A METHOD TO REDUCE COMPUTER RUNNING TIME

FOR THE INTEGER PROGRAMMING

LOCKBOX PROBLEM

by

Joseph D. Huber Jr.

A thesis submitted to the faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Mathematics).

Golden, Colorado

Date *March 19, 1991* Signed: *Joseph D. Huber Jr.*

Joseph D. Huber Jr.

Approved: *Robert E.D. Woolsey*

Dr. Robert E.D. Woolsey

Golden, Colorado

Date *March 19, 1991*

Ardel Boes

Dr. Ardel J. Boes

Professor and Head,

Mathematics Department

ABSTRACT

One class of integer programming problems is the plant location problem, or lockbox problem. Lockbox problems are unique in that a fixed charge is incurred with each opening of a plant location or lockbox. These problems have historically been solved using branch and bound-based integer programming computer codes. These types of codes, however, frequently require a long time to find the optimum. It has been conjectured that reordering of the objective function could cause reductions in computer running time for this class of problems. This thesis confirms this assertion for pure branch and bound integer programming codes.

We conclude from this study that reordering the objective function coefficients, from smallest to largest, can give reductions in computer running time on pure branch and bound codes of up to 18%. Results on other codes proved to be unpredictable. When the objective function coefficients were rearranged from smallest to largest by region, or row, a greater reduction in computer running time (up to 28%) was obtained on the pure branch and bound code. As before, results on the other codes were unpredictable.

TABLE OF CONTENTS

Page

ABSTRACT iii

LIST OF TABLES vi

ACKNOWLEDGEMENTS vii

Chapter

1. THE LOCKBOX PROBLEM 1

   History of Lockbox Problem Formulation 2
   Standard Lockbox Formulation 3
   Annotated Bibliography 7

2. COMPUTER HARDWARE, SOFTWARE, AND SAMPLE PROBLEMS 12

   Sample Problems 12
   Integer Programming Codes 14
   Computer Platforms 17

3. COMPUTER RUNNING TIME RESULTS 19

   Overview 19
   Objective Function Formulation 20
   Computer Running Time, STORM 2.0 21
   Computer Running Time, LINDO 26
   Computer Running Time, VAX 8600, HS/LP 29
   Computer Running Time, VAX 8600, HS/LP Elevated Priority 35

4. CONCLUSIONS AND SUGGESTIONS FOR FURTHER STUDY . 38

Why Does It Work? 38
Conclusions 41
Suggestions for Further Study 43

REFERENCES CITED 47

SELECTED BIBLIOGRAPHY 52

APPENDIX, SAMPLE PROBLEMS/FORMULATIONS 53

LIST OF TABLES

Table | Page

1. Sample Problems 13

2. Computer Running Time Results, IBM PC/STORM 23

3. Computer Running Time Results, COMPAQ 386s/STORM . . 24

4. Computer Running Time Results, IBM PC/LINDO 27

5. Computer Running Time Results, COMPAQ 386s/LINDO . . 28

6. Computer Running Time Results, CSM VAX 8600, HS/LP . 33

7. Computer Running Time Results, CSM VAX 8600, HS/LP Elevated Priority 36

ACKNOWLEDGEMENTS

I would first like to thank the members of my graduate thesis committee, Dr. Jean Bell, Dr. Ruth A. Maurer, and Dr. Robert E. D. Woolsey.

Professor William R. Astle reviewed the statistical procedure used in Chapter 3. Kata McCarville, the Assistant Director of the CSM Academic Computing Center, was helpful on all matters concerning the VAX 8600 mainframe.

I would like to thank all the members of the "Guild" for the help and assistance they have given me over the past two years. I would especially like to thank fellow graduate student Jim Watson for his continuing advice and assistance.

A special thanks to my wife, Darcee, for without her support and patience this masters degree would not have been possible.

I would also like to thank the U. S. Army for sending me to school, and Dr. R. E. D. Woolsey for giving me an education.

Chapter 1

THE LOCKBOX PROBLEM

One class of integer programming problem is the plant location problem, or lockbox problem. This type of problem is unique in that a fixed charge is associated with opening or operating a location.

Woolsey, writing in Salkin (1975, 512) describes the lockbox problem as follows:

> We assume that a company has a choice of having its accounts receivable sent to a local bank near the accounts receivable. Now, if all the accounts receivable come to the main office, the company will lose interest that it could earn on that money if it were deposited immediately. On the other hand, if the company maintains a bank account near each one of the accounts receivable, the company has to pay each bank a fixed charge for taking care of their business .... The company is thus confronted with the following problem: How many banks should they use, where should they be located, and which accounts receivable should go into a given bank? If we designate banks as "lockboxes" we can formulate this scenario as a zero-one integer programming problem. This problem also represents the simple plant location problem.

Put simply, the company wants to minimize its costs. The company can assign each accounts receivable to only one lockbox. Opening each lockbox involves a fixed charge to the company.

History of Lockbox Problem Formulation

Lockbox problems have been formulated and solved in many mathematical and business publications. Ferdinand K. Levy developed the first heuristic for solving lockbox problems in 1966. Levy used a simple flow chart to reach a satisfactory solution (Levy 1966, B-236). Levy's "Quick and Dirty" algorithm is still being taught in integer programming classes today. However, using Levy's heuristic, "one does not know whether {the} solution represents an optimal (minimal cost) combination or not." (Kraus, Janssen, and McAdams 1970, 52).

The first integer programming formulation of the lockbox problem appeared in the Journal of Bank Research in 1970. Authored by Kraus, Janssen, and McAdams, their integer programming formulation did guarantee optimality (1970, 53). Their integer programming model was a maximization problem, maximizing the savings (due the reduction of mail and clearing times) minus the fixed charge for maintaining the lockboxes.

In 1971 Ciochetto, a graduate student at Colorado School of Mines, used the opposite approach to the lockbox problem. In his Master's thesis "An Application of Binary Integer Programming to the LockBox Problem" Ciochetto's formulation minimized the variable and fixed costs of operating a lockbox, plus the cost of establishing new lock-boxes (Ciochetto 1971, 2). In April 1972 Ciochetto et al presented a working paper on lockbox problems at the 41st National Meeting of the Operations Research Society of America. Subsequently, most journals and textbooks have used a minimization formulation.

Standard Lockbox Formulation

> Simply stated, we wish to minimize the sum of the variable and fixed costs subject to the constraints that each account receivable may be assigned to only one lockbox; and further, if any account receivable is assigned to a lockbox, the fixed charge for use of the lockbox must be assessed (Salkin 1975, 512).

The 'standard' mathematical formulation, incorporating a variety of authors, is:

$$\min \quad z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} f_i y_i$$

$$\text{s.t.} \quad \sum_{i=1}^{m} x_{ij} = 1 \quad (\text{for } j = 1,\ldots,n)$$

$$\sum_{j=1}^{n} x_{ij} - n y_i \le 0 \quad (\text{for } i = 1,\ldots,m)$$

$$x_{ij} = 0, 1 \quad (\text{for all } i, j)$$

$$y_i = 0, 1 \quad (\text{for } i = 1,\ldots,m)$$

where:

m = number of lockboxes

n = number of regions

c_{ij} = variable cost of assigning region j to lockbox i

x_{ij} = 1 if region j sends payments to lockbox i

x_{ij} = 0 if region j does *not* send payments to lockbox i

f_i = fixed charge for using lockbox i

y_i = 1 if lockbox i used

y_i = 0 if lockbox i is *not* used

There are m lockboxes and n regions. This by itself gives us $m * n$ variables. The second constraint above adds another n variables, for a total of $m * n + n$ variables and $m + n$ constraints (Salkin 1975, 513).

There are two ways lockbox problems are presented in the literature. The first, older method is a m by n matrix, with fixed costs given along the right hand side. The second, more recent textbook presentation transposes the matrix by interchanging rows and columns. This puts the fixed costs along the bottom row. An example of each method follows.

A small sample lockbox problem (Salkin 1975, 494) follows in its standard matrix form (m rows and n columns, or i by j). Lockboxes are along the side (m) and regions on the top (n).

Lockbox	Variable Costs: Region 1	Region 2	Region 3	Region 4	Fixed Costs
1	200	200	1050	1200	450
2	100	150	1200	1400	400
3	300	100	750	2000	500

The same sample lockbox problem, transposed, is:

Costs	Region	Lockbox 1	Lockbox 2	Lockbox 3
Variable	1	200	100	300
	2	200	150	100
	3	1050	1200	750
	4	1200	1400	2000
Fixed		450	400	500

This second, more recent approach to lockbox problems simply transposes the matrix. The old first row is now the new first column. This transposed matrix is now n by m (or j by i). Regions are now along the side (n) and lockboxes on the top (m). Fixed costs are visible along the bottom row. This second approach, while confusing at first, is another way lockbox problems are presented in the literature and the Appendix.

The standard mathematical formulation of either lockbox problem is the same:

$$\begin{aligned}
\min z = {} & 200x_{11} + 200x_{12} + 1050x_{13} + 1200x_{14} + 100x_{21} + 150x_{22} \\
& + 1200x_{23} + 1400x_{24} + 300x_{31} + 100x_{32} + 750x_{33} + 2000x_{34} \\
& + 450y_1 + 400y_2 + 500y_3 \\
\text{s.t.} \quad & x_{11} + x_{21} + x_{31} = 1 \\
& x_{12} + x_{22} + x_{32} = 1 \\
& x_{13} + x_{23} + x_{33} = 1 \\
& x_{14} + x_{24} + x_{34} = 1 \\
& x_{11} + x_{12} + x_{13} + x_{14} - 4y_1 \leq 0 \\
& x_{21} + x_{22} + x_{23} + x_{24} - 4y_2 \leq 0 \\
& x_{31} + x_{32} + x_{33} + x_{34} - 4y_3 \leq 0
\end{aligned}$$

The sample problem has 3 possible lockbox locations (m). There are 4 regions (n). It follows that there are $m * n + m$ ($3 * 4 + 4 = 16$) variables and $m + n$ ($3 + 4 = 7$) constraints.

The optimal solution to this sample problem is

$$z^* = 3{,}100$$

$$x_{11} = 1,\ x_{12} = 1,\ x_{13} = 1,\ x_{14} = 1$$

$$y_1 = 1$$

all other x_{ij}, $y_i = 0$

Interpreting this result, the minimum cost (z^*) is \$3,100, and lockbox 1 (y_1) should be used for regions 1, 2, 3, and 4 (x_{11}, x_{12}, x_{13}, and x_{14}).

The explicit search for this optimal zero-one solution would have 2^{mn+m} combinations, or 2^{15} possible binary solutions. Using an IBM PC running STORM, this small problem took 36 seconds (see Chapter 3). For large problems the search for the optimal zero-one solution can take a long time.

The run time required to solve lockbox problems is the main focus of this research. Twelve sample problems were run using three different objective function configurations, three different computer platforms (from personal computers to a mainframe), and using three different integer programming codes. Comparisons were then made. The objective was to test Woolsey's hypothesis presented below.

Annotated Bibliography

Formulating integer programming problems is one thing; solving integer programming problems in a reasonable amount of time is another. During the literature search it appeared there are two methods used to solve integer programming problems in a reasonable amount of time. The first, older method involves reordering the objective function coefficients to decrease the computer running time of the existing problem. The second, more recent technique reduces the size of the problem to be solved.

The importance of ordering objective function coefficients while formulating integer programming problems first appeared in 1974 when Stanley Zionts produced the first edition of Linear and Integer Programming. Chapter 17 of Zionts' book was written by Dr. R. E. D. Woolsey from the Colorado School of Mines. Woolsey's chapter was entitled "Some Practical Aspects of Solving Integer Programming Problems." It was in Zionts' book that Woolsey first introduced his concept of ordering objective function coefficients when formulating integer programming problems.

> The ordering of variables in branch and bound methods can be particularly important, because many of the methods branch on the "next" free variable that is required to be an integer. Therefore, zero-one variables should appear, from left to right, in order of increasing costs for a minimizing problem .... The positioning of variables is undoubtedly important for algorithms that branch on the "next" integer ... (Zionts 1974, 481).

One year later Harvey Salkin published his book, Integer Programming. Woolsey wrote a chapter in Salkins' 1975 book called "How to do Integer Programming in the Real World." In his section about formulating lockbox problems Woolsey wrote in more detail about ordering objective function coefficients:

> If we formulated the ... problem so that {the coefficients} are in increasing order from left to right, it would be simpler to view the combinations of solutions. If the implicit enumeration code we applied to the problem considered fixing variables from left to right also, it would tend to first find a binary solution by assigning every customer to the bank with the smallest total of fixed and variable costs. That is, it would assign *first* all the customers to the bank with the leftmost fixed charge. It would then move rightward through the higher fixed charges if and only if there existed some combination of fixed and variable charges cheaper than the present solution. In short, we have forced the early solutions to be "good" and then the remaining possibilities should be implicitly enumerated quickly. This is a basic rule whenever we are confronted with the problem of solving an integer programming problem on a enumerative code. Simply put, there is little question that running time will be reduced if we can start off with the best possible bound on the solution (Salkin 1975, 513).

In general terms, for a minimization problem, Woolsey rearranges the objective function coefficients from smallest to largest values. This preprocessed problem is then fed to an integer programming computer code. Woolsey stated that the computer will start with a "good" starting solution, and running time will be demonstrably reduced. However, Woolsey did not show any computational results.

The second method of solving integer programming problem reduces the size of the problem to be solved. There are three articles concerning methods of simplifying and reducing integer programming problems. Two articles on solving large scale zero-one problems appeared in <u>Operations Research</u> in

1983 and 1985, respectively. Both of these articles were recommended by Camm, Raturi, and Tsubakitani in their 1990 Interfaces article, "Cutting Big M Down to Size."

> Two excellent, readable, articles on how to preprocess integer programs are Crowder, Johnson, and Padberg {Crowder 1983, 803} and Johnson, Kostreva, and Suhl {Johnson 1985, 803}. In addition to discussing the fixing of variables in more detail, each article also discusses coefficient reduction, a technique which tightens constraints to force the feasible region of the linear program to match that of the integer program more closely (Camm 1990, 64).

The third article, published in 1983, appeared in the Lecture Notes in Economics and Mathematical Systems. Contributing to a special volume entitled "Redundancy in Mathematical Programming" H. Paul Williams wrote Chapter 9, "A Reduction Procedure for Linear and Integer Programming Models." Williams prefaces his chapter as follows:

> A procedure is described for simplifying linear and integer programming models. The procedure preforms test which may:
>
> i. Detect infeasibility or weakness;
> ii. Detect and remove weakly and strongly redundant constraints;
> iii. Detect strongly binding constraints and remove them by a suitable adjustment of the objective function;
> iv. Fix variable and remove them;
> v. Replace constraints by simple bounds;
> vi. Replace columns by bounds on shadow prices;
> vii. Tighten (or relax) bounds on variables and shadow prices.

> An intuitive explanation is given of all the tests. This is followed by a detailed flow chart of the procedure (Karwan et al. 1983, 87).

These last three articles describe different ways to reduce the size of the integer programming problem to be solved. None of these methods address reordering the objective function coefficients to reduce computer running time.

Summarizing, there are two references in the field of integer programming, both by Woolsey, that concern reordering objective function coefficients to reduce computer running time. And, as stated earlier, Woolsey did not show any computational results for his work.

Chapter 2

COMPUTER HARDWARE, SOFTWARE, AND SAMPLE PROBLEMS

Sample Problems

The literature search yielded twelve sample lockbox problems. Much of the literature refers to particular sample problems the authors solved, but most authors do not reproduce or reference these sample lockbox problems in their articles.

The twelve published sample problems are referenced and reproduced in the Appendix to this thesis, including their mathematical formulation. These sample problems range from 15 variables and 7 constraints to a matrix that was 115 by 27.

