Appendices

I	Derivation of the Lagrange Multiplier Method
II	Problem Formulation: Some Guidelines
III	Interpretation of Results: Some Guidelines
IV	Introduction to Solving Linear Programs using LINDO
V	The Simplex Algorithm
VI	Integer and Integer-Linear Programming: A Branch-and-Bound Algorithm
VII	Selected Bibliography

A Derivation of the Lagrange Multiplier Method

Problem:

Max $Z = f(X_1, X_2)$

Subject To (S.T.):

$g(X_1, X_2) = b$, b is a constant

[If constraint could be reformulated as: $X_2 = g_1(X_1)$, then the constraint could be substituted into $f(X_1, X_2)$ and solved as unconstrained optimization. (This is often not possible or expedient.

First-order Conditions (2 of 'em)

To Max Z, subject to the constraint, a first-order condition is:

$$(1) \quad \boxed{\frac{df}{dX_1} = 0 = \frac{\partial f}{\partial X_1} + \frac{\partial f}{\partial X_2}\frac{dX_2}{dX_1}}$$

The second part of the derivative accounts for the second decision variable; $\left(\frac{dX_2}{dX_1}\right) = 0$ if there is no constraint.)

Another first-order condition comes from the constraint:

$$(2) \quad \boxed{\frac{dg}{dX_1} = 0 = \frac{\partial g}{\partial X_1} + \frac{\partial g}{\partial X_2}\frac{dX_2}{dX_1}}$$

This keeps us on the equality constraint; $g(X_1, X_2)$ cannot vary (it must = b) as X_1 varies.

Re-arranging both equations to isolate $\frac{dX_2}{dX_1}$:

$$(3) \quad \boxed{\frac{dX_2}{dX_1} = \frac{-\partial f/\partial X_1}{\partial f/\partial X_2} - \frac{-\partial g/\partial X_1}{\partial g/\partial X_2}}$$

Equation (3) contains two 1st order conditions and the constraint $g(X_1, X_2) = b$ are now sufficient to find the extreme points of the problem. But they will often be awkward.

Modifying (3), so derivatives on each side are w.r.t. only one decision variable:

$$(4) \quad \boxed{\frac{-\partial f/\partial X_1}{\partial g/\partial X_1} = \frac{-\partial f/\partial X_2}{\partial g/\partial X_2} = \lambda} \quad \boxed{\approx \frac{-\partial f}{\partial b}}$$

Let $\lambda \equiv$ Lagrange Multiplier

Breaking Equation (4) into two separage equations, both =λ, and re-arranging gives the final first-order conditions:

$$\boxed{0 = \frac{\partial f}{\partial X_1} + \lambda \frac{\partial g}{\partial X_1}} \tag{5}$$

and

$$\boxed{0 = \frac{\partial f}{\partial X_2} + \lambda \frac{\partial g}{\partial X_2}} \tag{6}$$

To this must be added the constraint

$$\boxed{g\,(X_1, X_2) = b} \tag{7}$$

These are the same first-order conditions we get from maximizing or minimizing the Lagrangian:

$$L = f(X_1, X_2) + \lambda\,[g(X_1, X_2) - b]$$

That's why the Lagrange multiplier method works. Derivation is from: Donald A. Pierre (1969), Optimization Theory With Applications.

PROBLEM FORMULATION: SOME GUIDELINES

PURPOSE OF FORMULATION

The formulation of a problem organizes the problem into a mathematical form that can be understood in terms of the real problem and, perhaps, solved by mathematical or numerical methods. Proper formulation is essential to solving the "right problem."

GUIDELINES

To formulate a problem as a mathematical program, *decisions*, *objectives*, and *constraints* must be identified and stated in mathematical terms.

SOME STEPS FOR PROBLEM FORMULATION

Part I: Organize the problem into words by answering the following questions in words:

Step 1) What decisions must be made? (E.g. Pipe diameters must be specified.)

Step 2) What is the objective? What characterises a "good" decision? (E.g., A "good" decision has a low total cost.)

Step 3) What constraints restrict the decisions that can be made? (E.g., the pipe must convey a constant flow of 10 cfs.)

Part II: Restate the decisions, objective(s), and constraints in mathematical terms.

Step 4) Restate the decisions as decision variables. (E.g., D = pipe diameter (feet)) Define these decision variables completely with units.

