The installation scheduling problem involves finding a program for installing a large number of sizes and types of items (e.g., machines) over time so as to optimize some measure (e.g., initial capital investment), subject to various resource constraints. Examples of this problem are scheduling the installation of point-of-sale terminals in supermarket and retail chains, and teller terminals in banks.

We have formulated the installation scheduling problem as a mixed integer linear program and developed a computer code for solving the model. By using techniques for exploiting the special structure of the model, our formulation allows rather quick solution times.

Solving the installation scheduling problem using mixed integer linear programming

by R. Chen, H. Crowder, and E. L. Johnson

The installation scheduling problem is often a difficult aspect of capital investment programs. Simply stated, the installation scheduling problem involves finding an orderly schedule for installing a large number of systems of different sizes and types over time so as to optimize some measure (e.g., initial capital investment), subject to various resource constraints. Examples of this problem are scheduling the installation of point-of-sale terminals in supermarkets and retail stores, and teller terminals in banks.

We have formulated the installation scheduling problem as a mixed integer linear program and developed a computer program, the Installation Optimization System (IOS), for solving the model. IOS uses IBM's mixed integer linear programming system MIP/370 as a subroutine. Because of special structure in the model, our formulation allows rather quick solution times, thus avoiding the usual criticism of integer linear programs that they require excessive computational effort.

We will first give a detailed description of the installation scheduling problem. Next we will present our mixed integer programming formulation of the problem. We will then show how the

problem may be solved using IOS. Finally, we will give computational results obtained by applying IOS to an actual installation scheduling problem.

The installation scheduling problem

In general, the installation scheduling problem arises when a decision must be made as to when to perform all or part of a set of actions (typically, money investments) over a given time interval. The time interval is divided into a number of time periods; one time period is the approximate time required to perform an action. The problem has the following characteristics:

- Each action has associated costs and benefits determined by functions. The cost function specifies the amount of capital required to perform the action; this cost is dependent upon the time period in which the action is performed. At a given time period, for those actions that were performed in previous time periods, the benefit function determines the amount of capital that is made currently available. In general, benefit functions will generate nonnegative returns, although this is not a restriction. Both costs and benefits are estimates which take into account such factors as possible future inflation, investment tax credits, tax rates, and depreciation.
- The problem has an associated initial capital investment that is used to start performing actions. At any time period, capital is expended by performing actions, the amount of capital used being determined by the appropriate cost function. At the same time, capital benefits are derived in two ways: (a) any capital not expended at the previous time period is carried forward, and (b) capital benefits arise from previously performed actions, as determined by the appropriate benefit functions.

We can now ask:

1. Is it possible to perform all of the actions in the given time interval, given the initial capital investment?
2. If so, what is an action schedule that will optimize some measure? For example, we might wish to maximize the final capital position at the end of the time interval or, alternately, we might wish to minimize the time required to return the initial capital investment.
3. If 1 is not possible, can we find a schedule for performing part of the set of actions that will optimize some function?

Our interest in the installation scheduling problem resulted from investigating the installation of point-of-sale systems in supermarket chains. In this case, an action is either the conversion of

one store to a point-of-sale system or the installation of a system in a new store. Stores are divided into several types depending primarily upon business volume, and there are many stores of each type. For a given store type, the cost of installation depends upon the time during the year that the installation is made. The benefit at any time derived from a particular store which has previously had a system installed depends, again, upon what time during the year that the original installation was made, the type of store, and upon how long it has been since installation. A typical problem specification would be these cost and benefit functions, the number of stores of each type, the time interval in which total installation of the store chain must be accomplished (usually several years), and the initial capital investment. The objective is to determine an installation schedule in which all stores in the chain have had systems installed, and the final capital position at the end of the time interval is maximized.

Many decision problems of this type can be cast as mixed integer programming problems. A good introduction to these types of models is Wagner.[1]

The mixed integer linear programming model

We will cast the model for the installation problem in terms of the example in the previous section for the installation of point-of-sale systems in supermarkets. Note, however, that the model is applicable to a wide class of similar problems.

