-2
466		7				3			19	
308	-10	13	-6				10			
520	11	19		20			20	8	-4	

Problem 2: Used in "=", ">" and "<" types

f_j	738	198	826	33	68	646	797	688	833	629
c_j	13	11	10	19	2	13	7	20	4	1
b_i	a_{ij}									
540					20	13		-5	1	
211	11	-2		-5	14	-17			12	-2
583			-12	11		10		13	5	18
563		20		3	-8		16			20
829	17	8	12		7				9	

Appendix E: Random 5 x 10 Problems (continued)

Problem 3: Used in "=", ">" and "<" types

f_j	581	194	219	137	301	369	345	860	200	380
c_j	5	3	4	8	15	15	17	5	20	17
b_i	a_{ij}									
257					13			2		-1
829	15	6	14			14			15	
563	3	16		8	10	-5	14	-2		
323				3	2				5	7
177			-7		17		-8	-7		

Problem 4: Used in "=", ">" and "<" types

f_j	385	487	252	343	519	74	842	415	543	411
c_j	15	18	9	14	11	1	11	5	12	16
b_i	a_{ij}									
130	18		2	-18	-17			-18	-10	14
454	2	7	4	-1			19	14		
105		-16	1	18		13	15	-14		
607	17		-5	-6	7			3	16	10
492	12	15		-8		-4	12		-11	

Problem 5: Used in "=", ">" and "<" types

f_j	944	940	193	118	551	33	629	290	517	198
c_j	14	20	15	4	4	4	11	14	2	13
b_i	a_{ij}									
266				19						
20				8	-11		4	5		
953	6	13	15	4	14	15		13	13	
23	13	-19	-6	-2			15			-18
489	14		1		16		-4			

Appendix E: Random 5 x 10 Problems (continued)

Problem 6: Used in "=", ">" and "<" types

f_j	916	920	340	608	28	160	542	561	229	375
c_j	11	5	15	13	5	18	4	6	5	19
b_i	a_{ij}									
833	18	1	9	-19	18	15	9		11	11
351			-4				4	19		11
44	5		19		-16			-15		
583	-19	9	-18	20		2	13	6	6	20
549	-7				-1		15	12	3	15

Problem 7: Used in "=", ">" and "<" types

f_j	899	334	402	414	190	757	77	845	161	671
c_j	15	1	11	11	1	8	18	7	14	9
b_i	a_{ij}									
2	-15	-15			-11		16		3	19
254					2	11				
580		16		-4		16		10	2	1
486			15	20	4			-4	7	6
848	19	15		-8	15		11			13

Problem 8: Used in "=", ">" and "<" types

f_j	420	600	935	661	801	373	862	246	63	207
c_j	7	9	8	4	15	20	18	7	9	4
b_i	a_{ij}									
44		-18		-17	5	9		11	17	
459		9			-15	19	12		16	
495	17	9			12	-14		11		5
276		-20	3		10	-18	-9	18	19	
486	10			7		19		10		

Appendix E: Random 5 x 10 Problems (continued)

Problem 9: Used in "=", ">" and "<" types

f_j	506	953	951	116	557	647	858	751	766	352
c_j	9	7	19	8	5	14	8	3	19	16
b_i	a_{ij}									
183	-8	5						14		
316	10	18	9		2		10	-8	14	
589			17		3	14		19		
253	13		1		-18	-3				7
290	-18			12			17	10		

Problem 10: Used in "=", ">" and "<" types

f_j	421	615	965	327	991	720	547	262	62	888
c_j	18	4	15	8	11	11	18	18	17	2
b_i	a_{ij}									
306		3	9	-12	10	3	16	1		
47	2	-11	-7		17		-13		16	
391	5	6	-10	19	14	13			17	-1
101		8		19	-2			-5		
65				2		-2				9