Step 5) Restate the objective as an objective function or objective functions. (E.g., MIN total cost = capital cost + present value of pumping cost = f(D) + p(D); Here, you would have to specify a capital cost function of pipeline diameter f(D) and a pumping cost function of pipeline diameter p(D))

Step 6) Restate the constraints as equations or inequalities. (E.g., $Q = 10 = \pi^2 g\, D^5/(8fL)$, where L is the given pipe length and f is a (given?) friction factor)

Note: The objective function and/or constraint equations must include decision variables.

Part III: Summarize the mathematical formulation of the problem.

Step 7) State the objective function.

Step 8) State the constraints.

This should look something like:

$$\text{MINIMIZE} \quad f(D) + p(D)$$

$$\text{SUBJECT TO:} \quad Q = \pi^2 g\, D^5/(8fL).$$

Step 9) Add a list of variable definitions, including their units and specifying which variables are decision variables.

Part IV: Look over the mathematical programming formulation of the problem and see if it makes sense. Does the math represent the logic inherent in the original word problem?

PRESENTING YOUR FORMULATION

Here are some ideas.

- Use common-sense variable names. E.g., let Q_{TW} = the water flow for tomatoes in Winter.
- Use capital letters for variables and small letters for constants in the formulation.
- Organize your presentation in a systematic way.
- Explain where the objective function and each constraint come from. But try to do this in a way that isn't tedious.

INTERPRETATION OF RESULTS: SOME GUIDELINES

PURPOSES OF INTERPRETING SOLUTIONS

An engineer's results must be used by other engineers and non-engineers, most of whom will be unfamiliar with any analytical approaches used to get the results. These clients or employers require your results in an informative, but simple and straight-forward form. They also need to know any important limitations of your analysis.

SOME GUIDELINES FOR PRESENTING AND INTERPRETING RESULTS

1) Answer questions your client has. For our aggregate-mix problem, the client is likely to be a contractor or an engineer-estimator. They would want to know:

-) How many cubic yards of material should be brought from each site?
-) How much will this cost (i.e., the best value of the cost/objective function)?
-) How much will it cost if the total aggregate volume required should change?
-) How much will it cost if the specifications of the total aggregate should change?

Answer these questions and any others you think are important. Note which important questions you cannot answer. Also qualify your answers when these qualifications are important or useful.

2) Present your results and interpretations in terms familiar to the your client. Who will be using your work and how can they best understand what your analysis has to say? For our aggregate mix problem, they would want these answers in terms of words and numbers with units. Telling them, "X1 = 3,333.33" is less useful than, "3,333 yd^3 of aggregate should be taken from Site 1." For large problems, a tabular presentation is often best.

Sensitivity analysis should also be presented in a useful form. Instead of, "$\lambda_1 = 6.33$," it is much more useful to write, "The cost of changing the total amount of aggregate required is \$6.33/yd^3, if the same sand and gravel specifications are maintained. This is given by λ_1." Since many engineers are not used to sensitivity analysis, it is often useful to mention where these numbers come from.

3) Structure your results.

Often, the user first wants to know what the best solution is and how it performs on the objective function.

The user of your work might then want to know how sensitive this solution is to changes in the problem's formulation, sensitivity analysis. You might draw the reader's attention to any important highly-sensitive parameters.

You should comment on how well the real problem is represented by the optimization model.

Finally, present an overview and highlight the most important aspects of your results and interpretations.

4) Sensitivity Analysis. Use Lagrange multipliers, slack variables, new model runs, etc. to answer important questions regarding the stability of your solutions and how the solution and its cost will change with reasonable changes in the formulation. Tables and figures are also useful for illustrating interpretations and sensitivities.

5) Tables, Figures, and Text. Use tables and figures to summarize results and text to highlight specific points.

6) Be brief and concise. Don't ramble. State your results in clear terms and move on. Use tables to present large quantities of results. This takes less time to read, and less time to write.

7) As a practicing engineer, your results and interpretation are your product. This product is only useful if it is right, understandable, and helps the client make an informed decision. A correct solution alone can be useless without a good presentation and interpretation.

INTRODUCTION TO SOLVING LINEAR PROGRAMS USING LINDO

This exercise should help you get acquainted with the linear program solution package LINDO. In this exercise, you type the underlined words. Follow these commands with a return.