Let the constant N be the number of types of stores to be installed and let the constant m_i be the number of stores of type i, with $i = 1, \cdots, N$. The total number of stores to be installed is $\Sigma_i m_i$. The time interval specified for the problem will be divided into T time periods, one period being the approximate time required to install one system.

The cost of installing a system is dependent upon the store type and the time period in which it is installed. Let the constant C_{it} be the cost of installing a system in a store of type i in time period t, with $1 \leq i \leq N$ and $1 \leq t \leq T$.

The capital benefit derived at some time period from a system that has been installed at a previous time period is dependent upon the store type, the time period in which the system was originally installed, and the number of time periods that have elapsed between installation and the current period. Let B_{itr} be the capital available at time period t for a store of type i that had a system installed at time period $t - r$, where $1 \leq i \leq N$, $2 \leq t \leq T$, and $1 \leq r < t$.

The initial capital investment will be denoted by K. We note that K must be at least equal to some C_{it}, else no stores will have systems installed and the model as defined in the sequel will have no feasible solution.

The installation activities x_{it} will denote the number of stores of type i having systems installed in time period t, with $1 \leq i \leq N$ and $1 \leq t \leq T$. For an optimal solution to the model, this set of activities will represent an optimal installation schedule. Fractional parts of x_{it} values indicate that installations are started during one time period and finished during the next period. For example, for a particular i, $x_{it} = 2.75$ and $x_{i,t+1} = 1.25$ means that two installations are started at the beginning of period t and one at the beginning of period $t + 1$. In addition, one installation is started 0.25 through period t (i.e., 0.75 of the work will be done in period t) and is finished in the first quarter of period $t + 1$. We require that the x_{it} be nonnegative.

The activities d_{it} are the number of starts of installations of type i up to and including time period t, for $1 \leq i \leq M$, $1 \leq t \leq T - 1$. We require that d_{it} be nonnegative and integer.

The activity p_t will be the capital position at the *end* of time period t. For $t < T$, p_t is the amount of capital that is carried forward from period t and made available for system installations in period $t + 1$. Clearly p_T is the final capital position at the end of the problem's time interval; this is the quantity that the model attempts to maximize.

Given these specifications, we construct the following mixed integer linear programming formulation:

$$\text{Maximize } p_T \tag{1}$$

subject to

$$\sum_{t=1}^{T} x_{it} = m_i;\ i = 1, \cdots, N \tag{2}$$

$$\sum_{i=1}^{N} C_{i1}x_{i1} + p_1 = K \tag{3}$$

$$\sum_{i=1}^{N} \sum_{r=1}^{t-1} B_{itr}x_{i,t-r} - \sum_{i=1}^{N} C_{it}x_{it} + p_{t-1} - p_t = 0;\ t = 2, \cdots, T \tag{4}$$

$$\sum_{k=1}^{t} x_{ik} - d_{it} \leq 0;\ t = 1, \cdots, T - 1;\ i = 1, \cdots, N \tag{5}$$

$$\sum_{k=1}^{t+1} x_{ik} - d_{it} \geq 0;\ t = 1, \cdots, T - 2;\ i = 1, \cdots, N \tag{6}$$

$$d_{it} \leq m_i;\ i = 1, \cdots, N;\ t = 1, \cdots, T-1 \tag{7}$$

$$\text{all } x_{it} \geq 0;\ \text{all } p_t \geq 0;\ \text{all } d_{it} \geq 0 \text{ and integer} \tag{8}$$

The equations given by 2 require that all stores have systems installed within the specified time interval. Equation 3 specifies that the capital position at the end of the first time period is the initial investment minus the amount expended for installations in that period. The equations represented by 4 specify the capital positions for the end of time periods 2 through T: p_t is the sum of the carry-over from the previous period (p_{t-1}) and the capital derived at period t for all previously installed stores, minus the capital outlay for installations in period t. Relations 5–7 impose restrictions on the values of the scheduling activities x_{it}; in effect, these relations ensure that any system installations initiated in period t are completed in period $t + 1$. To see this, we observe that for some t, Relations 5 and 6 ensure that the value of $x_{i,t+1}$ gives

$$\sum_{k=1}^{t+1} x_{ik} \geq \text{next integer above } \sum_{k=1}^{t} x_{ik}.$$

In general, the fractional part, if any, of

$$\sum_{k=1}^{t} x_{ik}$$

represents the part of an installation begun in time t which must be completed in time $t + 1$.