The sample problem number, reference, size, and optimal solutions are presented in Table 1

Table 1. Sample Problems

Problem Number	Reference, Page	Size (m by n)	Optimal Solutions: Lockbox	Optimal Solutions: Cost
1	Hesse, 268	33 by 13	1	$z^* =$ 276
2	Hesse, 269	44 by 14	1,2	$z^* =$ 610
3	Hesse, 270	85 by 21	2,4	$z^* =$ 410
4	Hesse, 271	65 by 17	1,5	$z^* =$ 630
5	Hesse, 271	115 by 27	4,5	$z^* =$ 933
6	Hesse, 272	55 by 15	3,4,5	$z^* =$ 232
7	Hesse, 279	84 by 19	3,4,6	$z^* =$ 244
8	Winston, 372	20 by 8	1,3	$z^* =$ 242
9	Erlenkotter, 1000	45 by 13	4,5	$z^* =$ 1,235
10	Salkin, 492	24 by 9	2,4	$z^* =$ 1,490
11	Salkin, 494	15 by 7	1	$z^* =$ 3,100
12	Salkin, 495	20 by 8	4	$z^* =$ 920

All twelve problems were run on three different computer platforms (from personal computers to a mainframe), using three different integer programming codes.

Integer Programming Codes

Three different integer programming codes were used to solve the sample problems: STORM, LINDO, and HS/LP. STORM and HS/LP use different branch and bound algorithms. LINDO uses a variable and set based branching algorithm of its own design.

STORM. STORM Personal Version 2.0, Quantitative Modeling for Decision Support is published by Holden-Day, Inc. STORM is an integrated software package consisting of the most frequently used quantitative modeling techniques for business and engineering problems (Emmons 1989, 1).

STORM's Integer Programming module uses the branch and bound algorithm, with running time controllable by prioritizing variables, limiting the number of nodes searched, and acceptance of a suboptimal solution (Emmons 1989, 82). The maximum problem size for the Personal Version of STORM 2.0 is limited to 50 variables and 40 constraints (Emmons 1989, 82).

STORM runs on any IBM-compatible personal computer, is easy to use, and is definitely user-friendly.

LINDO. LINDO, now in its fourth edition, is produced by LINDO Systems, Inc., of Chicago, Ill. LINDO (Linear, INteractive, and Discrete Optimizer) is an interactive linear, quadratic, and integer programming system which is designed to be useful to a wide range of users (Schrage 1989, 1).

LINDO uses a variable and set based branching algorithm of its own design.

> The selection of the variable upon which to branch is biased towards variables which:
> 1. have a fractional part close to 0.5,
> 2. appear early in the problem,
> 3. cause a large change in the objective function value if their values are changed.
>
> Superimposed upon this variable based branching, LINDO may use a set based branching. Suppose that in some IP, variable X3, X7, and X8 have value zero when solved as an LP. If the reduced costs of these variables are large (suggesting that these variables should remain at zero), LINDO may create two new subproblems by alternately adding the Martin (due to Kipp Martin) cuts: X3 + X7 + X8 = 0 or X3 + X7 + X8 ≥ 1 (Schrage 1989, 38).

LINDO's branching algorithm is unique to LINDO, and which algorithm LINDO is using is not transparent to the user.

Student LINDO/PC allows for 60 constraint rows and a maximum variable limit of 120 variables (100 of which may be integer) (Schrage 1989, 1).

LINDO runs on personal computers and mainframes, and is simple to use; after beginning the program you merely type in your formulated problem. Once again, LINDO's branching algorithm is unique to LINDO.

HS/LP. Haverly Systems' Linear Programming System is a system to solve large linear programming models. With the optional auxiliary mixed integer module, HS/LP can handle mixed integer LP problems and binary (0-1) variables.

HS/LP's underlying solution method is branch and bound (Haverly 1985, C-1), but not a pure branch and bound as found in STORM. HS/LP optimizes each node explored using a parametric algorithm. When it is determined that a node cannot lead to an integer feasible solution better than the current best, it is "cut off" and will not be explored further (Haverly 1985, 2-20). HS/LP also uses a penalty estimation which allows the branch to be cut off at the earliest possible opportunity (Haverly 1985, C-1).

Haverly Systems has made HS/LP available to Colorado School of Mines for educational use. HS/LP is on the CSM VAX 8600 Mainframe, running under VMS V5.2. For a computer with word length of 32 bits, as is the CSM VAX, the maximum problem size is 32,000 variables and 8,192 rows (Haverly 1985, I-1).

HS/LP is not user-friendly; properly formatting and entering a problem is painstaking. Nevertheless, HS/LP is one of the few programs capable of solving large linear and integer programming models.

Computer Platforms

Three dissimilar computer platforms were used to run the three integer programming codes; an IBM PC, a COMPAQ 386s PC, and the Colorado School of Mines VAX 8600 mainframe.

IBM PC. The IBM PC used for running STORM and LINDO is located in Room 102 Stratton Hall. This original (1982) IBM PC has been slightly modified. It has a NEC V20 main processor and an Intel 8087 coprocessor. STORM Version 2.0 will automatically use a math coprocessor (Emmons 1989, iii); LINDO (Fourth Edition) does not (Schrage 1989, 1).

Based on the Norton Utilities System Information performance tests this IBM PC has a Computing Index (CI) of 1.8, a Disk Index (DI) of 2.6, and an overall Performance Index (PI) of 2.0. These test results are reported as indexes relative to the (8088-based) IBM PC-XT (with the XT as 1.0) (Norton 1988, 190).

COMPAQ 386s PC. A COMPAQ 386s was also used for running the Personal Version of STORM 2.0 and LINDO (Fourth Edition). It has an Intel 386s main processor, and no coprocessor. As stated above, STORM will use a math coprocessor (Emmons 1989, iii); LINDO does not (Schrage 1989, 1).

The COMPAQ 386s has a Norton CI of 15.6, DI of 3.4, and PI of 11.5, relative to an IBM PC-XT.

VAX 8600 Mainframe. Norton Utilities does not make a System Information performance test for mainframe computers. Measuring mainframe performance is not as straightforward as performance tests on stand-alone personal computers.

One measure of computer performance is processor speed, measured in Millions of Instructions Per Second (MIPS). The VAX 8600 processor provides up to 4.2 times the performance of the VAX-11/780 (Digital 1985, 1-23). The VAX-11/780 processes up to one million VAX instructions per second, equivalent to 1 MIP (Kata McCarville, 18 February 1991). It follows that the VAX 8600 has a processing speed of 4.2 MIPS.

The Colorado School of Mines VAX 8600 runs under the VMS Version 5.2 operating system.

Twelve sample problems were run on three different computer platforms using the three previously-described integer programming codes. The time required to run each problem was recorded. Chapter 3 will demonstrate that the computer time required to solve integer programming problems can be noticeably reduced by simply rearranging the objective function coefficients.

Chapter 3

COMPUTER RUNNING TIME RESULTS

Overview

Using STORM's branch and bound algorithm, and running on a personal computer, Chapter Three documents a reduction in computer running time ranging from 0% to 18% on twelve sample problems by rearranging the objective function coefficients, from smallest to largest. LINDO, also running on a personal computer, did not respond to a reordering of the objective function coefficients. The third integer programming code, HS/LP, gave unpredictable results. Using HS/LP computer running time ranged from -13% to +13% when the objective function coefficients were arranged from smallest to largest.

If the objective function coefficients are rearranged from smallest to largest by region, or row, this thesis discovered a greater reduction in computer running time using STORM (from 0% up to 28%). Using LINDO and rearranging the objective function by region gave the same results as before; there was not a uniform decrease in computer running time. For some problems running time increased. HS/LP also gave the same unpredictable results, ranging from -13% to +20% time savings when the objective function coefficients were reordered from smallest to largest by region, or row.

Objective Function Formulation

Each of the twelve sample problems were run using three different objective function formulations. The first formulation is the standard mathematical formulation presented in Chapter 1 and listed in the Appendix. The second method was first proposed by Woolsey in 1974 (Zionts 1974, 481). This method rearranges the objective function coefficients from smallest to largest variable costs (x_{ij}) and fixed costs (y_i). The third formulation, and the real topic of this thesis, is the run-time experiment that rearranges the objective function from smallest to largest x_{ij} and y_i by region, or row.

Standard Formulation. A small sample problem (Salkin 1975, 494) was introduced in Chapter 1. The objective function of the original lockbox problem was:

$$\begin{aligned}\min z = {} & 200x_{11} + 200x_{12} + 1050x_{13} + 1200x_{14} + 100x_{21} + 150x_{22} \\ & + 1200x_{23} + 1400x_{24} + 300x_{31} + 100x_{32} + 750x_{33} + 2000x_{34} \\ & + 450y_1 + 400y_2 + 500y_3\end{aligned}$$

Smallest to Largest x_{ij} y_i Values. Rearranging the objective function coefficients, from the smallest to largest x_{ij} (variable costs) and y_i (fixed costs), the objective function is:

$$\min z = 100x_{21} + 100x_{32} + 150x_{22} + 200x_{11} + 200x_{12} + 300x_{31} + 750x_{33} + 1050x_{13} + 1200x_{14} + 1200x_{23} + 1400x_{24} + 2000x_{34} + 400y_2 + 450y_1 + 500y_3$$

Smallest to Largest x_{ij} y_i Values, By Region. This method rearranges the objective function from smallest to largest x_{ij} and y_i by region, or row. Explaining further, the variable costs for region one lockboxes are arranged from smallest to largest, then the variable costs for region two are arranged from smallest to largest, etc., and finally the fixed costs for each lockbox are arranged from smallest to largest.

Rearranging our small sample problem from smallest to largest variable costs (x_{ij}) and fixed costs (y_i), by region, the objective function is now:

$$\min z = 100x_{21} + 200x_{11} + 300x_{31} + 100x_{32} + 150x_{22} + 200x_{12} + 750x_{33} + 1050x_{13} + 1200x_{23} + 1200x_{14} + 1400x_{24} + 2000x_{34} + 400y_2 + 450y_1 + 500y_3$$

Computer Running Time, STORM 2.0

As stated in Chapter 2, a slightly modified IBM PC was used for running STORM Version 2.0. The computer running times listed in this Chapter reflect the twelve sample problems, each with the three different objective function formulations described above.

Measuring computer running time using STORM required writing a batch file and using a program animator. STORM is a menu driven, integrated software package (Emmons 1989, 1). As such, it cannot be run, by itself, from within a batch file to determine computer running time. Instead, a PC Magazine utility named PAN (Program ANimator) was used to automate each sample run (Maclean 1990, 253).

A standard batch file was used to reset the internal computer clock prior to each sample run. This was done using the normal MS-DOS TIME command (Wolverton 1988, 455). PAN was then called from within the batch file, to run each sample problem within STORM the same way each time. PAN started STORM, loaded a sample problem, ran the sample problem, and then exited STORM. Finally, the batch file displayed the elapsed computer running time.

Table 2 shows the computer running times and percentage savings compared to the standard formulation for an IBM PC running STORM 2.0.

Table 2. Computer Running Time Results

Computer: IBM PC

Software: STORM 2.0

Running Time (in seconds)

Problem	Standard Formulation	Small to Large $x_{ij}y_i$	%	Small to Large $x_{ij}y_i$ By Region	%
1	60	56	6.7%	54	10.0%
2	417	342	18.0%	330	20.7%
3	N/A	N/A		N/A	
4	N/A	N/A		N/A	
5	N/A	N/A		N/A	
6	N/A	N/A		N/A	
7	N/A	N/A		N/A	
8	87	77	11.5%	76	12.6%
9	621	509	18.0%	450	27.5%
10	74	66	10.8%	66	10.8%
11	36	36	0.0%	35	2.8%
12	11	10	9.1%	10	9.1%

N/A: Not Available. The maximum problem size (variables by constraints) is 50 by 40 in the Personal Version of STORM (Emmons 1989, 82).

As stated in Chapter 2, a COMPAQ 386s was also used for running STORM Version 2.0. The computer running time using this faster computer for the same twelve sample problems are shown in Table 3.

Table 3. Computer Running Time Results
Computer: COMPAQ 386s
Software: STORM 2.0

	Running Time (In Seconds)				
Problem	Standard Formulation	Small to Large $x_{ij}y_i$	%	Small to Large $x_{ij}y_i$ By Region	%
1	9	9	0.0%	8	11.1%
2	59	49	17.0%	47	20.3%
3	N/A	N/A		N/A	
4	N/A	N/A		N/A	
5	N/A	N/A		N/A	
6	N/A	N/A		N/A	
7	N/A	N/A		N/A	
8	14	12	14.3%	12	14.3%
9	93	77	17.2%	67	28.0%
10	11	10	9.1%	10	9.1%
11	6	6	0.0%	6	0.0%
12	2	2	0.0%	2	0.0%

N/A: Not Available. The maximum problem size (variables by constraints) is 50 by 40 in the Personal Version of STORM (Emmons 1989, 82).

Using an IBM PC or a COMPAQ 386s both Table 2 and Table 3 reflect consistent results. That is, the conclusions below are machine-independent.

Both Tables show that, using STORM's branch and bound algorithm, computer running time may be reduced when the objective function coefficients are reordered. As shown in Table 2 and 3, the reduction in computer running time ranges from 0% up to a 18% by rearranging the objective function coefficients from smallest to largest. A greater reduction in computer running time when using STORM (from 0% up to 28%) can be achieved if the objective function coefficients are reordered from smallest to largest by region

Larger integer programming problems showed the greatest reduction in computer running time. Smaller integer programming problems demonstrated little, if any, reduction. Why is there a difference?

One explanation may come from an analysis of how branch and bound works. Most branch and bound codes solve the embedded linear programming problem first. Then, if required, the linear relaxation subproblems of the integer programming problem are solved (the actual branching and bounding). If the objective function coefficients are reordered from smallest to largest branch and bound will look first at the terms that are contributing the least to the objective function. Explaining the difference in computer running time then becomes a "chicken or the egg" problem. Are the larger problems solved faster because the LP runs faster or because branching is quicker after the LP is solved? This may also imply that linear programming problems would benefit from reordering the objective function coefficients. Further research is needed to support this conjecture.

Computer running time was also affected by the range of the objective function coefficients. For problems of similar size, if the objective function coefficients were relatively close together the problem took longer to solve. Those sample problems with a larger range of objective function coefficients were solved more quickly.

Computer Running Time, LINDO (Fourth Edition)

LINDO (Fourth Edition) was run on the same two IBM-compatible personal computers as STORM 2.0. Computer running time was measured directly for LINDO. LINDO can be run in a batch mode (Schrage 1989, 28), and a batch file (TIMER.BAT) was used to determine computer running time (Wolverton 1986, 86). The computer running time, in seconds, as well as the percentage savings compared to the standard formulation, are shown in Table 4. A negative percentage indicates that the computer running time increased, or got worse.

Table 4. Computer Running Time Results

Computer: IBM PC

Software: LINDO (Fourth Edition)

Running Time (In Seconds)

Problem	Standard Formulation	Small to Large $x_{ij}y_i$	%	Small to Large $x_{ij}y_i$ By Region	%
1	96	93	3.1%	96	0.0%
2	198	205	-3.5%	206	-4.0%
3	482	497	-3.1%	495	-2.7%
4	213	214	-0.5%	209	1.9%
5	N/A	N/A		N/A	
6	703	590	16.1%	628	10.7%
7	1,160	894	22.9%	972	16.2%
8	74	74	0.0%	74	0.0%
9	181	180	0.6%	180	0.6%
10	48	47	2.1%	47	2.1%
11	55	55	0.0%	55	0.0%
12	22	22	0.0%	22	0.0%

N/A: Not Available. The maximum problem size of Student LINDO/PC is 60 rows and 120 variables. Slack variables count against the maximum variable limit (Schrage 1989, 1)

Using LINDO on a COMPAQ 386s the computer running times for the twelve sample problems and three different objective function formulations are shown in Table 5.