1) Open LINDO from the Programs menu under the Start menu.

2) You will be in an empty window called “untitled”.

3) Look under the **Help** menu, under “Contents”.
This is an introduction to the help available within the LINDO program.
Be sure to poke under the “File”>>”Take Commands” and “Reports” menus.

4) Return to the empty “untitled” window. You can use the “Window” menu to do this.

Type MIN 5X1 + 7X2

6) Type SUBJECT TO

7) Type X1 + X2 >= 10000.
8) Type -0.2X1 + 0.1X2 >= 0
9) Type 0.1X1 - 0.2X2 <= 0

10) Type END
You have entered a linear program.

11) Under the **Reports** menu, hit **Formulation**

12) LINDO will ask you how much of the formulation you want to look at. Check “All” and then “OK”.
LINDO will show you (in the Reports Window) the linear program you have just typed in. It will even clean it up and standardize the formatting.

12) Under the **Solve** menu, hit **Solve**
LINDO will now solve the LP and present some results (in the Reports Window). LINDO will then ask if you want to see more sensitivity analysis.

13) Hit “Yes”
The sensitivity analysis will be presented in the Report window.

14) Under the **File** menu, hit **Save as …**
This will save the reports window, including the formulation, solution, and sensitivity analysis. You specify the name and location of the file.

15) You can now quit LINDO.

NOTES:
a) The R.H.S. of all constraints must be only a constant.

b) Often it is easier to use a text editor (e.g., Wordpad) to write out the formulation and “Take” the file into LINDO.

THE SIMPLEX ALGORITHM

The following steps form the fundamentals of the Simplex Algorithm for solving linear programs.

1) Set up an "initial tableau."

2) Is the trial solution represented by the initial tableau optimal?

If all coefficients in Row 0 are ≥ 0, then the solution is optimal and you may STOP.

If any coefficients in Row 0 of the tableau are < 0, the solution represented by this tableau is not the optimal solution and the set of basic variables must be changed.

3) Select a new basic variable.

Select as a new basic variable the variable with the most negative coefficient in Row 0. The column for this variable is the "pivot column."

4) Select a new non-basic variable.

For each row (except Row 0 and any rows where the pivot column elements are negative), divide the RHS entry of each row by the coefficient in the column of the new basic variable.

The new non-basic variable is the old basic variable associated with the row having the minimum ratio. This row is called the "pivot row."

5) Update the tableau.

Form a new tableau which has zero for each entry in the pivot column, except for the coefficient in the pivot row. The entry in both the pivot row and pivot column must be equal to one. Use linear algebra and Gaussian elimination techniques to do this.

Replace the old basic variable with the new basic variable in the "Basic Variable" column.

This new tableau represents a new corner-point solution.

6) Is this new solution optimal?

If all coefficients in Row 0 of the new tableau are ≥ 0, then STOP; the solution is optimal.

If any coefficients in Row 0 of the tableau are < 0, the solution represented by this tableau is not the optimal solution and the set of basic variables must be changed. GO TO STEP 3.

Reading the final tableau.

The following tableau represents an optimal solution and final tableau for the problem:

MAX $2X1 + X2$
Subject To: 1) $X1 + X2 \leq 4$
2) $X2 \leq 3$

Basic Variable	Eqn. No.	Coefficient of Variable X1	X2	S1	S2	RHS
Z	0	0	1	2	0	8
X1	1	1	1	1	0	4
S2	2	0	1	0	1	3

The values of each basic variable is given by the RHS column. The value of each non-basic variable is zero. Z = 8, X1 = 4, S2 = 3, X2 = 0, S1 = 0.

The value of the Lagrange multiplier (shadow price) for each constraint is given by the coefficient in Row 0 under the associated slack variable (S1 or S2). $\lambda_1 = 2$; $\lambda_2 = 0$.

The Reduced Cost of each coefficient in the objective function is given by the coefficient in Row 0 under the associated decision variable (X1 or X2).

Can you obtain these results?