That is, $x_{i,t+1}$ must be large enough to complete any installation started but not completed in time period t. This device is similar to Beale and Tomlin's SOS2 special ordered sets.[2] SOS2 is a set of variables such that, at most, two adjacent variables may take nonzero values. Our device makes no set restrictions, but rather makes restrictions on the possible values of adjacent variables only.

Our initial formulation of this problem resulted in a model that required all the x_{it} to be integer (the model consisted of Relations 1–4 and all $p_t \geq 0$, all $x_{it} \geq 0$ and integer). This model had two deficiencies; first, because of the integer-valued activities x_{it}, all installations were required to begin and end in the same time period. This was an unrealistic assumption in terms of the real problem being modeled. The second deficiency was the fact that this formulation resulted in a mixed integer program that was very difficult to solve using the branch and bound procedures in MIP/370. Because of the severe restrictions on the integer activities, the problem was very tightly over-constrained, making it very difficult to find feasible integer solutions. For these reasons, we reformulated the relaxed problem given by Relations 1–8 above, obtaining a model that was not only a better reflection of the real problem but was also more amenable to solution by MIP/370.

The installation optimization system

IOS is a PL/I program that uses IBM's mixed integer linear programming system MIP/370 to solve the installation scheduling problem. MIP/370 is a special feature of the IBM Mathematical Programming System Extended/370 (MPSX/370). Under OS/VS, the minimum machine configuration to install, execute, and support IOS is the same as for MPSX/370: the OS/VS minimum real storage requirement and the Universal Instruction Set. For exact storage requirements as a function of problem size, see the MPSX/370 Program Reference Manual.[3]

The functions of IOS can be divided in three categories: matrix generation, problem solution, and report generation.

matrix generation

The problem matrix is generated from the parameters described in the previous section. These data are input to IOS in a predetermined format, where they are transformed to an internal format suitable for processing by MIP/370.

The size of the resulting integer linear program is a function of N and T: the number of logical variables (constraints) is

$$2N(T-1)+T \tag{11}$$

The number of structural variables is

$$N(2T-1)+T \tag{12}$$

of which $N(T-1)$ are integer.

problem solution

IOS uses the branch-and-bound facilities of MIP/370 to solve the integer linear programming model. The interested reader is referred to the MIP/370 Program Reference Manual[4] for a detailed description of MIP/370. A good exposition on branch-and-bound algorithms is Benichou, et al.[5]

Mixed integer programming problems are often very difficult to solve because there is usually no clear indication as to the order in which branching should occur over the integer variable set. As a result, most sophisticated integer programming codes, including MIP/370, have a variety of built-in heuristic procedures for automatically deciding the branching order. For a given problem, some rules may work remarkably well while others may perform very badly. It is usually left to the investigator to make several runs employing different combinations of branching heuristics until the correct combination is found; this can often be a very expensive and frustrating experience. Fortunately our model has a branching order that is simple and straightforward: the integer variables are ordered on the basis of time. For example, for $N = 2$ and $T = 3$, we would order the integer variables d_{it} as

$$\{d_{1,1}, d_{2,1}, d_{1,2}, d_{2,2}\}$$

This ordering, combined with the branch-and-bound algorithm used by MIP/370, leads rather quickly to a fairly respectable (in terms of objective value) first integer solution, thus providing a good bound for, hopefully, expeditious optimization of the problem.

report generation

The output from MIP/370 is usually not especially readable to novices; sometimes it is even inscrutable to experts. For this reason, we have incorporated a report writer into IOS that outputs the solution(s) in a more comprehensible form. As a rule, MIP/370 will find several feasible integer solutions to a problem before it finally finds the optimal integer solution. These solutions are generated in a certain order, each having a better objective function value than the previous one. It is sometimes useful to inspect this series of solutions; for this reason, all solutions are printed by the IOS report writer. The form of the output is given in the next section.