Table 5. Computer Running Time Results
Computer: COMPAQ 386s
Software: LINDO (Fourth Edition)

Running Time (In Seconds)

Problem	Standard Formulation	Small to Large $x_{ij}y_i$	%	Small to Large $x_{ij}y_i$ By Region	%
1	13	12	7.7%	13	0.0%
2	26	27	-3.9%	27	-3.9%
3	73	75	-2.7%	75	-2.7%
4	28	28	0.0%	27	3.6%
5	N/A	N/A		N/A	
6	70	77	-10.0%	82	-17.1%
7	151	117	22.5%	127	15.9%
8	10	10	0.0%	10	0.0%
9	24	23	4.2%	23	4.2%
10	6	6	0.0%	6	0.0%
11	7	7	0.0%	7	0.0%
12	3	3	0.0%	3	0.0%

N/A: Not Available. The maximum problem size of Student LINDO/PC is 60 rows and 120 variables. Slack variables count against the maximum variable limit (Schrage 1989, 1)

Both Table 4 and Table 5 again reflect consistent computer running times, whether on a slower IBM PC or faster COMPAQ 386s. When using LINDO on an IBM PC or COMPAQ 386s there is not a uniform decrease in computer running time when the objective function coefficients are reordered. For some problems the running time is increased.

As stated in Chapter 2, LINDO uses a variable and set based branching algorithm of its own design. LINDO's branching algorithm is unique to LINDO. Which algorithm LINDO is using not visible to, and not controlled by, the user.

LINDO's branching algorithm makes reordering the objective function coefficients irrelevant.

Computer Running Time, VAX 8600, HS/LP Version 1.57.A

Haverly Systems' Linear Programming System (HS/LP) was run on the Colorado School of Mines' VAX 8600 mainframe. HS/LP is typically used to solve large linear and integer programming models. HS/LP's underlying solution method is a modified branch and bound (Haverly 1985, C-1).

In contrast to personal computers used to run the three previous software packages, mainframes are multi-user, multi-task, systems. Precisely measuring computer running time on a multi-user system is difficult. One reason for this difficulty is a mainframe's memory management.

> The virtual memory management carried out by the VMS operating system allows many processes to share the physical memory of the machine. This is done by moving portions of user programs and data in and out of physical memory (to disk) under the control of the operating system.
>
> The total amount of physical memory allocated to a process is limited, but can float between an upper bound and a lower bound based on system load. Hence, the movement of programs and data between the physical memory and disk may be increased or decreased, based on system load.
>
> Some of the CPU time required for the memory management activities is charged to the process being managed. Some is charged against system processes (PAGER and SWAPPER) (Kata McCarville, Assistant Director of the CSM Academic Computing Center, February 1, 1991).

In other words, the Central Processor Unit (CPU) time will vary from run to run, based on a myriad of factors not controlled by the user. Confidence intervals, a statistical analysis technique, were employed to estimate computer running time on this multi-user system, the Colorado School of Mines' VAX 8600 mainframe.

Confidence intervals are used to estimate the average, or mean, computer running time.

> Frequently, we wish to know how large a sample is necessary to ensure that the error in estimating μ {the mean} will be less than a specified {error} amount *e*. (Walpole and Myers 1989, 239).

The required sample size was calculated using the following formula from Walpole and Myers (1989, 239).

$$n = \left(\frac{z_{\alpha/2}\ \sigma}{e}\right)^2$$

where $1 - \alpha$ = desired confidence interval

n = required sample size

$z_{\alpha/2}$ = z value at $\alpha / 2$

σ = population standard deviation

e = specified amount of error

For purposes of estimating the mean (μ) CPU running time a 95% confidence interval ($z_{\alpha/2} = 1.96$) was selected with a specified amount of error (e) of .1. The population standard deviation (σ) came from a preliminary sample of size $n = 30$ to provide an estimate of σ (Walpole and Myers 1989, 239).

Solving for n above identifies how large a random sample size is required to be 95% confident that the estimate of μ (the mean CPU running time) is off by less than .1 CPU seconds. In some cases the required sample size was in excess of 75 mainframe problem runs. The minimum n used for this thesis was 30. A narrower confidence interval or smaller specified amount of error would increase the required sample size; a wider confidence interval or larger specified amount of error would decrease it.

Two other techniques were applied to the mainframe problem runs. A batch file was set up to run each problem's three different objective function formulations in a repeating, alternating series. This reduced any multi-user or multi-task variation from run to run. Further, the number of runs was based upon the largest required sample size, calculated above, for each problem's three objective function formulation. Said another way, the one objective function with the largest standard deviation was used to determine the number of runs for all three objective function formulations.

As in the four personal computer runs, the mainframe multi-user runs included twelve sample problems, and each of these twelve problems consisted of three different objective function formulations. The sample mean ($\bar{x}$) is used to approximate the population mean (μ) running times. The sample mean and standard deviation for the VAX 8600, running HS/LP, are shown in Table 6.

Table 6. Computer Running Time Results
Computer: CSM VAX 8600 Mainframe, VMS V5.2
Software: HS/LP System Version 1.57.A

	Running Time (In CPU Seconds)				
Problem	Standard Formulation	Small to Large $x_{ij}y_i$	%	Small to Large $x_{ij}y_i$ By Region	%
1	$\bar{x}$ = 2.387	$\bar{x}$ = 2.098	12.1%	$\bar{x}$ = 1.920	19.6%
	s = .126	s = .108		s = .081	
2	$\bar{x}$ = 3.215	$\bar{x}$ = 3.084	4.1%	$\bar{x}$ = 3.606	-12.2%
	s = .138	s = .111		s = .182	
3	$\bar{x}$ = 4.381	$\bar{x}$ = 3.857	12.0%	$\bar{x}$ = 4.294	2.0%
	s = .269	s = .267		s = .313	
4	$\bar{x}$ = 3.447	$\bar{x}$ = 3.103	10.0%	$\bar{x}$ = 3.346	2.9%
	s = .350	s = .316		s = .296	
5	$\bar{x}$ = 6.389	$\bar{x}$ = 5.794	9.3%	$\bar{x}$ = 6.747	-5.6%
	s = .411	s = .408		s = .444	
6	$\bar{x}$ = 3.192	$\bar{x}$ = 3.228	-1.1%	$\bar{x}$ = 4.809	-5.1%
	s = .145	s = .114		s = .042	
7	$\bar{x}$ = 4.466	$\bar{x}$ = 3.871	13.3%	$\bar{x}$ = 4.296	3.8%
	s = .258	s = .289		s = .303	
8	$\bar{x}$ = 2.528	$\bar{x}$ = 2.537	-0.4%	$\bar{x}$ = 2.469	2.3%
	s = .096	s = .096		s = .093	
9	$\bar{x}$ = 4.110	$\bar{x}$ = 4.035	1.8%	$\bar{x}$ = 4.184	-1.8%
	s = .193	s = .187		s = .199	
10	$\bar{x}$ = 1.433	$\bar{x}$ = 1.489	-3.9%	$\bar{x}$ = 1.490	-0.4%
	s = .174	s = .134		s = .159	
11	$\bar{x}$ = 2.001	$\bar{x}$ = 1.989	0.1%	$\bar{x}$ = 1.801	10.0%
	s = .120	s = .115		s = .091	
12	$\bar{x}$ = 1.330	$\bar{x}$ = 1.500	-12.8%	$\bar{x}$ = 1.330	0.0%
	s = .088	s = .071		s = .093	

<u>Note</u>: $\bar{x}$ = sample mean
s = sample standard deviation

When the objective function coefficients were arranged from smallest to largest, computer running time ranged from -13% to +13%. This range appears to be due to the underlying solution method of Haverly System's HS/LP.

As stated earlier, HS/LP uses a modified branch and bound algorithm (Haverly 1985, C-1). When HS/LP determines that a node cannot lead to an integer feasible solution better than the current best, it is "cut off" and will not be explored further (Haverly 1985, 2-20). This implies that HS/LP, while not using a pure branch and bound algorithm, may benefit from reordering of the objective function coefficients from smallest to largest, as Woolsey suggests. When HS/LP begins with a good starting solution computer running time should be reduced. Table 6 generally reflects this conclusion.

Table 6 also showed that larger problems exhibited the greatest reduction in computer running time when the objective function is arranged from smallest to largest. The smaller problems demonstrated little, if any reduction. This is consistent with the STORM results presented in Tables 2 and 3. STORM uses pure branch and bound; HS/LP uses a modified branch and bound.

When the objective function coefficients were arranged from smallest to largest by region, or row, HS/LP exhibited uneven improvement, ranging from -13% to +20%. By reordering the objective function coefficients from smallest to largest by region the computer should begin with a "better" starting solution. This reordering is based upon the first set of constraints that HS/LP encounters (the equality constraints) and running time should be reduced.

In short, Haverly System's HS/LP does not use a pure branch and bound algorithm, which appears to account for the inconsistent response to reordering of the objective function coefficients. Large problems run on HS/LP may benefit from reordering the objective function coefficients.

Computer Running Time, VAX 8600, HS/LP Version 1.57.A, Elevated Priority

Through a special arrangement with Derek Wilson and Kata McCarville of the Colorado School of Mines Academic Computing Center an elevated priority run was made on the CSM VAX 8600. The purpose of this elevated priority run was to validate the previous use of confidence intervals.

A batch file containing each of the twelve problems' three different objective function formulations was run on February 1, 1991. The results of this elevated priority run are shown in Table 7.

Table 7. Computer Running Time Results, Elevated Priority

Computer: CSM VAX 8600 Mainframe, VMS V5.2

Software: HS/LP System Version 1.57.A

	Running Time (In CPU Seconds)				
Problem	Standard Formulation	Small to Large $x_{ij}y_i$	%	Small to Large $x_{ij}y_i$ By Region	%
1	2.45	2.09	14.7%	2.01	18.0%
2	3.24	3.23	0.3%	3.57	-10.2%
3	5.90	5.09	13.7%	5.63	4.6%
4	3.87	3.36	13.2%	3.44	11.1%
5	7.67	6.47	15.7%	7.90	-3.0%
6	3.41	3.13	8.2%	3.30	3.2%
7	5.63	4.93	12.4%	5.67	-0.7%
8	2.56	2.49	2.7%	2.26	11.7%
9	4.32	4.59	-6.3%	4.29	0.7%
10	1.63	1.58	3.1%	1.75	-7.4%
11	2.07	1.85	10.6%	1.73	16.4%
12	1.37	1.48	-8.0%	1.28	6.6%

The actual numbers recorded above on Table 7 are not important. What is important is their relationship and magnitude. These values are generally consistent (within one standard deviation) to the values obtained by multiple multi-user computer runs, supporting the use of confidence intervals on a multi-user, multi-task system.

The results and conclusions obtained from this elevated priority run generally remain the same as those established earlier using confidence intervals. The elevated priority computer running time when the objective function coefficients were arranged from smallest to largest ranged from -8% and +16% compared to the standard mathematical formulation. When the objective function coefficients were arranged from smallest to largest by region, or row, the range was between -10% and + 18%.

Summarizing, Haverly System's HS/LP uses a modified branch and bound algorithm, which appears to account for the unpredictable response to reordering of the objective function coefficients. HS/LP may benefit from reordering the objective function coefficients

Chapter 4

CONCLUSIONS AND SUGGESTIONS FOR FURTHER STUDY

Why Does it Work?

For integer programming codes that use a pure branch and bound algorithm this thesis documents a reduction in computer running time ranging from 0% up to 18% by rearranging the objective function coefficients, from smallest to largest. Why does this work?

Dr. R. E. D. Woolsey of the Colorado School of Mines first proposed reordering objective function coefficients in Zionts' 1974 book Linear and Integer Programming (481) and a year later in Salkin's book Integer Programming (513). However, Woolsey did not show any computational results for his work. In Integer Programming Woolsey explains why ordering objective function coefficients should reduce computer running time.

> If the implicit enumeration {branch and bound} code we applied to the problem considered fixing variables from left to right also, it would tend to first find a binary solution by assigning every customer to the bank with the smallest total of fixed and variable costs. That is, it would assign *first* all the customers to the bank with the leftmost fixed charge. It would then move rightward through the higher fixed charges if and only if there existed some combination of fixed and variable charges cheaper than the present solution. In short, we have forced the early solutions to be "good" and then the remaining possibilities should

> be implicitly enumerated quickly.... Simply put, there is little question that running time will be reduced if we can start off with the best possible bound on the solution (Salkin 1975, 513).

When the objective function coefficients were rearranged from smallest to largest by region, or row, a greater reduction in computer running time when using branch and bound (from 0% up to 28%) was discovered. Why does reordering by region work?

Referring back to the small sample problem in Chapter 1 (Salkin 1975, 494), the standard mathematical formulation of this lockbox problem was:

$$\min z = 200x_{11} + 200x_{12} + 1050x_{13} + 1200x_{14} + 100x_{21} + 150x_{22} + 1200x_{23} + 1400x_{24} + 300x_{31} + 100x_{32} + 750x_{33} + 2000x_{34} + 450y_1 + 400y_2 + 500y_3$$

$$\text{s.t.} \quad x_{11} + x_{21} + x_{31} = 1$$

$$x_{12} + x_{22} + x_{32} = 1$$

$$x_{13} + x_{23} + x_{33} = 1$$

$$x_{14} + x_{24} + x_{34} = 1$$

$$x_{11} + x_{12} + x_{13} + x_{14} - 4y_1 \le 0$$

$$x_{21} + x_{22} + x_{23} + x_{24} - 4y_2 \le 0$$

$$x_{31} + x_{32} + x_{33} + x_{34} - 4y_3 \le 0$$

Rearranging the objective function of our small sample problem from smallest to largest variable costs (x_{ij}) and fixed costs (y_i) by region, or row, as explained earlier in Chapter 3, the new mathematical formulation is:

$$\min z = 100x_{21} + 200x_{11} + 300x_{31} + 100x_{32} + 150x_{22} + 200x_{12} + 750x_{33} + 1050x_{13} + 1200x_{23} + 1200x_{14} + 1400x_{24} + 2000x_{34} + 400y_2 + 450y_1 + 500y_3$$

s.t.

$$x_{11} + x_{21} + x_{31} = 1$$

$$x_{12} + x_{22} + x_{32} = 1$$

$$x_{13} + x_{23} + x_{33} = 1$$

$$x_{14} + x_{24} + x_{34} = 1$$

$$x_{11} + x_{12} + x_{13} + x_{14} - 4y_1 \leq 0$$

$$x_{21} + x_{22} + x_{23} + x_{24} - 4y_2 \leq 0$$

$$x_{31} + x_{32} + x_{33} + x_{34} - 4y_3 \leq 0$$

Looking at the first set of constraints, observe that they are all equalities, equal to one. That is, only one of the x_{ij} values could be one, the rest <u>have</u> to be zero. If the objective function coefficients were arranged from smallest to largest <u>by region</u>, each equality constraints would <u>start</u> with the smallest variable cost by region. In the example above the smallest x_{ij} value, by region, is x_{21}, x_{32}, x_{33}, and x_{14}.

Continuing, each of the equality constraints would fix each of the smallest x_{ij} values, by region, to one. The rest <u>must</u> be zero.

This thesis continually referred to reordering the objective function coefficients by region, or row. It is really reordering the objective function based upon the equality constraints. The key is to solve the equalities first.

Summarizing, this thesis supports the conclusion that, paraphrasing Woolsey, by reordering the objective function coefficients from smallest to largest by region the computer will start with a "better" starting solution. This reordering is based upon the first set of constraints, the equality constraints. Running time will be reduced because the algorithm only has to check if the solution is getting better or worse, not how much. It has to compare one number to another, and determine which one is bigger. Computers, and people, can do this very quickly.