Problem 11: Used in "=", ">" and "<" types

f_j	836	267	611	436	744	674	540	922	225	501
c_j	10	18	11	1	20	2	8	14	1	6
b_i	a_{ij}									
28			-13			-9		11	15	
208	5	5				16		16		
300	19		-18				19			8
194	-18	-4	9	13			9	14		
237	13	16	10		3		-6	-8	-6	14

Appendix E: Random 5 x 10 Problems (continued)

Problem 12: Used in "=" and ">" types

f_j	121	460	968	950	439	650	916	334	777	137
c_j	3	12	14	7	15	3	6	20	2	7
b_i	a_{ij}									
63	-1		-18	-4	18	3			6	-9
223		9			-7	8	8	-1		11
486	19			17			7		11	
68	-14		-12		17			10		
3				12	-16		9	-2	-5	-6

Problem 12a: Used in "<" types

f_j	121	460	968	950	439	650	916	334	777	137
c_j	-2	-11	-13	-6	-14	-2	-5	-19	-1	-6
b_i	a_{ij}									
63	-1		18	-4	-18	3			6	-9
223		9			-7	8	8	-1		11
486	19			17			7		11	
68	-14		-12		17			10		
3				12	-16		9	-2	-5	-6

Problem 13: Used in "=" and ">" types

f_j	397	598	866	554	451	571	310	665	598	425
c_j	16	16	12	3	14	10	13	18	7	7
b_i	a_{ij}									
314	12	8	13			-6	3			
5	2			-3	-16		16		14	-14
14	2			-20		3	15	-5	11	
290	1						13		14	
83			10		9	4	-8		1	

Appendix E: Random 5 x 10 Problems (continued)

Problem 13a: Used in "<" types

f_j	397	598	866	554	451	571	310	665	598	425
c_j	-15	-15	-11	-2	-13	-9	-12	-17	-6	-6
b_i	a_{ij}									
314	12	8	13			-6	3			
5	2			3	16		16		14	14
14	2			-20		3	15	5	11	
290	1						13		14	
83			10		9	4	-8		1	

Problem 14: Used in "=" and ">" types

f_j	103	486	586	741	560	923	464	921	906	651
c_j	12	1	13	6	1	8	20	4	19	9
b_i	a_{ij}									
398	-3	4	16	-3			-5	20		1
289	-13	15		1	8			17	18	
38			-5	-7			7	14		4
365			15		-11	-2		7	18	-9
531	17	-2	15		6	12	-4			2

Problem 14a: Used in "<" type

f_j	103	486	586	741	560	923	464	921	906	651
c_j	-11		-12	-5		-7	-19	-3	-18	-8
b_i	a_{ij}									
398	3	4	16	-3			-5	20		1
289	13	15		1	8			17	18	
38			-5	-7			7	14		4
365			15		-11	-2		7	18	-9
531	17	-2	15		6	12	-4			2

Appendix E: Random 5 x 10 Problems (continued)

Problem 15: Used in "=", ">" and "<" types

f_j	596	332	307	879	586	371	388	41	307	136
c_j	18	18	14	1	17	11	6	3	17	19
b_i	a_{ij}									
126				2	5				4	
112	13		-11					-13	13	
900	8	2	17	13		4	15	13		17
308	8		14	5	4	4		17	-20	
321	17	-11				-11	11		17	-11

Appendix F: Waste Disposal Problems

Waste Generating Centers

Center	Demand (tons/day)	Haul Cost ($/ton/hour)
1	2722	2.85
2	305	2.85
3	2077	2.85
4	2499	2.85
5	95	2.85
6	904	2.85
7	2750	2.85
8	968	2.85
9	5181	0.85
10	5976	0.85

Waste Treatment Centers

	Type	Capacity (tons/day)	Haul Cost ($/ton/hour)	Fraction Left	Fixed ($)	Variable ($/ton)
1	Int.	20000	1.14	0.32	11000	3.50
2	Int.	8000	1.14	0.10	9000	0.92
3a	Int.	9000	0.85	0.30	0	10.17
3b	Int.	9000	0.85	0.30	14530	2.91
4	Int.	8000	1.14	0.05	20000	1.75
5	Int.	8000	0.85	0.25	4400	5.79
6	Final	6000	--	--	50000	104.85
7	Final	8000	--	--	50000	104.85

Problems

Size	Number of Problems	Generating Centers	Treatment Centers	Arc Density
1	3	10	7	100%
1	3	10	7	80%
2	1	20	14	100%
2	3	20	14	80%
2	3	20	14	60%
3	3	30	21	40%

Notes -Variable cost of hauling and treating waste is calculated from haul cost times hauling time plus the variable treating cost.