Computational experience

A typical problem solved by IOS was the task of obtaining a schedule for installing point-of-sale systems in a supermarket chain. The actual data to IOS was generated by the Benefit/Investment System (BIS). This system, programmed in APL, is used by IBM to assist in measuring the financial impact of computer systems and allows the quantification of benefits and investments associated with various proposals. The data that we obtained using BIS took into account such factors as possible future inflation, investment tax credits, tax rates, and depreciation.

The supermarket chain had three types of stores: eight of Type 1, 10 of Type 2, and 12 of Type 3. In the notation of the earlier section on the programming model, $N = 3$, $m_1 = 8$, $m_2 = 10$, and $m_3 = 12$. The time period required for installation of one system was three months, and the time interval allowed for installation of the entire chain was nine years; thus $T = 36$.

The cost of installing a system in a store was in the range \$113,143–119,317 for store Type 1, \$125,696–132,233 for Type 2, and \$137,472–144,261 for Type 3. The capital benefit derived from installed stores was in the range \$7,612–14,617 per quarter for store Type 1, \$9,370–19,282 for Type 2, and \$11,129–23,837 for Type 3.

We solved this problem for various values of K, the initial capital investment. Figure 1 is the IOS output for the first (but not optimal) integer solution obtained with K = \$200,000; this solution required 1.00 minute of System/370 Model 168 CPU execution time. The columns are, from left to right, the time period index, the number of store types 1, 2, and 3 to have systems in-

Figure 1 IOS output—first integer solution

```
INSTALLATION OPTIMIZATION SYSTEM ---- EXPERIMENTAL VERSION 3   9/13/77
INITIAL INVESTMENT:   200000.00
```

TIME PERIOD	STORE TYPES 1	2	3	CASH POSITION
1	.	1.00	.	74304.
2	.	.	.	92954.
3	.	.85	.	.
4	.	.15	.	10564.
5	.	.	.	34082.
6	.	.	.	57599.
7	.	.	.	80595.
8	.	.	.	103963.
9	.	.	.94	.
10	.	.	.06	35083.
11	.	.	.	81584.
12	.	.	.	128133.
13	.	.	1.00	30402.
14	.	.	.66	.
15	.	.	.34	27099.
16	.	.	.	107563.
17	.	.	1.32	.
18	.	.	.68	4340.
19	.	.	.83	.
20	.	.	.17	100559.
21	.	.	1.60	.
22	.	.	1.10	.
23	.	.	1.22	.
24	.	.	.08	182166.
25	.	.74	2.00	.
26	.	1.88	.	.
27	.77	1.38	.	.
28	.23	.	.	269533.
29	.44	4.00	.	.
30	3.12	.	.	.
31	2.44	.	.	114595.
32	.	.	.	540434.
33	1.00	.	.	831687.
34	.	.	.	1245557.
35	.	.	.	1662776.
36	.	.	.	2080478.

stalled in the time period, and the accrued capital position at the end of the time period. Fractional values of stores to have systems installed in a time period arise because installation can be initiated in one time period and completed in the following period. Figure 2 gives a time sequence graph of the schedule of Figure 1. The left-hand column is the time period index t. The next three columns give a graphic interpretation of the installation schedule.

Figure 2 Installation schedule—first integer solution

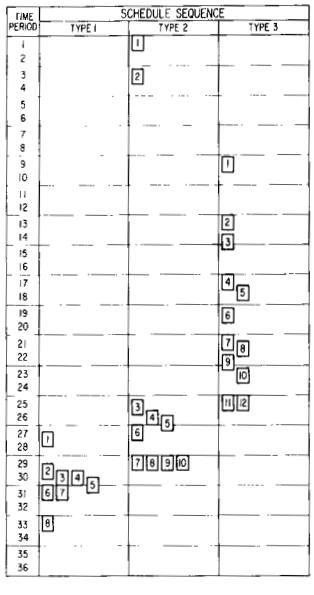

Figure 3 is the IOS output for the optimal integer solution with K = \$200,000, obtained after 3.27 minutes of CPU execution time. Note that the accrued capital position at the end of the last time period is the quantity to be maximized, and has a better value in this solution than in the first integer solution of Figure 1. Figure 4 is a time sequence graph of the optimal integer solution.