Conclusions

When using STORM's pure branch and bound algorithm and reordering the objective function coefficients computer running time was reduced. As Table 2 and 3 in Chapter 3 documented, the reduction in computer running time using STORM ranged from 0% to 18% by rearranging the objective function coefficients from smallest to largest. If the objective function coefficients were rearranged from smallest to largest based on region, or row, this thesis discovered a greater reduction in computer running time using (from 0% up to 28%) when using STORM's standard branch and bound algorithm.

When using LINDO, however, there was <u>not</u> a uniform decrease in computer running time when the objective function coefficients were reordered. For some problems the running time increased. This appears to be caused by LINDO's unique variable and set based branching algorithms. LINDO's branching algorithm makes reordering the objective function coefficients irrelevant.

Running on a mainframe, Haverly Systems' HS/LP uses a modified pure branch and bound algorithm. This appears to account for the unpredictable response to reordering of the objective function coefficients from smallest to largest, ranging from -13% to +13% when compared to the standard mathematical formulation. When the objective function coefficients were arranged from smallest to largest by region, or row, the range was between -12% and +20%. HS/LP may benefit from reordering the objective function coefficients.

In order to successfully implement the time savings discovered in this thesis the computer user must know what integer programming algorithm their software uses. This is found in the software documentation. If the software uses a pure branch and bound algorithm, computer running time for some types of integer programming problems may be reduced from 0% up to 28% by reordering the objective function coefficients.

Suggestions for Further Study

This thesis concentrated on the importance of ordering objective function coefficients when solving integer programming problems. The method described in this thesis is generalizable to many other types of integer programming problems. One of the most important is the Lorie-Savage (1955, 229) formulations of capital budgeting problems. These problems attempt to maximize the net present value of a group of alternative investment possibilities subject to a series of constraints on corporate burden, and different budgets available. Examples of these problems may be found in Hesse (1974, 263), Winston (1987, 366), and Zionts (1975, 385).

Another group of problems that may benefit from the application of reordering the objective function coefficients is distribution/allocation problems. One immediate example is that of distributing a mineral resource from many production facilities to many demand points, with upper and lower bounds on the flows. These are also known as the capacitated network flow problems. These problems are particularly difficult to solve when there are different costs for different levels of demand.

The importance of ordering constraints in integer programming problems is another area for study. This idea first appeared in 1974; Zionts published Linear and Integer Programming. Writing Chapter 17 of Zionts' book, Woolsey first introduced his concept of ordering constraints when formulating integer programming problems.

> The order of rows is also important for implicit enumeration methods .... {W}e should order the constraints ... so that those most likely to be decisive are scanned first.
>
> The first classification of constraints likely to be decisive, based on experience, is equality constraints .... The next most decisive classification of constraints is inequalities which enforce logical conditions (Zionts 1974, 478).

The next year Woolsey wrote Chapter 13 in Salkin's 1975 book Integer Programming. In his section titled "How to Formulate so as to Minimize Solution Time (Implicit Enumeration Methods)" Woolsey again addresses ordering constraints (Salkin 1975, 513):

> It should be equally apparent that we should structure the problem so as to force the code to eliminate as many unfruitful paths as possible. Another way to do this is to put the most binding constraints at the top of the problem and put the least binding constraints at the bottom. By "binding" we mean the constraint that is most likely to limit the set of solutions.... As many implicit enumeration criteria consider constraints from top to bottom of the constraint matrix, it should be clear that putting the {most binding} constraint at the top will reduce computation time.

These are the only two references in the field of integer programming, both by Woolsey, that concern ordering constraints. Neither reference showed any computational results. Ordering constraints is an area for further study.

Another area for further study is to use the process of aggregating constraints proposed by Elmaghraby and Wig (1970). This combines all the equalities in the constraints into one equality constraint. Sample problems could then be run with and without the aggregation to discover if there is a reduction in computer running time.

A final suggestion is to use the reordering procedure suggested in this thesis, but using a Balǎs additive algorithm to determine possible reductions in computer running time. This could be tested with and without aggregation of the equality constraints. We conjecture that an additive algorithm, such as Balǎs, would quickly obtain a near-optimal solution. The remainder of the computer running time would be simply confirming that the earlier answer was indeed optimum. Confirmation of this conjecture would require a study similar to that of this thesis.

Summarizing, we set out to confirm or deny the conjecture that reordering of the objective function for integer programming lockbox problems could improve running time. Based on this study we may assert that computer running time can be reduced up to 28% by thoughtful reordering of the objective function coefficients. We assert that the above finding could, in many cases, justify preprocessing the data as described in this thesis prior to running the problem on a branch and bound integer programming code.

REFERENCES CITED

Camm, J. D., A. S. Raturi, and S. Tsubakitani. 1990. Cutting big M down to size. Interfaces. 20 (no. 5): 61-66.

Ciochetto, F. F. Jr. 1971. An application of binary integer programming to the lock-box problem. Master's thesis, Colorado School of Mines, Golden, Colo.

Ciochetto, F. F., H. S. Swanson, J. R. Lee, and R. E. D. Woolsey. 1972. The lock-box problem and some startling but true computational results for large scale systems. Working paper presented at the 41st National Meeting of the Operations Research Society of America, April 26-28, 1972. New Orleans, La.

Crowder, H., E. L. Johnson, and M. Padberg. 1983. Solving large-scale zero-one linear programming problems. Operations research. 31 (no. 5): 803-834.

Digital Equipment Corporation. 1985. VAX hardware handbook, volume 1-1986. Maynard, Mass: Digital Equipment Corporation.

Elmaghraby, S., and M. Wig. 1970. On the treatment of stock cutting problems as diophantine programs. Operations Reseach Center Report No. 61, North Carolina State University at Raleigh. May 11, 1970.

Emmons, H., et al. 1989. STORM personal version 2.0, quantitative modeling for decision support. Oakland, Calif: Holden-Day, Inc.

Erlenkotter, D. 1978. A dual-based procedure for uncapacitated facility location. Operations research. 26 (no. 6): 992-1009.

Haverly Systems Inc. 1985. HS/LP version 1.50 haverly systems linear programming system user's reference manual. Denville, N.J.: Haverly Systems Inc.

Hesse, R. and R. E. D. Woolsey. 1980. Applied management science. Chicago: Science Research Associates.

Johnson, E. L., M. M. Kostreva, and U. H. Suhl. 1985. Solving 0-1 integer programming problems arising from large scale planning models. Operations research. 33 (no. 4): 803-819.

Kraus, A., C. Janssen, and A. McAdams. 1970. The lock-box location problem. Journal of bank research. Autumn 1970: 51-58.

Levy, F. K. 1966. An application of heuristic problem solving to accounts receivable management. Management science. February 1966: B-236 - B-244.

Lorie, J., and L. Savage. 1955. Three problems in capital rationing. Journal of business. 28: 229-239.

Maclean, P. 1990. Introducing PAN.COM, the program animator. PC Magazine. 15 May, 253-259.

McCarville, K. 1991. Interview with author.

Peter Norton Computing, Inc. 1988. The norton utilities advanced edition, version 4.5. Santa Monica, CA: Peter Norton Computing, Inc.

Salkin, H. M. 1975. Integer programming. Reading, Mass: Addison-Wesley.

Schrage, L. 1989. User's manual for linear, integer, and quadratic programming with LINDO. San Francisco, Calif: The Scientific Press

Walpole, R. E., and R. H. Myers. 1989. Probability and statistics for engineers. 4th ed. New York: MacMillan Publishing Co.

Williams, H. P. 1983. A reduction procedure for linear and integer programming models. In Redundancy in mathematical programming, ed. Karwan, M. H., et al., 87-107. Lecture Notes in Economics and Mathematical Systems. Berlin: Springer-Verlag.

Winston, W. L. 1987. Operations research: applications and algorithms. Boston: Dusbury Press.

Wolverton, V. 1986. Supercharging MS-DOS. Redmond, Wash: Microsoft Press.

________. 1988. Running MS-DOS. Redmond, Wash: Microsoft Press.

Zionts, S. 1974. Linear and integer programming. New Jersey: Prentice-Hall, Inc.

SELECTED BIBLIOGRAPHY

Dunham, L. 1987. Thesis writer's handbook. Graduate School, Colorado School of Mines, Golden, Colo.

Kickbush, D. A. 1990. A heuristic for solving the integer programming lockbox problem. Master's thesis, Colorado School of Mines, Golden, Colo.

Woolsey, R. E. D. 1989. Class Notes - Mathematics Course MA522A - Operations Research. Colorado School of Mines, Golden, Colo.

Woolsey, R. E. D. 1990. Class Notes - Mathematics Course MA582A - Integer Programming. Colorado School of Mines, Golden, Colo.

Appendix

SAMPLE PROBLEM FORMULATIONS

Problem 1 (Hesse 1980, 268).

Lockbox Problem

Costs	Region	Lockbox 1	Lockbox 2	Lockbox 3
Variable	1	25	24	29
	2	20	32	26
	3	16	18	14
	4	16	25	27
	5	40	42	45
	6	18	20	25
	7	19	20	13
	8	16	19	18
	9	24	36	45
	A	32	25	15
Fixed		50	35	70

Problem 1 (Hesse 1980, 268).

Standard Mathematical Formulation

$$\min z = 25x_{11} + 20x_{12} + 16x_{13} + 16x_{14} + 40x_{15} + 18x_{16} + 19x_{17}$$
$$+ 16x_{18} + 24x_{19} + 32x_{1A} + 24x_{21} + 32x_{22} + 18x_{23} + 25x_{24}$$
$$+ 42x_{25} + 20x_{26} + 20x_{27} + 19x_{28} + 36x_{29} + 25x_{2A} + 29x_{31}$$
$$+ 26x_{32} + 14x_{33} + 27x_{34} + 45x_{35} + 25x_{36} + 13x_{37} + 18x_{38}$$
$$+ 45x_{39} + 15x_{3A} + 50y_1 + 35y_2 + 70y_3$$

s.t.

$$x_{11} + x_{21} + x_{31} = 1$$
$$x_{12} + x_{22} + x_{32} = 1$$
$$x_{13} + x_{23} + x_{33} = 1$$
$$x_{14} + x_{24} + x_{34} = 1$$
$$x_{15} + x_{25} + x_{35} = 1$$
$$x_{16} + x_{26} + x_{36} = 1$$
$$x_{17} + x_{27} + x_{37} = 1$$
$$x_{18} + x_{28} + x_{38} = 1$$
$$x_{19} + x_{29} + x_{39} = 1$$
$$x_{1A} + x_{2A} + x_{3A} = 1$$
$$x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} + x_{18} + x_{19} + x_{1A} - 10y_1 \le 0$$
$$x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} + x_{27} + x_{28} + x_{29} + x_{2A} - 10y_2 \le 0$$
$$x_{31} + x_{32} + x_{33} + x_{34} + x_{35} + x_{36} + x_{37} + x_{38} + x_{39} + x_{3A} - 10y_3 \le 0$$

Problem 1 (Hesse 1980, 268).

New Objective Function,

Smallest to Largest x_{ij} y_i Values

$$\min z = 13x_{37} + 14x_{33} + 15x_{3A} + 16x_{13} + 16x_{14} + 16x_{18} + 18x_{16} + 18x_{23} + 18x_{38} + 19x_{17} + 19x_{28} + 20x_{12} + 20x_{26} + 20x_{27} + 24x_{19} + 24x_{21} + 25x_{11} + 25x_{24} + 25x_{2A} + 25x_{36} + 26x_{32} + 27x_{34} + 29x_{31} + 32x_{1A} + 32x_{22} + 36x_{29} + 40x_{15} + 42x_{25} + 45x_{35} + 45x_{39} + 35y_2 + 50y_1 + 70y_3$$

Problem 1 (Hesse 1980, 268).

New Objective Function,

Smallest to Largest x_{ij} y_i Values,

By Region

$$\min z = 24x_{21} + 25x_{11} + 29x_{31} + 20x_{12} + 26x_{32} + 32x_{22} + 14x_{33} + 16x_{13} + 18x_{23} + 16x_{14} + 25x_{24} + 27x_{34} + 40x_{15} + 42x_{25} + 45x_{35} + 18x_{16} + 20x_{26} + 25x_{36} + 13x_{37} + 19x_{17} + 20x_{27} + 16x_{18} + 18x_{38} + 19x_{28} + 24x_{19} + 36x_{29} + 45x_{39} + 15x_{3A} + 25x_{2A} + 32x_{1A} + 35y_2 + 50y_1 + 70y_3$$

Problem 2 (Hesse 1980, 269).

Lockbox Problem

Costs	Region	Lockbox 1	Lockbox 2	Lockbox 3	Lockbox 4
Variable	1	40	10	70	90
	2	60	30	40	80
	3	30	90	50	60
	4	20	80	90	60
	5	70	30	80	40
	6	50	20	40	70
	7	50	90	30	30
	8	20	50	30	50
	9	60	70	20	40
	A	90	70	60	30
Fixed		150	120	140	170

Problem 2 (Hesse 1980, 269).

Standard Mathematical Formulation

$$\begin{aligned}\min z = {} & 40x_{11} + 60x_{12} + 30x_{13} + 20x_{14} + 70x_{15} + 50x_{16} + 50x_{17} \\ & + 20x_{18} + 60x_{19} + 90x_{1A} + 10x_{21} + 30x_{22} + 90x_{23} + 80x_{24} \\ & + 30x_{25} + 20x_{26} + 90x_{27} + 50x_{28} + 70x_{29} + 70x_{2A} + 70x_{31} \\ & + 40x_{32} + 50x_{33} + 90x_{34} + 80x_{35} + 40x_{36} + 30x_{37} + 30x_{38} \\ & + 20x_{39} + 60x_{3A} + 90x_{41} + 80x_{42} + 60x_{43} + 60x_{44} + 40x_{45} \\ & + 70x_{46} + 20x_{47} + 50x_{48} + 40x_{49} + 30x_{4A} + 150y_1 + 120y_2 \\ & + 140y_3 + 170y_4\end{aligned}$$

$$\begin{aligned}\text{s.t.}\quad & x_{11} + x_{21} + x_{31} + x_{41} = 1 \\ & x_{12} + x_{22} + x_{32} + x_{42} = 1 \\ & x_{13} + x_{23} + x_{33} + x_{43} = 1 \\ & x_{14} + x_{24} + x_{34} + x_{44} = 1 \\ & x_{15} + x_{25} + x_{35} + x_{45} = 1 \\ & x_{16} + x_{26} + x_{36} + x_{46} = 1 \\ & x_{17} + x_{27} + x_{37} + x_{47} = 1 \\ & x_{18} + x_{28} + x_{38} + x_{48} = 1 \\ & x_{19} + x_{29} + x_{39} + x_{49} = 1 \\ & x_{1A} + x_{2A} + x_{3A} + x_{4A} = 1 \\ & x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} + x_{18} + x_{19} + x_{1A} \\ & \quad - 10y_1 \le 0 \\ & x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} + x_{27} + x_{28} + x_{29} + x_{2A} \\ & \quad - 10y_2 \le 0\end{aligned}$$

Problem 2 (Continued).

Standard Mathematical Formulation

$$x_{31} + x_{32} + x_{33} + x_{34} + x_{35} + x_{36} + x_{37} + x_{38} + x_{39} + x_{3A} - 10y_3 \leq 0$$

$$x_{41} + x_{42} + x_{43} + x_{44} + x_{45} + x_{46} + x_{47} + x_{48} + x_{49} + x_{4A} - 10y_4 \leq 0$$

Problem 2 (Hesse 1980, 269).