-Distance is determined by randomly placing each center on a grid 3.5 hours by 3.5 hours hauling time.

Appendix F: Waste Disposal Problems (continued)

-Treatment Center 3 is modeled using economies of scale.

-For Size 2 Problems, the generating centers and treatment centers given above are repeated twice. For Size 3 Problems, they are repeated three times.

-Each possible arc (generating center-treatment center or intermediate treatment center-final treatment center combination) has a probability of being infeasible (1.0 - Arc Density).

Appendix G: Capacitated Warehouse Location Problems

Demand Centers

Center	Demand
1	10
2	20
3	10
4	30

Supply Centers

Center	Capacity
1	20
2	15
3	20
4	15
5	25
6	20
7	25

Problems

Size	Number of Problems	Demand Centers	Supply Centers	Arc Density
1	6	4	7	100%
1	6	4	7	80%
2	6	8	14	100%
2	6	8	14	80%
2	6	8	14	60%
3	6	12	21	100%
3	6	12	21	80%
3	6	12	21	40%
4	3	16	28	100%
4	3	16	28	80%
4	3	16	28	30%

Notes –Variable cost is a randomly chosen integer number between and including 3 and 7.

–Fixed Cost is a randomly chosen integer number between 70 and 140.

–For Size 2 Problems, the supply centers and demand centers given above are repeated twice. For Size 3 Problems, they are repeated three times. For Size 4 Problems, they are repeated four times.

–Each possible arc (demand center-supply center combination) has a probability of being infeasible (1.0 – Arc Density).

BIBLIOGRAPHY

1. Afentakis, P., B. Gavis and U. Karmarkar (1984), "Computationally Efficient Optimal Solutions to the Lot-Sizing Problem in Multi-Stage Assembly Systems", Management Science, Vol. 30, No. 3, p222-39.

2. Akinc, U. and B. M. Khumawala, (1977), "An Efficient Branch and Bound Algorithm for the Capacitated Warehouse Location Problem", Management Science, Vol. 23, No. 6, p585-94.

3. Baker, K. R., P. Dixon, M. J. Magazine, and E. A. Silver, (1978), "An Algorithm for the Dynamic Lot-Size Problem with Time-Varying Capacity Constraints", Management Science, Vol. 24, No. 6, p1710-20.

4. Balinski, M. L., (1961), "Fixed Cost Transportation Problem", Naval Research Logistics Quarterly, Vol. 8, No. 1, p41-54.

5. Balinski, M. L., (1964), "On Finding Integer Solutions to Linear Programs", Mathematica, Princeton, N. J., May, 1964.

6. Barany, Imre, Tony J. Van Roy and Laurence A. Wolsey, (1984), "Strong Formulations for Multi-Item Capacitated Lot Sizing", Management Science, Vol. 30, No. 10, p1255-1261.

7. Barr, Richard S., Fred Glover and Darwin Klingman, (1981), "A New Optimization Method for Large Scale Fixed Charge Transportation Problems", Operations Research, Vol. 29, No. 3, p448-463.

8. Bitran, G. R., and A. C. Hax, (1977), "On the Design of Hierarchical Production Planning Systems", Decision Sciences, Vol. 8, No. 1, p28-55.

9. Blackburn, Joseph D., and Robert A. Millen, (1982), "Improved Heuristics for Multi-Stage Requirements Planning Systems", Management Science, Vol. 28, No. 1, p41-56.