We attempted to solve this problem with K = \$150,000. We obtained a first integer solution after 0.72 minute of CPU execution time. However, IOS was unable to prove optimality after 20 minutes of execution time, and the search was halted. This demonstrates the unpredictability of attempting to obtain optimal solutions to mixed integer programs, even with a sophisticated sys-

Figure 3 IOS output—optimal integer solution

```
INSTALLATION OPTIMIZATION SYSTEM ---- EXPERIMENTAL VERSION 3   9/13/77
INITIAL INVESTMENT:   200000.00
```

TIME PERIOD	STORE TYPES 1	2	3	CASH POSITION
1	.	.	1.00	62528.
2	.	.	.	85053.
3	.	.	.	107578.
4	.	.	.	130103.
5	.	.	1.00	6519.
6	.	.	.	42932.
7	.	.	.	79345.
8	.	.	.	115758.
9	.	.	1.00	6835.
10	.	.	.	57909.
11	.	.	.76	.
12	.	.	.24	28966.
13	.	.	.64	.
14	.	.	.36	22763.
15	.	.	.72	.
16	.	.	.28	50526.
17	.	.	1.00	5011.
18	.	.	.85	.
19	.	.	.90	.
20	.	.	.25	106783.
21	.	.	1.80	.
22	.	.10	1.20	.
23	.	1.55	.	.
24	.	.35	.	180605.
25	.	3.17	.	.
26	.	2.13	.	.
27	.	1.70	.	87087.
28	.	.	.	417082.
29	5.37	1.00	.	.
30	2.63	.	.	105759.
31	.	.	.	549640.
32	.	.	.	994540.
33	.	.	.	1413005.
34	.	.	.	1827659.
35	.	.	.	2247908.
36	.	.	.	2669486.

tem like MIP/370. Making seemingly innocuous changes to the input data can sometimes drastically alter the solution times. It is often the case that users of mixed integer programming models never realistically expect to obtain optimal solutions to their problems, but are quite happy to settle for several good feasible integer solutions, provided the linear programming bound assures that no integer solution can be significantly better.

Summary

We have defined the installation scheduling problem, formulated it as a mixed integer linear program, and have shown how to solve it using the Installation Optimization System. We have indicated that the solution techniques embodied in IOS are applicable to wide classes of similar problems. There are extensions of the installation scheduling problem which we have not addressed but which might be interesting to pursue. For example, in many applications a substantial benefit is to be derived when the final action is performed. We have not investigated what effect, if any, such terminal benefits could have on optimal installation schedules.

Another topic that could be significant in certain applications is the question of purchase versus lease of systems. We feel that investigators seeking solutions for these and other extensions of the installation scheduling problem should consider the mixed integer programming approach that we have presented.

ACKNOWLEDGMENT

We wish to thank Michael Held of the IBM Systems Research Institute for helpful discussions in the course of this work.

CITED REFERENCES

1. H. M. Wagner, *Principles of Management Science*, Prentice-Hall, Englewood Cliffs, NJ (1970).
2. E. M. L. Beale and J. A. Tomlin, "Special facilities in a general mathematical programming system for nonconvex problems using ordered sets of variables," *Proceedings of the Fifth International Conference on Operations Research*, Editor: J. Lawrence, 447–454, Tavistock Publications, London (1970).
3. *IBM Mathematical Programming System Extended/370 Program Reference Manual*, SH19-1095, IBM Corporation, Paris, France (1974).
4. *Mixed Integer Programming/370 Program Reference Manual*, SH19-1099, IBM Corporation, Paris, France (1974).
5. M. Benichou, J. M. Gauthier, P. Girodet, G. Hentges, G. Ribiere, and O. Vincent, "Experiments in mixed-integer linear programming," *Mathematical Programming* **1**, 76–94 (1971).

Reprint Order No. G321-5065.

Figure 4 Installation schedule—optimal integer solution