New Objective Function,

Smallest to Largest x_{ij} y_i Values

$$\min z = 10x_{21} + 20x_{14} + 20x_{18} + 20x_{26} + 20x_{39} + 20x_{47} + 30x_{13} + 30x_{22} + 30x_{25} + 30x_{37} + 30x_{38} + 30x_{4A} + 40x_{11} + 40x_{32} + 40x_{36} + 40x_{45} + 40x_{49} + 50x_{16} + 50x_{17} + 50x_{28} + 50x_{33} + 50x_{48} + 60x_{12} + 60x_{19} + 60x_{3A} + 60x_{43} + 60x_{44} + 70x_{15} + 70x_{29} + 70x_{2A} + 70x_{31} + 70x_{46} + 80x_{24} + 80x_{35} + 80x_{42} + 90x_{1A} + 90x_{23} + 90x_{27} + 90x_{34} + 90x_{41} + 120y_2 + 140y_3 + 150y_1 + 170y_4$$

Problem 2 (Hesse 1980, 269).

New Objective Function,

Smallest to Largest x_{ij} y_i Values,

By Region

$$\min z = 10x_{21} + 40x_{11} + 70x_{31} + 90x_{41} + 30x_{22} + 40x_{32} + 60x_{12} + 80x_{42} + 30x_{13} + 50x_{33} + 60x_{43} + 90x_{23} + 20x_{14} + 60x_{44} + 80x_{24} + 90x_{34} + 30x_{25} + 40x_{45} + 70x_{15} + 80x_{35} + 20x_{26} + 40x_{36} + 50x_{16} + 70x_{46} + 20x_{47} + 30x_{37} + 50x_{17} + 90x_{27} + 20x_{18} + 30x_{38} + 50x_{28} + 50x_{48} + 20x_{39} + 40x_{49} + 60x_{19} + 70x_{29} + 30x_{4A} + 60x_{3A} + 70x_{2A} + 90x_{1A} + 120y_2 + 140y_3 + 150y_1 + 170y_4$$

Problem 3 (Hesse 1980, 270).

Lockbox Problem

Costs	Region	Lbox 1	Lbox 2	Lbox 3	Lbox 4	Lbox 5
Variable	1	15	20	25	28	32
	2	45	41	36	28	24
	3	18	25	19	14	16
	4	13	15	20	18	19
	5	25	22	20	17	18
	6	45	30	42	43	40
	7	27	29	25	20	16
	8	14	10	18	20	16
	9	26	30	32	22	20
	A	29	26	24	20	25
	B	28	32	25	22	26
	C	17	15	21	18	25
	D	38	32	23	27	20
	E	17	19	24	20	26
	F	20	16	25	21	24
	G	18	25	32	30	37
Fixed		50	40	35	50	70

Problem 3 (Hesse 1980, 270).

Standard Mathematical Formulation

$$
\begin{aligned}
\min z = {} & 15x_{11} + 45x_{12} + 18x_{13} + 13x_{14} + 25x_{15} + 45x_{16} + 27x_{17} \\
& + 14x_{18} + 26x_{19} + 29x_{1A} + 28x_{1B} + 17x_{1C} + 38x_{1D} + 17x_{1E} \\
& + 20x_{1F} + 18x_{1G} + 20x_{21} + 41x_{22} + 25x_{23} + 15x_{24} + 22x_{25} \\
& + 30x_{26} + 29x_{27} + 10x_{28} + 30x_{29} + 26x_{2A} + 32x_{2B} + 15x_{2C} \\
& + 32x_{2D} + 19x_{2E} + 16x_{2F} + 25x_{2G} + 25x_{31} + 36x_{32} + 19x_{33} \\
& + 20x_{34} + 20x_{35} + 42x_{36} + 25x_{37} + 18x_{38} + 32x_{39} + 24x_{3A} \\
& + 25x_{3B} + 21x_{3C} + 23x_{3D} + 24x_{3E} + 25x_{3F} + 32x_{3G} + 28x_{41} \\
& + 28x_{42} + 14x_{43} + 18x_{44} + 17x_{45} + 43x_{46} + 20x_{47} + 20x_{48} \\
& + 22x_{49} + 20x_{4A} + 22x_{4B} + 18x_{4C} + 27x_{4D} + 20x_{4E} + 21x_{4F} \\
& + 30x_{4G} + 32x_{51} + 24x_{52} + 16x_{53} + 19x_{54} + 18x_{55} + 40x_{56} \\
& + 16x_{57} + 16x_{58} + 20x_{59} + 25x_{5A} + 26x_{5B} + 25x_{5C} + 20x_{5D} \\
& + 26x_{5E} + 24x_{5F} + 37x_{5G} + 50y_1 + 40y_2 + 35y_3 + 50y_4 + 70y_5
\end{aligned}
$$

$$
\begin{aligned}
\text{s.t.}\quad & x_{11} + x_{21} + x_{31} + x_{41} + x_{51} = 1 \\
& x_{12} + x_{22} + x_{32} + x_{42} + x_{52} = 1 \\
& x_{13} + x_{23} + x_{33} + x_{43} + x_{53} = 1 \\
& x_{14} + x_{24} + x_{34} + x_{44} + x_{54} = 1 \\
& x_{15} + x_{25} + x_{35} + x_{45} + x_{55} = 1 \\
& x_{16} + x_{26} + x_{36} + x_{46} + x_{56} = 1 \\
& x_{17} + x_{27} + x_{37} + x_{47} + x_{57} = 1 \\
& x_{18} + x_{28} + x_{38} + x_{48} + x_{58} = 1 \\
& x_{19} + x_{29} + x_{39} + x_{49} + x_{59} = 1
\end{aligned}
$$

Problem 3 (Continued).

Standard Mathematical Formulation

$$x_{1A} + x_{2A} + x_{3A} + x_{4A} + x_{5A} = 1$$

$$x_{1B} + x_{2B} + x_{3B} + x_{4B} + x_{5B} = 1$$

$$x_{1C} + x_{2C} + x_{3C} + x_{3C} + x_{5C} = 1$$

$$x_{1D} + x_{2D} + x_{3D} + x_{4D} + x_{5D} = 1$$

$$x_{1E} + x_{2E} + x_{3E} + x_{4E} + x_{5E} = 1$$

$$x_{1F} + x_{2F} + x_{3F} + x_{4F} + x_{5F} = 1$$

$$x_{1G} + x_{2G} + x_{3G} + x_{4G} + x_{5G} = 1$$

$$x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} + x_{18} + x_{19} + x_{1A}$$
$$+ x_{1B} + x_{1C} + x_{1D} + x_{1E} + x_{1F} + x_{1G} - 16y_1 \leq 0$$

$$x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} + x_{27} + x_{28} + x_{29} + x_{2A}$$
$$+ x_{2B} + x_{2C} + x_{2D} + x_{2E} + x_{2F} + x_{2G} - 16y_2 \leq 0$$

$$x_{31} + x_{32} + x_{33} + x_{34} + x_{35} + x_{36} + x_{37} + x_{38} + x_{39} + x_{3A}$$
$$+ x_{3B} + x_{3C} + x_{3D} + x_{3E} + x_{3F} + x_{3G} - 16y_3 \leq 0$$

$$x_{41} + x_{42} + x_{43} + x_{44} + x_{45} + x_{46} + x_{47} + x_{48} + x_{49} + x_{4A}$$
$$+ x_{4B} + x_{4C} + x_{4D} + x_{4E} + x_{4F} + x_{4G} - 16y_4 \leq 0$$

$$x_{51} + x_{52} + x_{53} + x_{54} + x_{55} + x_{56} + x_{57} + x_{58} + x_{59} + x_{5A}$$
$$+ x_{5B} + x_{5C} + x_{5D} + x_{5E} + x_{5F} + x_{5G} - 16y_5 \leq 0$$

Problem 3 (Hesse 1980, 270).

New Objective Function,

Smallest to Largest x_{ij} y_i Values

$$
\begin{aligned}
\min z = {} & 10x_{28} + 13x_{14} + 14x_{18} + 14x_{43} + 15x_{11} + 15x_{24} + 15x_{2C} \\
& + 16x_{2F} + 16x_{53} + 16x_{57} + 16x_{58} + 17x_{1C} + 17x_{1E} + 17x_{45} \\
& + 18x_{13} + 18x_{1G} + 18x_{38} + 18x_{44} + 18x_{4C} + 18x_{55} + 19x_{2E} \\
& + 19x_{33} + 19x_{54} + 20x_{1F} + 20x_{21} + 20x_{34} + 20x_{35} + 20x_{47} \\
& + 20x_{48} + 20x_{4A} + 20x_{4E} + 20x_{59} + 20x_{5D} + 21x_{3C} + 21x_{4F} \\
& + 22x_{25} + 22x_{49} + 22x_{4B} + 23x_{3D} + 24x_{3A} + 24x_{3E} + 24x_{52} \\
& + 24x_{5F} + 25x_{15} + 25x_{23} + 25x_{2G} + 25x_{31} + 25x_{37} + 25x_{3B} \\
& + 25x_{3F} + 25x_{5A} + 25x_{5C} + 26x_{19} + 26x_{2A} + 26x_{5B} + 26x_{5E} \\
& + 27x_{17} + 27x_{4D} + 28x_{1B} + 28x_{41} + 28x_{42} + 29x_{1A} + 29x_{27} \\
& + 30x_{26} + 30x_{29} + 30x_{4G} + 32x_{2B} + 32x_{2D} + 32x_{39} + 32x_{3G} \\
& + 32x_{51} + 36x_{32} + 37x_{5G} + 38x_{1D} + 40x_{56} + 41x_{22} + 42x_{36} \\
& + 43x_{46} + 45x_{12} + 45x_{16} + 35y_{3} + 40y_{2} + 50y_{1} + 50y_{4} + 70y_{5}
\end{aligned}
$$

Problem 3 (Hesse 1980, 270).

New Objective Function,

Smallest to Largest x_{ij} y_i Values,

By Region

$$
\begin{aligned}
\min z = {} & 15x_{11} + 20x_{21} + 25x_{31} + 28x_{41} + 32x_{51} + 24x_{52} + 28x_{42} \\
& + 36x_{32} + 41x_{22} + 45x_{12} + 14x_{43} + 16x_{53} + 18x_{13} + 19x_{33} \\
& + 25x_{23} + 13x_{14} + 15x_{24} + 18x_{44} + 19x_{54} + 20x_{34} + 17x_{45} \\
& + 18x_{55} + 20x_{35} + 22x_{25} + 25x_{15} + 30x_{26} + 40x_{56} + 42x_{36} \\
& + 43x_{46} + 45x_{16} + 16x_{57} + 20x_{47} + 25x_{37} + 27x_{17} + 29x_{27} \\
& + 10x_{28} + 14x_{18} + 16x_{58} + 18x_{38} + 20x_{48} + 20x_{59} + 22x_{49} \\
& + 26x_{19} + 30x_{29} + 32x_{39} + 20x_{4A} + 24x_{3A} + 25x_{5A} + 26x_{2A} \\
& + 29x_{1A} + 22x_{4B} + 25x_{3B} + 26x_{5B} + 28x_{1B} + 32x_{2B} + 15x_{2C} \\
& + 17x_{1C} + 18x_{4C} + 21x_{3C} + 25x_{5C} + 20x_{5D} + 23x_{3D} + 27x_{4D} \\
& + 32x_{2D} + 38x_{1D} + 17x_{1E} + 19x_{2E} + 20x_{4E} + 24x_{3E} + 26x_{5E} \\
& + 16x_{2F} + 20x_{1F} + 21x_{4F} + 24x_{5F} + 25x_{3F} + 18x_{1G} + 25x_{2G} \\
& + 30x_{4G} + 32x_{3G} + 37x_{5G} + 35y_{3} + 40y_{2} + 50y_{1} + 50y_{4} + 70y_{5}
\end{aligned}
$$

Problem 4 (Hesse 1980, 271).

Lockbox Problem

Costs	Region	Lbox 1	Lbox 2	Lbox 3	Lbox 4	Lbox 5
Variable	1	40	60	80	100	120
	2	20	40	60	80	100
	3	40	20	40	60	80
	4	60	40	20	40	60
	5	80	60	40	20	40
	6	100	80	60	40	20
	7	20	40	60	80	100
	8	0	20	40	60	80
	9	20	0	20	40	60
	A	40	20	0	20	40
	B	60	40	20	0	20
	C	80	60	40	20	0
Fixed		200	275	280	150	110

Problem 4 (Hesse 1980, 271).

Standard Mathematical Formulation

$$\begin{aligned} \min z = {} & 40x_{11} + 20x_{12} + 40x_{13} + 60x_{14} + 80x_{15} + 100x_{16} + 20x_{17} \\ & + 0x_{18} + 20x_{19} + 40x_{1A} + 60x_{1B} + 80x_{1C} + 60x_{21} + 40x_{22} \\ & + 20x_{23} + 40x_{24} + 60x_{25} + 80x_{26} + 40x_{27} + 20x_{28} + 0x_{29} \\ & + 20x_{2A} + 40x_{2B} + 60x_{2C} + 80x_{31} + 60x_{32} + 40x_{33} + 20x_{34} \\ & + 40x_{35} + 60x_{36} + 60x_{37} + 40x_{38} + 20x_{39} + 0x_{3A} + 20x_{3B} \\ & + 40x_{3C} + 100x_{41} + 80x_{42} + 60x_{43} + 40x_{44} + 20x_{45} + 40x_{46} \\ & + 80x_{47} + 60x_{48} + 40x_{49} + 20x_{4A} + 0x_{4B} + 20x_{4C} + 120x_{51} \\ & + 100x_{52} + 80x_{53} + 60x_{54} + 40x_{55} + 20x_{56} + 100x_{57} + 80x_{58} \\ & + 60x_{59} + 40x_{5A} + 20x_{5B} + 0x_{5C} + 200y_1 + 275y_2 + 280y_3 \\ & + 150y_4 + 110y_5 \end{aligned}$$

$$\begin{aligned} \text{s.t.} \quad & x_{11} + x_{21} + x_{31} + x_{41} + x_{51} = 1 \\ & x_{12} + x_{22} + x_{32} + x_{42} + x_{52} = 1 \\ & x_{13} + x_{23} + x_{33} + x_{43} + x_{53} = 1 \\ & x_{14} + x_{24} + x_{34} + x_{44} + x_{54} = 1 \\ & x_{15} + x_{25} + x_{35} + x_{45} + x_{55} = 1 \\ & x_{16} + x_{26} + x_{36} + x_{46} + x_{56} = 1 \\ & x_{17} + x_{27} + x_{37} + x_{47} + x_{57} = 1 \\ & x_{18} + x_{28} + x_{38} + x_{48} + x_{58} = 1 \\ & x_{19} + x_{29} + x_{39} + x_{49} + x_{59} = 1 \\ & x_{1A} + x_{2A} + x_{3A} + x_{4A} + x_{5A} = 1 \\ & x_{1B} + x_{2B} + x_{3B} + x_{4B} + x_{5B} = 1 \end{aligned}$$

Problem 4 (Continued).

Standard Mathematical Formulation

$$x_{1C} + x_{2C} + x_{3C} + x_{4C} + x_{5C} = 1$$

$$x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} + x_{18} + x_{19} + x_{1A} + x_{1B} + x_{1C} - 12y_1 \leq 0$$

$$x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} + x_{27} + x_{28} + x_{29} + x_{2A} + x_{2B} + x_{2C} - 12y_2 \leq 0$$

$$x_{31} + x_{32} + x_{33} + x_{34} + x_{35} + x_{36} + x_{37} + x_{38} + x_{39} + x_{3A} + x_{3B} + x_{3C} - 12y_3 \leq 0$$

$$x_{41} + x_{42} + x_{43} + x_{44} + x_{45} + x_{46} + x_{47} + x_{48} + x_{49} + x_{4A} + x_{4B} + x_{4C} - 12y_4 \leq 0$$

$$x_{51} + x_{52} + x_{53} + x_{54} + x_{55} + x_{56} + x_{57} + x_{58} + x_{59} + x_{5A} + x_{5B} + x_{5C} - 12y_5 \leq 0$$

Problem 4 (Hesse 1980, 271).