10. Calantone, R. J. and U. de Brentani-Todorovic, (1981), "The Maturation of the Science of Media Selection", Journal of the Academy of Marketing Science, Vol. 9, No. 4, p490-524.

11. Christofides N. and J. E. Beasley, (1983), "Extension to a Lagrangean Relaxation Approach for the Capacitated Warehouse Location Problem", European Journal of Operations Research, Vol. 12, p19-28.

12. Cooper, L., (1964), "Heuristic Methods for Location Allocation Problems", SIAM Review, Vol. 6, No. 1, p37-53.

13. Cooper, L. and C. Drebes, (1967), "An Approximate Solution Method for the Fixed Charge Problem", Naval Research Logistics Quarterly, Vol. 14, No. 1, p101-113.

14. Cooper, M. W. and K. Farhangian, (1982), "An Integer Programming Algorithm for Portfolio Selection with Fixed Charges", Naval Research Logistics Quarterly, Vol. 29, No. 1, p147-50.

15. Cohen, Claude and Jack Stein, (1978), Multi-Purpose Optimization System User's Guide, Version 4, Vogelback Computing Center, Northwestern University.

16. Crowston, W. B. and M. H. Wagner, (1973), "Dynamic Lot Size Models for Multi-Stage Assembly Systems", Management Science, Vol. 20, No. 1.,

17. Dantzig, G. B., (1963), Linear Programming and Extensions, Princeton University Press, Princeton, N. J.

18. Davis, P. S. and T. L. Ray, (1969), "A Branch and Bound Algorithm for the Capacitated Facilities Location Problem", Naval Research Logistics Quarterly, Vol. 19, p331-44.

19. Denzler, D. R., (1969), "An Approximative Algorithm for the Fixed Charge Problem", Naval Research Logistics Quarterly, Vol. 16, No. 3, p411-416.

20. Dixon P. S. and E. A. Silver, (1981), "A Heuristic Solution Procedure for the Multi-Stage Single-Level Limited Capacity Lot-Sizing Problem", Journal of Operations Management, Vol. 2, p23-40.

21. Dutton, Ron, George Hinman and C. B. Milhan, (1972), "The Optimal Location of Nuclear Power Facilities in the Pacific Northwest", Operations Research, Vol. 20, p479-487.

22. Dzielinski, B. P. and R. E. Gomory, (1965), "Optimal Programming of Lot Sizes, Inventory and Labor", Management Science, Vol. 11, p874-90.

23. Edwards, John S., (1983), "A Survey of Manpower Planning Models and Their Applications", Journal of the Operational Research Society, Vol. 34, No. 11, p1031-1040.

24. Eisenhut, P. S., (1975), "A Dynamic Lot-Sizing Algorithm with Capacity Constraints", AIIE Transactions, Vol. 7, p170-176.

25. Erlenkotter, D., (1978), "A Dual Base Procedure for Uncapacitated Facility Location", Operations Reasearch, Vol. 26, p992-1009.

26. Erlenkotter, D., (1981), "A Comparative Study of Approches to Dynamic Location Problems", European Journal of Operations Research, Vol. 6, p133-143.

27. Falk, Patrick G., (1980), "Experiments in Mixed Integer Linear Programming in a Manufacturing System", Omega, Vol. 28, No. 4, p473-484.

28. Feldman, E., F. A. Lehrer and T. L. Ray, (1966), "Warehouse Location under Continuos Economies of Scale", Management Science, Vol. 12, No. 9, p12-20.

29. Fielitz, Bruce D. and Daniel L. White, (1981), "Two Stage Solution Procedure for the Lock Box Location Problem", Management Science, Vol. 27, No. 8, p881-886.

30. Florian, M. and M. Klein, (1971), "Deterministic Production Planning with Concave Costs and Capacity Constraints", Management Science, Vol. 18, No. 1, p287-313.

31. Francis, R. L., L. F. McGinnis, J. A. White, "Locational Analysis", (1983), European Journal of Operations Research, Vol. 12, p220-252.