New Objective Function,

Smallest to Largest x_{ij} y_i Values

$$\begin{aligned}
\min z = {} & 0x_{18} + 0x_{29} + 0x_{3A} + 0x_{4B} + 0x_{5C} + 20x_{12} + 20x_{17} + 20x_{19} \\
& + 20x_{23} + 20x_{28} + 20x_{2A} + 20x_{34} + 20x_{39} + 20x_{3B} + 20x_{45} \\
& + 20x_{4A} + 20x_{4C} + 20x_{56} + 20x_{5B} + 40x_{11} + 40x_{13} + 40x_{1A} \\
& + 40x_{22} + 40x_{24} + 40x_{27} + 40x_{2B} + 40x_{33} + 40x_{35} + 40x_{38} \\
& + 40x_{3C} + 40x_{44} + 40x_{46} + 40x_{49} + 40x_{55} + 40x_{5A} + 60x_{14} \\
& + 60x_{1B} + 60x_{21} + 60x_{25} + 60x_{2C} + 60x_{32} + 60x_{36} + 60x_{37} \\
& + 60x_{43} + 60x_{48} + 60x_{54} + 60x_{59} + 80x_{15} + 80x_{1C} + 80x_{26} \\
& + 80x_{31} + 80x_{42} + 80x_{47} + 80x_{53} + 80x_{58} + 100x_{16} + 100x_{41} \\
& + 100x_{52} + 100x_{57} + 120x_{51} + 110y_5 + 150y_4 + 200y_1 + 275y_2 \\
& + 280y_3
\end{aligned}$$

Problem 4 (Hesse 1980, 271).

New Objective Function,

Smallest to Largest x_{ij} y_i Values,

<u>By Region</u>

$$\begin{aligned} \min z = {} & 40x_{11} + 60x_{21} + 80x_{31} + 100x_{41} + 120x_{51} + 20x_{12} + 40x_{22} \\ & + 60x_{32} + 80x_{42} + 100x_{52} + 20x_{23} + 40x_{13} + 40x_{33} + 60x_{43} \\ & + 80x_{53} + 20x_{34} + 40x_{24} + 40x_{44} + 60x_{14} + 60x_{54} + 20x_{45} \\ & + 40x_{35} + 40x_{55} + 60x_{25} + 80x_{15} + 20x_{56} + 40x_{46} + 60x_{36} \\ & + 80x_{26} + 100x_{16} + 20x_{17} + 40x_{27} + 60x_{37} + 80x_{47} + 100x_{57} \\ & + 0x_{18} + 20x_{28} + 40x_{38} + 60x_{48} + 80x_{58} + 0x_{29} + 20x_{19} + 20x_{39} \\ & + 40x_{49} + 60x_{59} + 0x_{3A} + 20x_{2A} + 20x_{4A} + 40x_{1A} + 40x_{5A} + 0x_{4B} \\ & + 20x_{3B} + 20x_{5B} + 40x_{2B} + 60x_{1B} + 0x_{5C} + 20x_{4C} + 40x_{3C} + 60x_{2C} \\ & + 80x_{1C} + 110y_5 + 150y_4 + 200y_1 + 275y_2 + 280y_3 \end{aligned}$$

Problem 5 (Hesse 1980, 271).

Lockbox Problem

Costs	Region	Lbox 1	Lbox 2	Lbox 3	Lbox 4	Lbox 5
Variable	1	3	9	11	4	9
	2	70	90	40	40	20
	3	0	63	108	60	100
	4	28	36	26	0	31
	5	5	4	6	3	9
	6	91	108	15	48	60
	7	155	185	60	65	84
	8	320	440	240	200	50
	9	160	170	75	105	105
	A	210	270	75	105	105
	B	25	17	20	10	37
	C	40	42	10	15	30
	D	1	5	7	3	7
	E	70	15	13	15	10
	F	16	14	7	6	20
	G	100	125	60	44	40
	H	16	30	33	20	40
	I	2	2	1	1	2
	J	300	420	240	200	0
	K	15	20	26	9	34
	L	6	0	11	7	15
	M	27	30	21	7	37
Fixed		100	150	200	125	250

Problem 5 (Hesse 1980, 271).

Standard Mathematical Formulation

$$\begin{aligned}
\min z = {} & 3x_{11} + 70x_{12} + 0x_{13} + 28x_{14} + 5x_{15} + 91x_{16} + 155x_{17} \\
& + 320x_{18} + 160x_{19} + 210x_{1A} + 25x_{1B} + 40x_{1C} + 1x_{1D} + 70x_{1E} \\
& + 16x_{1F} + 100x_{1G} + 16x_{1H} + 2x_{1I} + 300x_{1J} + 15x_{1K} + 6x_{1L} \\
& + 27x_{1M} + 9x_{21} + 90x_{22} + 63x_{23} + 36x_{24} + 4x_{25} + 108x_{26} \\
& + 185x_{27} + 440x_{28} + 170x_{29} + 270x_{2A} + 17x_{2B} + 42x_{2C} + 5x_{2D} \\
& + 15x_{2E} + 14x_{2F} + 125x_{2G} + 30x_{2H} + 2x_{2I} + 420x_{2J} + 20x_{2K} \\
& + 0x_{2L} + 30x_{2M} + 11x_{31} + 40x_{32} + 108x_{33} + 26x_{34} + 6x_{35} \\
& + 15x_{36} + 60x_{37} + 240x_{38} + 0x_{39} + 75x_{3A} + 20x_{3B} + 10x_{3C} \\
& + 7x_{3D} + 13x_{3E} + 7x_{3F} + 60x_{3G} + 33x_{3H} + 1x_{3I} + 240x_{3J} \\
& + 26x_{3K} + 11x_{3L} + 21x_{3M} + 4x_{41} + 40x_{42} + 60x_{43} + 0x_{44} \\
& + 3x_{45} + 48x_{46} + 65x_{47} + 200x_{48} + 75x_{49} + 105x_{4A} + 10x_{4B} \\
& + 15x_{4C} + 3x_{4D} + 15x_{4E} + 6x_{4F} + 44x_{4G} + 20x_{4H} + 1x_{4I} \\
& + 200x_{4J} + 9x_{4K} + 7x_{4L} + 7x_{4M} + 9x_{51} + 20x_{52} + 100x_{53} \\
& + 31x_{54} + 9x_{55} + 60x_{56} + 84x_{57} + 50x_{58} + 135x_{59} + 105x_{5A} \\
& + 37x_{5B} + 30x_{5C} + 7x_{5D} + 10x_{5E} + 20x_{5F} + 40x_{5G} + 40x_{5H} \\
& + 2x_{5I} + 0x_{5J} + 34x_{5K} + 15x_{5L} + 37x_{5M} + 100y_1 + 150y_2 \\
& + 200y_3 + 125y_4 + 250y_5
\end{aligned}$$

$$\begin{aligned}
\text{s.t.}\quad & x_{11} + x_{21} + x_{31} + x_{41} + x_{51} = 1 \\
& x_{12} + x_{22} + x_{32} + x_{42} + x_{52} = 1 \\
& x_{13} + x_{23} + x_{33} + x_{43} + x_{53} = 1 \\
& x_{14} + x_{24} + x_{34} + x_{44} + x_{54} = 1
\end{aligned}$$

Problem 5 (Continued).

Standard Mathematical Formulation

$$x_{15} + x_{25} + x_{35} + x_{45} + x_{55} = 1$$

$$x_{16} + x_{26} + x_{36} + x_{46} + x_{56} = 1$$

$$x_{17} + x_{27} + x_{37} + x_{47} + x_{57} = 1$$

$$x_{18} + x_{28} + x_{38} + x_{48} + x_{58} = 1$$

$$x_{19} + x_{29} + x_{39} + x_{49} + x_{59} = 1$$

$$x_{1A} + x_{2A} + x_{3A} + x_{4A} + x_{5A} = 1$$

$$x_{1B} + x_{2B} + x_{3B} + x_{4B} + x_{5B} = 1$$

$$x_{1C} + x_{2C} + x_{3C} + x_{4C} + x_{5C} = 1$$

$$x_{1D} + x_{2D} + x_{3D} + x_{3D} + x_{5D} = 1$$

$$x_{1E} + x_{2E} + x_{3E} + x_{4E} + x_{5E} = 1$$

$$x_{1F} + x_{2F} + x_{3F} + x_{4F} + x_{5F} = 1$$

$$x_{1G} + x_{2G} + x_{3G} + x_{4G} + x_{5G} = 1$$

$$x_{1H} + x_{2H} + x_{3H} + x_{4H} + x_{5H} = 1$$

$$x_{1I} + x_{2I} + x_{3I} + x_{4I} + x_{5I} = 1$$

$$x_{1J} + x_{2J} + x_{3J} + x_{4J} + x_{5J} = 1$$

$$x_{1K} + x_{2K} + x_{3K} + x_{4K} + x_{5K} = 1$$

$$x_{1L} + x_{2L} + x_{3L} + x_{4L} + x_{5L} = 1$$

$$x_{1M} + x_{2M} + x_{3M} + x_{4M} + x_{5M} = 1$$

$$x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} + x_{18} + x_{19} + x_{1A}$$
$$+ x_{1B} + x_{1C} + x_{1D} + x_{1E} + x_{1F} + x_{1G} + x_{1H} + x_{1I} + x_{1J}$$
$$+ x_{1K} + x_{1L} + x_{1M} - 22y_1 \le 0$$

Problem 5 (Continued).

Standard Mathematical Formulation

$$x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} + x_{27} + x_{28} + x_{29} + x_{2A}$$
$$+ x_{2B} + x_{2C} + x_{2D} + x_{2E} + x_{2F} + x_{2G} + x_{2H} + x_{2I} + x_{2J}$$
$$+ x_{2K} + x_{2L} + x_{2M} - 22y_2 \leq 0$$

$$x_{31} + x_{32} + x_{33} + x_{34} + x_{35} + x_{36} + x_{37} + x_{38} + x_{39} + x_{3A}$$
$$+ x_{3B} + x_{3C} + x_{3D} + x_{3E} + x_{3F} + x_{3G} + x_{3H} + x_{3I} + x_{3J}$$
$$+ x_{3K} + x_{3L} + x_{3M} - 22y_3 \leq 0$$

$$x_{41} + x_{42} + x_{43} + x_{44} + x_{45} + x_{46} + x_{47} + x_{48} + x_{49} + x_{4A}$$
$$+ x_{4B} + x_{4C} + x_{4D} + x_{4E} + x_{4F} + x_{4G} + x_{4H} + x_{4I} + x_{4J}$$
$$+ x_{4K} + x_{4L} + x_{4M} - 22y_4 \leq 0$$

$$x_{51} + x_{52} + x_{53} + x_{54} + x_{55} + x_{56} + x_{57} + x_{58} + x_{59} + x_{5A}$$
$$+ x_{5B} + x_{5C} + x_{5D} + x_{5E} + x_{5F} + x_{5G} + x5H + x5I + x5J$$
$$+ x_{5K} + x_{5L} + x_{5M} - 22y_5 \leq 0$$

Problem 5 (Hesse 1980, 271).

New Objective Function,

Smallest to Largest x_{ij} y_i Values

$$
\begin{aligned}
\min z = {} & 0x_{13} + 0x_{2L} + 0x_{39} + 0x_{44} + 0x_{5J} + 1x_{1D} + 1x_{3I} + 1x_{4I} \\
& + 2x_{1I} + 2x_{2I} + 2x_{5I} + 3x_{11} + 3x_{45} + 3x_{4D} + 4x_{25} + 4x_{41} \\
& + 5x_{15} + 5x_{2D} + 6x_{1L} + 6x_{35} + 6x_{4F} + 7x_{3D} + 7x_{3F} + 7x_{4L} \\
& + 7x_{4M} + 7x_{5D} + 9x_{21} + 9x_{4K} + 9x_{51} + 9x_{55} + 10x_{3C} + 10x_{4B} \\
& + 10x_{5E} + 11x_{31} + 11x_{3L} + 13x_{3E} + 14x_{2F} + 15x_{1K} + 15x_{2E} \\
& + 15x_{36} + 15x_{4C} + 15x_{4E} + 15x_{5L} + 16x_{1F} + 16x_{1H} + 17x_{2B} \\
& + 20x_{2K} + 20x_{3B} + 20x_{4H} + 20x_{52} + 20x_{5F} + 21x_{3M} + 25x_{1B} \\
& + 26x_{34} + 26x_{3K} + 27x_{1M} + 28x_{14} + 30x_{2H} + 30x_{2M} + 30x_{5C} \\
& + 31x_{54} + 33x_{3H} + 34x_{5K} + 36x_{24} + 37x_{5B} + 37x_{5M} + 40x_{1C} \\
& + 40x_{32} + 40x_{42} + 40x_{5G} + 40x_{5H} + 42x_{2C} + 44x_{4G} + 48x_{46} \\
& + 50x_{58} + 60x_{37} + 60x_{3G} + 60x_{43} + 60x_{56} + 63x_{23} + 65x_{47} \\
& + 70x_{12} + 70x_{1E} + 75x_{3A} + 75x_{49} + 84x_{57} + 90x_{22} + 91x_{16} \\
& + 100x_{1G} + 100x_{53} + 105x_{4A} + 105x_{5A} + 108x_{26} + 108x_{33} \\
& + 125x_{2G} + 135x_{59} + 155x_{17} + 160x_{19} + 170x_{29} + 185x_{27} \\
& + 200x_{48} + 200x_{4J} + 210x_{1A} + 240x_{38} + 240x_{3J} + 270x_{2A} \\
& + 300x_{1J} + 320x_{18} + 420x_{2J} + 440x_{28} + 100y_1 + 125y_4 \\
& + 150y_2 + 200y_3 + 250y_5
\end{aligned}
$$

Problem 5 (Hesse 1980, 271).

New Objective Function,

Smallest to Largest x_{ij} y_i Values,

<u>By Region</u>

$$
\begin{aligned}
\min z = {} & 3x_{11} + 4x_{41} + 9x_{21} + 9x_{51} + 11x_{31} + 20x_{52} + 40x_{32} + 40x_{42} \\
& + 70x_{12} + 90x_{22} + 0x_{13} + 60x_{43} + 63x_{23} + 100x_{53} + 108x_{33} \\
& + 0x_{44} + 26x_{34} + 28x_{14} + 31x_{54} + 36x_{24} + 3x_{45} + 4x_{25} + 5x_{15} \\
& + 6x_{35} + 9x_{55} + 15x_{36} + 48x_{46} + 60x_{56} + 91x_{16} + 108x_{26} \\
& + 60x_{37} + 65x_{47} + 84x_{57} + 156x_{17} + 185x_{27} + 50x_{58} + 200x_{48} \\
& + 240x_{38} + 320x_{18} + 440x_{28} + 0x_{39} + 75x_{49} + 135x_{59} + 160x_{19} \\
& + 170x_{29} + 75x_{3A} + 105x_{4A} + 105x_{5A} + 210x_{1A} + 270x_{2A} + 10x_{4B} \\
& + 17x_{2B} + 20x_{3B} + 25x_{1B} + 37x_{5B} + 10x_{3C} + 15x_{4C} + 30x_{5C} \\
& + 40x_{1C} + 42x_{2C} + 1x_{1D} + 3x_{4D} + 5x_{2D} + 7x_{3D} + 7x_{5D} + 10x_{5E} \\
& + 13x_{3E} + 15x_{2E} + 15x_{4E} + 70x_{1E} + 6x_{4F} + 7x_{3F} + 14x_{2F} + 16x_{1F} \\
& + 20x_{5F} + 40x_{5G} + 44x_{4G} + 60x_{3G} + 100x_{1G} + 125x_{2G} + 16x_{1H} \\
& + 20x_{4H} + 30x_{2H} + 33x_{3H} + 40x_{5H} + 1x_{3I} + 1x_{4I} + 2x_{1I} + 2x_{2I} \\
& + 2x_{5I} + 0x_{5J} + 200x_{4J} + 240x_{3J} + 300x_{1J} + 420x_{2J} + 9x_{4K} \\
& + 15x_{1K} + 20x_{2K} + 26x_{3K} + 34x_{5K} + 0x_{2L} + 6x_{1L} + 7x_{4L} + 11x_{3L} \\
& + 15x_{5L} + 7x_{4M} + 21x_{3M} + 27x_{1M} + 30x_{2M} + 37x_{5M} + 100y_{1} \\
& + 125y_{4} + 150y_{2} + 200y_{3} + 250y_{5}
\end{aligned}
$$

Problem 6 (Hesse 1980, 272).