32. Fould, L. R., (1983), "The Heuristic Problem Solving Approach", Journal ~~of~~ the Operational Research Society, Vol. 34, No. 10, p927-934.

33. Geoffrion, A. M., (1975), "A Guide to Computer-Assisted Methods for Distribution Systems Planning", Sloan Management Review, Vol. 17, No. 2, p17-41.

34. Geoffrion, A. M. and G. W. Graves, (1974), "Multicommodity Distribution System Design by Benders Decomposition", Management Science, Vol. 20, No. 4, p822-844.

35. Geoffrion, A. M., G. W. Graves and S. J. Lee, (1982), "A Management Support System for Distribution Planning", Infor, Vol. 20, No. 4, p287-313.

36. Graves, Stephen C.,(1982), "Using Lagrangean Techniques to Solve Hierarchical Production Planning Problems", Management Science, Vol. 28, No. 3, p260-275.

37. Gray, Paul, (1967), "Mixed Integer Programming Algorithms for Site Selection and Other Fixed Charge Problems Having Capacity Constraints", Phd. Dissertation, Stanford University.

38. Haehling von Lanzenauer, Christoph, Richard Horwitz and Don D. Wright (1978), "Manpower Planning, Mathematical Programming and the Development of Policies", Studies in Operations Management, Arnoldo C. Hax, Ed., North-Holland.

39. Haessler, Robert W., (1983), "Developing an Industrial Grade Heuristic Problem Solving Procedure", Interfaces, Vol. 13, No. 3, p62-71.

40. Hansen, D. R. and S. G. Taylor, (1983), "Optimal Production Strategies for Plants with Multiple Shutdown Levels and Multiple Production Lines", IIE, Vol. 15, No. 1, p46-53.

41. Hax, A. C., (1978), "Aggregate Production Planning", Handbook of Operations Research, J. Moders and S. Elmaghraby, Editors, Van Nostran Reinhold.

42. Hax, A. C. and J. J. Golovin, (1978), "Hierarchical Production Planning System", Studies in Operations Management, Arnoldo C. Hax, Ed., North-Holland.

43. Hax, A. C. and H. C. Meal, (1975), "Hierarchical Integration of Production Planning and Scheduling", TIMS Studies in Management Sciences, Vol. 1, Logistics, Murray Geisler, Ed., North-Holland.

44. Hiraki, Shusaku, (1980), "A Simplex Procedure for a Fixed Charge Problem", Journal of the Operations Research Society of Japan, Vol. 23, No. 3, p243-266.

45. Hirsch, W. M., and G. B. Dantzig, (1968), "The Fixed Charge Problem", Naval Research Logistics Quarterly, Vol. 15, No. 3, p413-468.

46. Jacobsen, S. K., (1983), "Heuristics for the Capacitated Plant Location Model", European Journal of Operations Research, Vol. 12, p253-61.

47. Jagnathan, R. and M. R. Rao, (1973), "A Class of Deterministic Production Planning Problems", Management Science, Vol. 19, No. 11, p1295-1300.

48. Jarvis, J. J., E. Unger, R. L. Rardin and R. W. Moore, (1978), "Optimal Design of Regional Waster Water System: A Fixed Charge Network Flow Model", Operations Research, Vol. 26, p538-550.

49. Jenkins, L., (1979), "Optimal Location of Facilities for Recycling Municipal Solid Waste in Southern Ontario", Phd. Thesis, Department of Industrial Engineering, University of Toronto.

50. Jenkins, L., (1982), "Parametric Mixed Integer Programming: An Application to Solid Waste Management", Management Science, Vol. 28, No. 11, p1270-84.

51. Kennington, J., (1976), "The Fixed-Charge Transportation Problem: A Computational Study with a Branch and Bound Code", AIIE Transactions, Vol. 8, No. 2, p241-247.

52. Kennington, Jeff and Ed Unger, (1976), "A New Branch-and-Bound Algorithm for the Fixed-Charge Transportation Problem", Management Science, Vol. 22, No. 10, p1110-1116.

53. Khumawala, B. M., (1972), "An Efficient Branch and Bound Algorithm for the Warehouse Location Problem", Management Science, Vol. 18, No. 12, pB718-31.

54. Kuehn, A. A. and M. J. Hamburger, (1963), "A Heuristic Program for Locating Warehouses", Management Science, Vol. 9, p643-666.

55. Lambrecht, M. and H. Vanderveken, (1979), "Heuristic Procedures for the Single Operation Multi-Item Loading Problem", AIIE Transactions, Vol. 11, p319-26.

56. Land, A., and A. Doig (1960), "An Automatic Method of Solving Discrete Programming Problems", Econometrica, Vol. 28, No. 3, p497- 520.

57. Lasden, R. L. and R. C. Terjung, (1971), "An Efficient Algorithm for Multi-Item Scheduling", Operations Research, Vol. 19, p946-969.

58. Lasdon L. S., (1970), Optimization Theory for Large Systems, The MacMillan Company, N. Y.

59. Love, S. F., (1973), "Bounded Production and Inventory Models with Piecewise Concave Costs", Management Science, Vol. 20, No. 3, p313-318.

60. Mangiameli, P. M. and L. J. Krajewski, (1983), "The Effects of Workforce Strategies on Manufacturing Operations", Journal of Operations Management, Vol. 3, No. 4, p183-196.

61. Manes, R. P., S. H. Park and R. Jensen, (1982), "Relevant Costs of Intermediate Goods and Services", The Accounting Review, Vol. 17, No. 3, p594-606.

62. Manne, A. J., (1958), "Programming of Economic Lot Sizes", Management Science, Vol. 4, p115-35.

63. Markowitz, H. M., (1959), Portfolio Selection: Efficient Diversification of Investments, John Wiley & Sons, New York.

64. McGinnis, L. F., (1977), "A Survey of Recent Results for a Class of Facilities Location Problems", AIIE Transactions, Vol. 9, No. 1, p11-18.

65. McKeown, P., (1975), "A Vertex Ranking Procedure for Solving the Linear Fixed Charge Problem", Operations Research, Vol. 23, p1183- 1191.

66. McKeown, P., (1978), "A Branch and Bound Algorithm for the Linear Fixed Charge Problem", Working Paper, University of Georgia.

67. McKeown, P., (1981), "A Branch-and-Bound Algorithm for Fixed Charge Problems", Naval Research Logistics Quarterly, Vol. 28, No. 4, p607-617.

68. McKeown, P. and P. Sinha (1980), "An Easy Solution for a Special Class of Fixed Charge Problems", Naval Research Logistics Quarterly, Vol. 27, No. 4, p621-624.

69. Montgomery, D. B. and G. L. Urban, (1969), Management Science in Marketing, Prentice-Hall, Englewood Cliffs, N. J.

70. Murty, K. G., (1968), "Solving the Fixed Charge Problem by Ranking the Extreme Points", Operations Research, Vol. 16, p268-279.

71. Nauss, R. M., (1978), "An Improved Algorithm for the Capacitated Facility Location Problem", Journal of the Operational Research Society, Vol. 29, p1145-1201.

72. Nauss, R. M. and R. E. Markland, (1981), "Optimizing Procedure for Lock Box Location Analysis", Management Science, Vol. 27, No. 8, p855-865.

73. Newson, E. F. P., (1975), "Multi-Item Lot Size Scheduling by Heuristic, Part I: with Fixed Resources, Part II: with Variable Resources", Management Science, Vol. 21, No. 10, p1186-1203.

74. Patel, N. R. and M. G. Subrahmanyam, (1982), "A Simple Algorithm for Optimal Portfolio Selection with Fixed Transaction Costs", Management Science, Vol. 21, No. 10, p1186-1203.

75. Price, W. L., A. Martel and K. A. Lewis (1980), "A Review of Mathematical Models in Human Resource Planning", Omega, Vol. 8, No. 6, p639-645.