Lockbox Problem

Costs	Region	Lbox 1	Lbox 2	Lbox 3	Lbox 4	Lbox 5
Variable	1	29	40	15	29	36
	2	16	56	92	30	19
	3	25	24	14	65	22
	4	23	73	16	5	42
	5	21	25	26	11	76
	6	95	16	49	22	48
	7	23	22	76	25	14
	8	17	18	41	43	27
	9	15	25	14	17	35
	A	16	40	23	14	22
Fixed		45	65	19	32	26

Problem 6 (Hesse 1980, 272).

Standard Mathematical Formulation

$$\min z = 29x_{11} + 16x_{12} + 25x_{13} + 23x_{14} + 21x_{15} + 95x_{16} + 23x_{17}$$
$$+ 17x_{18} + 15x_{19} + 16x_{1A} + 40x_{21} + 56x_{22} + 24x_{23} + 73x_{24}$$
$$+ 25x_{25} + 16x_{26} + 22x_{27} + 18x_{28} + 25x_{29} + 40x_{2A} + 15x_{31}$$
$$+ 92x_{32} + 14x_{33} + 16x_{34} + 26x_{35} + 49x_{36} + 76x_{37} + 41x_{38}$$
$$+ 14x_{39} + 23x_{3A} + 29x_{41} + 30x_{42} + 65x_{43} + 5x_{44} + 11x_{45}$$
$$+ 22x_{46} + 25x_{47} + 43x_{48} + 17x_{49} + 14x_{4A} + 36x_{51} + 19x_{52}$$
$$+ 22x_{53} + 42x_{54} + 76x_{55} + 48x_{56} + 14x_{57} + 27x_{58} + 35x_{59}$$
$$+ 22x_{5A} + 45y_1 + 65y_2 + 19y_3 + 32y_4 + 26y_5$$

s.t.

$$x_{11} + x_{21} + x_{31} + x_{41} + x_{51} = 1$$
$$x_{12} + x_{22} + x_{32} + x_{42} + x_{52} = 1$$
$$x_{13} + x_{23} + x_{33} + x_{43} + x_{53} = 1$$
$$x_{14} + x_{24} + x_{34} + x_{44} + x_{54} = 1$$
$$x_{15} + x_{25} + x_{35} + x_{45} + x_{55} = 1$$
$$x_{16} + x_{26} + x_{36} + x_{46} + x_{56} = 1$$
$$x_{17} + x_{27} + x_{37} + x_{47} + x_{57} = 1$$
$$x_{18} + x_{28} + x_{38} + x_{48} + x_{58} = 1$$
$$x_{19} + x_{29} + x_{39} + x_{49} + x_{59} = 1$$
$$x_{1A} + x_{2A} + x_{3A} + x_{4A} + x_{5A} = 1$$
$$x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} + x_{18} + x_{19} + x_{1A}$$
$$- 10y_1 \leq 0$$

Problem 6 (Continued).

Standard Mathematical Formulation

$$x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} + x_{27} + x_{28} + x_{29} + x_{2A}$$
$$- 10y_2 \leq 0$$

$$x_{31} + x_{32} + x_{33} + x_{34} + x_{35} + x_{36} + x_{37} + x_{38} + x_{39} + x_{3A}$$
$$- 10y_3 \leq 0$$

$$x_{41} + x_{42} + x_{43} + x_{44} + x_{45} + x_{46} + x_{47} + x_{48} + x_{49} + x_{4A}$$
$$- 10y_4 \leq 0$$

$$x_{51} + x_{52} + x_{53} + x_{54} + x_{55} + x_{56} + x_{57} + x_{58} + x_{59} + x_{5A}$$
$$- 10y_5 \leq 0$$

Problem 6 (Hesse 1980, 272).

New Objective Function,

Smallest to Largest x_{ij} y_i Values

$$\begin{aligned}\min z = {} & 5x_{44} + 11x_{45} + 14x_{33} + 14x_{39} + 14x_{4A} + 14x_{57} + 15x_{19} \\ & + 15x_{31} + 16x_{12} + 16x_{1A} + 16x_{26} + 16x_{34} + 17x_{18} + 17x_{49} \\ & + 18x_{28} + 19x_{52} + 21x_{15} + 22x_{27} + 22x_{46} + 22x_{53} + 22x_{5A} \\ & + 23x_{14} + 23x_{17} + 23x_{3A} + 24x_{23} + 25x_{13} + 25x_{25} + 25x_{29} \\ & + 25x_{47} + 26x_{35} + 27x_{58} + 29x_{11} + 29x_{41} + 30x_{42} + 35x_{59} \\ & + 36x_{51} + 40x_{21} + 40x_{2A} + 41x_{38} + 42x_{54} + 43x_{48} + 48x_{56} \\ & + 49x_{36} + 56x_{22} + 65x_{43} + 73x_{24} + 76x_{37} + 76x_{55} + 92x_{32} \\ & + 95x_{16} + 19y_3 + 26y_5 + 32y_4 + 45y_1 + 65y_2\end{aligned}$$

Problem 6 (Hesse 1980, 272).

New Objective Function,

Smallest to Largest x_{ij} y_i Values,

By Region

$$\begin{aligned} \min z = {} & 15x_{31} + 29x_{11} + 29x_{41} + 36x_{51} + 40x_{21} + 16x_{12} + 19x_{52} \\ & + 30x_{42} + 56x_{22} + 92x_{32} + 14x_{33} + 22x_{53} + 24x_{23} + 25x_{13} \\ & + 65x_{43} + 5x_{44} + 16x_{34} + 23x_{14} + 42x_{54} + 73x_{24} + 11x_{45} \\ & + 21x_{15} + 25x_{25} + 26x_{35} + 76x_{55} + 16x_{26} + 22x_{46} + 48x_{56} \\ & + 49x_{36} + 95x_{16} + 14x_{57} + 22x_{27} + 23x_{17} + 25x_{47} + 76x_{37} \\ & + 17x_{18} + 18x_{28} + 27x_{58} + 41x_{38} + 43x_{48} + 14x_{39} + 15x_{19} \\ & + 17x_{49} + 25x_{29} + 35x_{59} + 14x_{4A} + 16x_{1A} + 22x_{5A} + 23x_{3A} \\ & + 40x_{2A} + 19y_3 + 26y_5 + 32y_4 + 45y_1 + 65y_2 \end{aligned}$$

Problem 7 (Hesse 1980, 279).

Lockbox Problem

Costs	Region	Lbox1	Lbox2	Lbox3	Lbox4	Lbox5	Lbox6
Variable	1	15	12	26	23	83	14
	2	25	18	19	15	92	26
	3	30	25	40	16	14	35
	4	65	29	55	14	17	19
	5	5	75	73	36	23	22
	6	7	82	24	29	36	41
	7	23	95	4	83	45	36
	8	9	4	15	16	19	19
	9	16	16	10	25	23	22
	A	22	17	25	14	42	49
	B	15	26	16	36	91	76
	C	23	43	4	42	16	41
	D	19	17	16	18	21	14
Fixed		75	32	26	17	25	19

Problem 7 (Hesse 1980, 279).

Standard Mathematical Formulation

$$\min z = 15x_{11} + 25x_{12} + 30x_{13} + 65x_{14} + 5x_{15} + 7x_{16} + 23x_{17}$$
$$+ 9x_{18} + 16x_{19} + 22x_{1A} + 15x_{1B} + 23x_{1C} + 19x_{1D} + 12x_{21}$$
$$+ 18x_{22} + 25x_{23} + 29x_{24} + 75x_{25} + 82x_{26} + 95x_{27} + 4x_{28}$$
$$+ 16x_{29} + 17x_{2A} + 26x_{2B} + 43x_{2C} + 17x_{2D} + 26x_{31} + 19x_{32}$$
$$+ 40x_{33} + 55x_{34} + 73x_{35} + 24x_{36} + 4x_{37} + 15x_{38} + 10x_{39}$$
$$+ 25x_{3A} + 16x_{3B} + 4x_{3C} + 16x_{3D} + 23x_{41} + 15x_{42} + 16x_{43}$$
$$+ 14x_{44} + 36x_{45} + 29x_{46} + 83x_{47} + 16x_{48} + 25x_{49} + 14x_{4A}$$
$$+ 36x_{4B} + 42x_{4C} + 18x_{4D} + 83x_{51} + 92x_{52} + 14x_{53} + 17x_{54}$$
$$+ 23x_{55} + 36x_{56} + 45x_{57} + 19x_{58} + 23x_{59} + 42x_{5A} + 91x_{5B}$$
$$+ 16x_{5C} + 21x_{5D} + 14x_{61} + 26x_{62} + 35x_{63} + 19x_{64} + 22x_{65}$$
$$+ 41x_{66} + 36x_{67} + 19x_{68} + 22x_{69} + 49x_{6A} + 76x_{6B} + 41x_{6C}$$
$$+ 14x_{6D} + 75y_1 + 32y_2 + 26y_3 + 17y_4 + 25y_5 + 19y_6$$

s.t.

$$x_{11} + x_{21} + x_{31} + x_{41} + x_{51} + x_{61} = 1$$
$$x_{12} + x_{22} + x_{32} + x_{42} + x_{52} + x_{62} = 1$$
$$x_{13} + x_{23} + x_{33} + x_{43} + x_{53} + x_{63} = 1$$
$$x_{14} + x_{24} + x_{34} + x_{44} + x_{54} + x_{64} = 1$$
$$x_{15} + x_{25} + x_{35} + x_{45} + x_{55} + x_{65} = 1$$
$$x_{16} + x_{26} + x_{36} + x_{46} + x_{56} + x_{66} = 1$$
$$x_{17} + x_{27} + x_{37} + x_{47} + x_{57} + x_{67} = 1$$
$$x_{18} + x_{28} + x_{38} + x_{48} + x_{58} + x_{68} = 1$$
$$x_{19} + x_{29} + x_{39} + x_{49} + x_{59} + x_{69} = 1$$

Problem 7 (Continued).

Standard Mathematical Formulation

$$x_{1A} + x_{2A} + x_{3A} + x_{4A} + x_{5A} + x_{6A} = 1$$

$$x_{1B} + x_{2B} + x_{3B} + x_{4B} + x_{5B} + x_{6B} = 1$$

$$x_{1C} + x_{2C} + x_{3C} + x_{4C} + x_{5C} + x_{6C} = 1$$

$$x_{1D} + x_{2D} + x_{3D} + x_{4D} + x_{5D} + x_{6D} = 1$$

$$x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} + x_{18} + x_{19} + x_{1A} + x_{1B} + x_{1C} + x_{1D} - 13y_1 \leq 0$$

$$x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} + x_{27} + x_{28} + x_{29} + x_{2A} + x_{2B} + x_{2C} + x_{2D} - 13y_2 \leq 0$$

$$x_{31} + x_{32} + x_{33} + x_{34} + x_{35} + x_{36} + x_{37} + x_{38} + x_{39} + x_{3A} + x_{3B} + x_{3C} + x_{3D} - 13y_3 \leq 0$$

$$x_{41} + x_{42} + x_{43} + x_{44} + x_{45} + x_{46} + x_{47} + x_{48} + x_{49} + x_{4A} + x_{4B} + x_{4C} + x_{4D} - 13y_4 \leq 0$$

$$x_{51} + x_{52} + x_{53} + x_{54} + x_{55} + x_{56} + x_{57} + x_{58} + x_{59} + x_{5A} + x_{5B} + x_{5C} + x_{5D} - 13y_5 \leq 0$$

$$x_{61} + x_{62} + x_{63} + x_{64} + x_{65} + x_{66} + x_{67} + x_{68} + x_{69} + x_{6A} + x_{6B} + x_{6C} + x_{6D} - 13y_6 \leq 0$$

Problem 7 (Hesse 1980, 279).

New Objective Function,

Smallest to Largest x_{ij} y_i Values

$$\begin{aligned}
\min z = {} & 4x_{28} + 4x_{37} + 4x_{3C} + 5x_{15} + 7x_{16} + 9x_{18} + 10x_{39} + 12x_{21} \\
& + 14x_{44} + 14x_{4A} + 14x_{53} + 14x_{61} + 14x_{6C} + 15x_{11} + 15x_{1B} \\
& + 15x_{38} + 15x_{42} + 16x_{19} + 16x_{29} + 16x_{3B} + 16x_{3D} + 16x_{43} \\
& + 16x_{48} + 16x_{5C} + 17x_{2A} + 17x_{2D} + 17x_{54} + 18x_{22} + 18x_{4D} \\
& + 19x_{1D} + 19x_{32} + 19x_{58} + 19x_{64} + 19x_{68} + 21x_{5D} + 22x_{1A} \\
& + 22x_{65} + 22x_{69} + 23x_{17} + 23x_{1C} + 23x_{41} + 23x_{55} + 23x_{59} \\
& + 24x_{36} + 25x_{12} + 25x_{23} + 25x_{3A} + 25x_{49} + 26x_{2B} + 26x_{31} \\
& + 26x_{62} + 29x_{24} + 29x_{46} + 30x_{13} + 35x_{63} + 36x_{45} + 36x_{4B} \\
& + 36x_{56} + 36x_{67} + 40x_{33} + 41x_{66} + 41x_{6C} + 42x_{4C} + 42x_{5A} \\
& + 43x_{2C} + 45x_{57} + 49x_{6A} + 55x_{34} + 65x_{14} + 73x_{35} + 75x_{25} \\
& + 76x_{6B} + 82x_{26} + 83x_{47} + 83x_{51} + 91x_{5B} + 92x_{52} + 95x_{27} \\
& + 17y_4 + 19y_6 + 25y_5 + 26y_3 + 32y_2 + 75y_1
\end{aligned}$$

Problem 7 (Hesse 1980, 279).

New Objective Function,

Smallest to Largest x_{ij} y_i Values,

By Region

$$\begin{aligned}\min z = {} & 12x_{21} + 14x_{61} + 15x_{11} + 23x_{41} + 26x_{31} + 83x_{51} + 15x_{42} \\ & + 18x_{22} + 19x_{32} + 25x_{12} + 26x_{62} + 92x_{52} + 14x_{53} + 16x_{43} \\ & + 25x_{23} + 30x_{13} + 35x_{63} + 40x_{33} + 14x_{44} + 17x_{54} + 19x_{64} \\ & + 29x_{24} + 55x_{34} + 65x_{14} + 5x_{15} + 22x_{65} + 23x_{55} + 36x_{45} \\ & + 73x_{35} + 75x_{25} + 7x_{16} + 24x_{36} + 29x_{46} + 36x_{56} + 41x_{66} \\ & + 82x_{26} + 4x_{37} + 23x_{17} + 36x_{67} + 45x_{57} + 83x_{47} + 95x_{27} \\ & + 4x_{28} + 9x_{18} + 15x_{38} + 16x_{48} + 19x_{58} + 19x_{68} + 10x_{39} \\ & + 16x_{19} + 16x_{29} + 22x_{69} + 23x_{59} + 25x_{49} + 14x_{4A} + 17x_{2A} \\ & + 22x_{1A} + 25x_{3A} + 42x_{5A} + 49x_{6A} + 15x_{1B} + 16x_{3B} + 26x_{2B} \\ & + 36x_{4B} + 76x_{6B} + 91x_{5B} + 4x_{3C} + 16x_{5C} + 23x_{1C} + 41x_{6C} \\ & + 42x_{4C} + 43x_{2C} + 14x_{6D} + 16x_{3D} + 17x_{2D} + 18x_{4D} + 19x_{1D} \\ & + 21x_{5D} + 17y_4 + 19y_6 + 25y_5 + 26y_3 + 32y_2 + 75y_1\end{aligned}$$

Problem 8 (Winston 1987, 372).