76. Rapp, Y., (1962), "Planning of Exchange Locations and Boundaries", Ericsson Technics, Vol. 2, p1-22.

77. Ravin[illegible], A., and D. L. Hanline (1980), "Optimal Location of Coal Blending Plants by Mixed Integer Programming", AIIE Transactions, Vol. 12, No. 2, p179-185.

78. Rousseau, Jean-Marc (1973), "A Cutting Plane Method for the Fixed Cost Problem", Phd. Thesis, Massachusetts Institute of Technology.

79. Sa, G., (1969), "Branch and Bound and Approximate Solutions to the Capacitated Plant Location Problem", Operations Research, Vol. 17, p1005-1016.

80. Sharp, W. F., (1970), Portfolio Theory and Capital Markets, McGraw-Hill.

81. Silver, E. A. and H. C. Meal, (1973), "A Heuristic for Selecting Lot-Size Quantities for the Case of a Deterministic Time Varying Demand Rate and Discrete Opportunities for Replacement", Production and Inventory Management, 2nd Quarter, 1973.

82. Steinberg, D. I., (1970), "The Fixed Charge Problem", Naval Research Logistics Quarterly, Vol. 17, No. 2, p217-236.

83. Steinberg, D. I., (1976), "On Solving Large-Scale Fixed Charge Problems", TIMS/ORSA meeting, Miami, November, 1976.

84. Stone, B. K., (1981), "Design of a Receivable Collection System", Management Science, Vol. 27, No. 8, p866-79.

85. Stroup, J., (1967), "Allocation of Launch Vehicles to Space Missions: A Fixed Cost Transportation Problem", Operations Research, Vol. 15, p1157-1163.

86. Taha, H. A., (1973), "Concave Minimization over a Convex Polyhedron", Naval Research Logistics Quarterly, Vol. 20, No. 3, p533-48.

87. Taylor, S. G., (1980), "Optimal Aggregate Production Strategies for Plants with Semifixed Operating Costs", AIIE, Vol. 12, No. 3, p253-257.

88. Teitz and Bart, (1968), "Heuristic Mothds for Estimating the Generalized Vertex Median of a Weighted Graph", Opoerations Research, Vol. 16, No. 5, p155-61.

89. Van Roy, Tony J., (1980), "A Cross Decomposition Algorithm for Capacitated Facility Location", TIMS/ORSA meeting, Washington, 1980.

90. Van Roy, Tony J., and Donald Erlenkotter, (1982), "Dual-Based Procedure for Dynamic Facility Location", Management Science, Vol. 28, No. 10, p1091-1105.

91. Van Wassenhove, L. N. and M. A. De Bodt, (1983), "Capacitated Lot Sizing for Injection Moulding, A Case Study", Journal of the Operational Research Society, Vol. 34, No. 6, p489-501.

92. Wagner, H. M. and T. M. Whitin, (1958), "A Dynamic Version of the Economic Lot Size Model", Management Science, Vol. 5, p89-96.

93. Walker, W. E., (1976), "A Heuristic Adjacent Extreme Point Algorithm for the Fixed Charge Problem", Management Science, Vol. 22, No. 5, p587-596.

94. Walker, W. E., M. Aquilina and D. Schur, (1974) "Development and Use of a Fixed Charge Programming Model for Regional Solid Waste Planning", The Rand Corporation, P-5307.

95. Winters, P. R., (1962), "Constrained Inventory Rules for Production Smoothing", Management Science, Vol. 8, No. 4, p470-81.

96. Zanakis, Stelios H. and James R. Evans, (1981), "Heuristic 'Optimization': Why, When, and How to Use It", Interfaces, Vol. 11, No. 5, p84-90.

97. Zangwill, W., (1969), "A Backlogging Model and Multi-Echelon Model of a Dynamic Lot Size Production System-A Network Approach", Management Science, Vol. 15, No. 9., p506-527.

END

14·11·85

FIN