Lockbox Problem

Costs	Region	Lockbox 1	Lockbox 2	Lockbox 3	Lockbox 4
Variable	1	28	84	112	112
	2	60	20	50	50
	3	96	60	24	60
	4	64	40	40	16
Fixed		50	50	50	50

Problem 8 (Winston 1987, 372).

Standard Mathematical Formulation

$$\min z = 28x_{11} + 60x_{12} + 96x_{13} + 64x_{14} + 84x_{21} + 20x_{22} + 60x_{23}$$
$$+ 40x_{24} + 112x_{31} + 50x_{32} + 24x_{33} + 40x_{34} + 112x_{41} + 50x_{42}$$
$$+ 60x_{43} + 16x_{44} + 50y_1 + 50y_2 + 50y_3 + 50y_4$$

s.t.

$$x_{11} + x_{21} + x_{31} + x_{41} = 1$$
$$x_{12} + x_{22} + x_{32} + x_{42} = 1$$
$$x_{13} + x_{23} + x_{33} + x_{43} = 1$$
$$x_{14} + x_{24} + x_{34} + x_{44} = 1$$
$$x_{11} + x_{12} + x_{13} + x_{14} - 4y_1 \leq 0$$
$$x_{21} + x_{22} + x_{23} + x_{24} - 4y_2 \leq 0$$
$$x_{31} + x_{32} + x_{33} + x_{34} - 4y_3 \leq 0$$
$$x_{41} + x_{42} + x_{43} + x_{44} - 4y_4 \leq 0$$

Problem 8 (Winston 1987, 372).

New Objective Function,

Smallest to Largest x_{ij} y_i Values

$$\min z = 16x_{44} + 20x_{22} + 24x_{33} + 28x_{11} + 40x_{24} + 40x_{34} + 50x_{32} + 50x_{42} + 60x_{12} + 60x_{23} + 60x_{43} + 64x_{14} + 84x_{21} + 96x_{13} + 112x_{31} + 112x_{41} + 50y_1 + 50y_2 + 50y_3 + 50y_4$$

Problem 8 (Winston 1987, 372).

New Objective Function,

Smallest to Largest x_{ij} y_i Values,

<u>By Region</u>

$$\min z = 28x_{11} + 84x_{21} + 112x_{31} + 112x_{41} + 20x_{22} + 50x_{32} + 50x_{42} + 60x_{12} + 24x_{33} + 60x_{23} + 60x_{43} + 96x_{13} + 16x_{44} + 40x_{24} + 40x_{34} + 64x_{14} + 50y_1 + 50y_2 + 50y_3 + 50y_4$$

Problem 9 (Erlenkotter 1978, 1000).

Lockbox Problem

Costs	Region	Lbox 1	Lbox 2	Lbox 3	Lbox 4	Lbox 5
Variable	1	120	210	180	210	170
	2	180	M	190	190	150
	3	100	150	110	150	110
	4	M	240	195	180	150
	5	60	55	50	65	70
	6	M	210	M	120	195
	7	180	110	M	160	200
	8	M	165	195	120	M
Fixed		100	70	60	110	80

Problem 9 (Erlenkotter 1978, 1000).

Standard Mathematical Formulation

$$
\begin{aligned}
\min z = {} & 120x_{11} + 180x_{12} + 100x_{13} + Mx_{14} + 60x_{15} + Mx_{16} + 180x_{17} \\
& + Mx_{18} + 210x_{21} + Mx_{22} + 150x_{23} + 240x_{24} + 55x_{25} + 210x_{26} \\
& + 110x_{27} + 165x_{28} + 180x_{31} + 190x_{32} + 110x_{33} + 195x_{34} + 50x_{35} \\
& + Mx_{36} + Mx_{37} + 195x_{38} + 210x_{41} + 190x_{42} + 150x_{43} + 180x_{44} \\
& + 65x_{45} + 120x_{46} + 160x_{47} + 120x_{48} + 170x_{51} + 150x_{52} + 110x_{53} \\
& + 150x_{54} + 70x_{55} + 195x_{56} + 200x_{57} + Mx_{58} + 100y_1 + 70y_2 \\
& + 60y_3 + 110y_4 + 80y_5
\end{aligned}
$$

$$
\begin{aligned}
\text{s.t.}\quad & x_{11} + x_{21} + x_{31} + x_{41} + x_{51} = 1 \\
& x_{12} + x_{22} + x_{32} + x_{42} + x_{52} = 1 \\
& x_{13} + x_{23} + x_{33} + x_{43} + x_{53} = 1 \\
& x_{14} + x_{24} + x_{34} + x_{44} + x_{54} = 1 \\
& x_{15} + x_{25} + x_{35} + x_{45} + x_{55} = 1 \\
& x_{16} + x_{26} + x_{36} + x_{46} + x_{56} = 1 \\
& x_{17} + x_{27} + x_{37} + x_{47} + x_{57} = 1 \\
& x_{18} + x_{28} + x_{38} + x_{48} + x_{58} = 1 \\
& x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} + x_{18} - 8y_1 \leq 0 \\
& x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} + x_{27} + x_{28} - 8y_2 \leq 0 \\
& x_{31} + x_{32} + x_{33} + x_{34} + x_{35} + x_{36} + x_{37} + x_{38} - 8y_3 \leq 0 \\
& x_{41} + x_{42} + x_{43} + x_{44} + x_{45} + x_{46} + x_{47} + x_{48} - 8y_4 \leq 0 \\
& x_{51} + x_{52} + x_{53} + x_{54} + x_{55} + x_{56} + x_{57} + x_{58} - 8y_5 \leq 0
\end{aligned}
$$

Problem 9 (Erlenkotter 1978, 1000).

New Objective Function,

Smallest to Largest x_{ij} y_i Values

$$\begin{aligned} \min z = {} & 50x_{35} + 55x_{25} + 60x_{15} + 65x_{45} + 70x_{55} + 100x_{13} + 110x_{27} \\ & + 110x_{33} + 110x_{53} + 120x_{11} + 120x_{46} + 120x_{48} + 150x_{23} \\ & + 150x_{43} + 150x_{52} + 150x_{54} + 160x_{47} + 165x_{28} + 170x_{51} \\ & + 180x_{12} + 180x_{17} + 180x_{31} + 180x_{44} + 190x_{32} + 190x_{42} \\ & + 195x_{34} + 195x_{38} + 195x_{56} + 200x_{57} + 210x_{21} + 210x_{26} \\ & + 210x_{41} + 240x_{24} + Mx_{14} + Mx_{16} + Mx_{18} + Mx_{22} + Mx_{36} \\ & + Mx_{37} + Mx_{58} + 60y_3 + 70y_2 + 80y_5 + 100y_1 + 110y_4 \end{aligned}$$

Problem 9 (Erlenkotter 1978, 1000).

New Objective Function,

Smallest to Largest x_{ij} y_i Values,

By Region

$$\begin{aligned} \min z = {} & 120x_{11} + 170x_{51} + 180x_{31} + 210x_{21} + 210x_{41} + 150x_{52} \\ & + 180x_{12} + 190x_{32} + 190x_{42} + Mx_{22} + 100x_{13} + 110x_{33} + 110x_{53} \\ & + 150x_{23} + 150x_{43} + 150x_{54} + 180x_{44} + 195x_{34} + 240x_{24} + Mx_{14} \\ & + 50x_{35} + 55x_{25} + 60x_{15} + 65x_{45} + 70x_{55} + 120x_{46} + 195x_{56} \\ & + 210x_{26} + Mx_{16} + Mx_{36} + 110x_{27} + 160x_{47} + 180x_{17} + 200x_{57} \\ & + Mx_{37} + 120x_{48} + 165x_{28} + 195x_{38} + Mx_{18} + Mx_{58} + 60y_3 + 70y_2 \\ & + 80y_5 + 100y_1 + 110y_4 \end{aligned}$$

Problem 10 (Salkin 1975, 492).

Lockbox Problem

Costs	Region	Lockbox 1	Lockbox 2	Lockbox 3	Lockbox 4
Variable	1	400	600	100	100
	2	900	90	180	1170
	3	560	1980	1210	220
	4	960	240	2530	1320
	5	450	560	750	1050
Fixed		110	130	140	160

Problem 10 (Salkin 1975, 492).

Standard Mathematical Formulation

$$\min z = 400x_{11} + 900x_{12} + 560x_{13} + 960x_{14} + 450x_{15} + 600x_{21}$$
$$+ 90x_{22} + 1980x_{23} + 240x_{24} + 560x_{25} + 100x_{31} + 180x_{32}$$
$$+ 1210x_{33} + 2530x_{34} + 750x_{35} + 100x_{41} + 1170x_{42} + 220x_{43}$$
$$+ 1320x_{44} + 1050x_{45} + 110y_1 + 130y_2 + 140y_3 + 160y_4$$

s.t.
$$x_{11} + x_{21} + x_{31} + x_{41} = 1$$
$$x_{12} + x_{22} + x_{32} + x_{42} = 1$$
$$x_{13} + x_{23} + x_{33} + x_{43} = 1$$
$$x_{14} + x_{24} + x_{34} + x_{44} = 1$$
$$x_{15} + x_{25} + x_{35} + x_{45} = 1$$
$$x_{11} + x_{12} + x_{13} + x_{14} + x_{15} - 5y_1 \leq 0$$
$$x_{21} + x_{22} + x_{23} + x_{24} + x_{25} - 5y_2 \leq 0$$
$$x_{31} + x_{32} + x_{33} + x_{34} + x_{35} - 5y_3 \leq 0$$
$$x_{41} + x_{42} + x_{43} + x_{44} + x_{45} - 5y_4 \leq 0$$

Problem 10 (Salkin 1975, 492).

New Objective Function,

Smallest to Largest x_{ij} y_i Values

$$\min z = 90x_{22} + 100x_{31} + 100x_{41} + 180x_{32} + 220x_{43} + 240x_{24}$$
$$+ 400x_{11} + 450x_{15} + 560x_{13} + 560x_{25} + 600x_{21} + 750x_{35}$$
$$+ 900x_{12} + 960x_{14} + 1050x_{45} + 1170x_{42} + 1210x_{33} + 1320x_{44}$$
$$+ 1980x_{23} + 2530x_{34} + 110y_1 + 130y_2 + 140y_3 + 160y_4$$

Problem 10 (Salkin 1975, 492).

New Objective Function,

Smallest to Largest x_{ij} y_i Values,

By Region

$$\min z = 100x_{31} + 100x_{41} + 400x_{11} + 600x_{21} + 90x_{22} + 180x_{32}$$
$$+ 900x_{12} + 1170x_{42} + 220x_{43} + 560x_{13} + 1210x_{33} + 1980x_{23}$$
$$+ 240x_{24} + 960x_{14} + 1320x_{44} + 2530x_{34} + 450x_{15} + 560x_{25}$$
$$+ 750x_{35} + 1050x_{45} + 110y_1 + 130y_2 + 140y_3 + 160y_4$$

Problem 11 (Salkin 1975, 494).

Lockbox Problem

Costs	Region	Lockbox 1	Lockbox 2	Lockbox 3
Variable	1	200	100	300
	2	200	150	100
	3	1050	1200	750
	4	1200	1400	2000
Fixed		450	400	500

Problem 11 (Salkin 1975, 494).

Standard Mathematical Formulation

$$\min z = 200x_{11} + 200x_{12} + 1050x_{13} + 1200x_{14} + 100x_{21} + 150x_{22}$$
$$+ 1200x_{23} + 1400x_{24} + 300x_{31} + 100x_{32} + 750x_{33} + 2000x_{34}$$
$$+ 450y_1 + 400y_2 + 500y_3$$

s.t.

$$x_{11} + x_{21} + x_{31} = 1$$
$$x_{12} + x_{22} + x_{32} = 1$$
$$x_{13} + x_{23} + x_{33} = 1$$
$$x_{14} + x_{24} + x_{34} = 1$$
$$x_{11} + x_{12} + x_{13} + x_{14} - 4y_1 \leq 0$$
$$x_{21} + x_{22} + x_{23} + x_{24} - 4y_2 \leq 0$$
$$x_{31} + x_{32} + x_{33} + x_{34} - 4y_3 \leq 0$$

Problem 11 (Salkin 1975, 494).

New Objective Function,

Smallest to Largest x_{ij} y_i Values

$$\min z = 100x_{21} + 100x_{32} + 150x_{22} + 200x_{11} + 200x_{12} + 300x_{31} + 750x_{33} + 1050x_{13} + 1200x_{14} + 1200x_{23} + 1400x_{24} + 2000x_{34} + 400y_2 + 450y_1 + 500y_3$$

Problem 11 (Salkin 1975, 494).

New Objective Function,

Smallest to Largest x_{ij} y_i Values,

By Region

$$\min z = 100x_{21} + 200x_{11} + 300x_{31} + 100x_{32} + 150x_{22} + 200x_{12} + 750x_{33} + 1050x_{13} + 1200x_{23} + 1200x_{14} + 1400x_{24} + 2000x_{34} + 400y_2 + 450y_1 + 500y_3$$

Problem 12 (Salkin 1975, 495).

Lockbox Problem

Costs	Region	Lockbox 1	Lockbox 2	Lockbox 3	Lockbox 4
Variable	1	100	110	90	120
	2	150	140	160	130
	3	80	70	100	110
	4	100	90	80	60
Fixed		800	600	700	500

Problem 12 (Salkin 1975, 495).

Standard Mathematical Formulation

$$\min z = 100x_{11} + 150x_{12} + 80x_{13} + 100x_{14} + 110x_{21} + 140x_{22} + 70x_{23}$$
$$+ 90x_{24} + 90x_{31} + 160x_{32} + 100x_{33} + 80x_{34} + 120x_{41} + 130x_{42}$$
$$+ 110x_{43} + 60x_{44} + 800y_1 + 600y_2 + 700y_3 + 500y_4$$

s.t.

$$x_{11} + x_{21} + x_{31} + x_{41} = 1$$
$$x_{12} + x_{22} + x_{32} + x_{42} = 1$$
$$x_{13} + x_{23} + x_{33} + x_{43} = 1$$
$$x_{14} + x_{24} + x_{34} + x_{44} = 1$$
$$x_{11} + x_{12} + x_{13} + x_{14} - 4y_1 \le 0$$
$$x_{21} + x_{22} + x_{23} + x_{24} - 4y_2 \le 0$$
$$x_{31} + x_{32} + x_{33} + x_{34} - 4y_3 \le 0$$
$$x_{41} + x_{42} + x_{43} + x_{44} - 4y_4 \le 0$$

Problem 12 (Salkin 1975, 495).

New Objective Function,

Smallest to Largest x_{ij} y_i Values

$$\min z = 60x_{44} + 70x_{23} + 80x_{13} + 80x_{34} + 90x_{24} + 90x_{31} + 100x_{11} + 100x_{14} + 100x_{33} + 110x_{21} + 110x_{43} + 120x_{41} + 130x_{42} + 140x_{22} + 150x_{12} + 160x_{32} + 500y_4 + 600y_2 + 700y_3 + 800y_1$$

Problem 12 (Salkin 1975, 495).

New Objective Function,

Smallest to Largest x_{ij} y_i Values,

By Region

$$\min z = 90x_{31} + 100x_{11} + 110x_{21} + 120x_{41} + 130x_{42} + 140x_{22} + 150x_{12} + 160x_{32} + 70x_{23} + 80x_{13} + 100x_{33} + 110x_{43} + 60x_{44} + 80x_{34} + 90x_{24} + 100x_{14} + 500y_4 + 600y_2 + 700y_3 + 800y_1$$