INTEGER AND INTEGER-LINEAR PROGRAMMING: A BRANCH-AND-BOUND METHOD

The following algorithm can be used to solve a mixed integer-linear program using linear programming solution methods. Consider the problem:

$$\begin{aligned} &\text{Maximize } z = 5X_1 + 3X_2 \\ &\text{Subject to:} \quad 4X_1 + 2X_2 \le 25 \\ &\qquad X_1 \le 5 \\ &\qquad X_2 \le 8 \\ &\qquad \text{Both } X_1 \text{ and } X_2 \text{ are integers.} \end{aligned}$$

Step 1: Find the optimal continuous solution. Relax the integer requirements for the decision variables and solve the resulting non-integer linear program. This gives an optimal continuous solution (OCS). If all originally integer variables have integer solutions in the OCS, then STOP; this is the optimal solution. Otherwise, continue with Step 2.

For the problem above, the solution is $X_1 = 2.25$, $X_2 = 8$, $z = 35.25$, so this is not the optimal solution to the integer programming problem.

Step 2: Branch. Divide the problem into two sub-problems, searching for good integer-valued solutions. For our problem, X_1 has a fractional value in Step 1 of 2.25. We produce the two sub-problems by adding new constraints on the non-integer value which keep it from its current non-integer value. Here, since $X_1 = 2.25$ above, we add the constraints $X_1 \le 2$ and $X_1 \ge 3$ to give the following two sub-problems. This is called branching.

Sub-problem 1:

$$\begin{aligned} &\text{Maximize } z = 5X_1 + 3X_2 \\ &\text{Subject to:} \quad 4X_1 + 2X_2 \le 25 \\ &\qquad X_1 \le 5 \text{ (now redundant)} \\ &\qquad X_2 \le 8 \\ &\qquad X_1 \le 2 \end{aligned}$$

Sub-problem 2:

$$\begin{aligned} &\text{Maximize } z = 5X_1 + 3X_2 \\ &\text{Subject to:} \quad 4X_1 + 2X_2 \le 25 \\ &\qquad X_1 \le 5 \\ &\qquad X_2 \le 8 \\ &\qquad X_1 \ge 3 \end{aligned}$$

Step 3: Solve the branched problems. When these sub-problems are solved by linear programming, the solutions are compared.

Sub-problem 1: $X_1 = 2$, $X_2 = 8$, $Z_1 = 34$

Sub-problem 2: $X_1 = 3$, $X_2 = 6.5$, $Z_2 = 34.5$

Step 4: Set the bound. The best solution to the sub-problem which satisfies all integer requirements is called the upper (or lower) bound. In this case, the sub-problem 1 solution, $z = 34$, is the lower bound of the optimal solution.

If no sub-problem solution satisfies the integer conditions, then further branching and solution is required until a bound can be set. (If a bound has already been set, any new bound must have a better value than the existing bound.)

Step 5: Continued branching. Branch on sub-problems that both do not satisfy integer conditions and have superior solutions. In this case the solution to sub-problem 2 both does not satisfy the integer conditions and has a superior solution to the bound (bound = 34, $Z_2 = 34.5$). If the solutions to the all other sub-problems are inferior to the bound, then STOP and the solution that sets the bound is the optimal solution which satisfies all constraints, including the integer conditions. Otherwise, GO TO Step 2.

Continuing with the solution to the example:
Following Step 2, continued branching for this problem (Sub-problem 2) gives the following two sub-problems:

Subproblem 2a:	*Subproblem 2b:*
Maximize $z = 5 X_1 + 3 X_2$	Maximize $z = 5 X_1 + 3 X_2$
Subject to: $4 X_1 + 2 X_2 \leq 25$	Subject to: $4 X_1 + 2 X_2 \leq 25$
$X_1 \leq 5$	$X_1 \leq 5$
$X_2 \leq 8$ (now redundant)	$X_2 \leq 8$
$X_1 \geq 3$	$X_1 \geq 3$
$X_2 \leq 6$ (new constraint)	$X_2 \geq 7$ (new constraint)

Repeating Step 3, the solutions to these new sub-problems are:
Sub-problem 2a: $X_1 = 3.25$, $X_2 = 6$, $Z_{2a} = 34.25$
Sub-problem 2b: infeasible

Step 4: Since neither of the new solutions both satisfies the integer conditions and has a better objective function value than the existing bound (existing bound = 34), the old bound remains.

Step 5: Since the objective function value for the solution to sub-problem 2a is superior to the bound, there remains a possibility that there is an integer solution better than the current bound (= 34). So branching on sub-problem 2a continues, using Step 2.

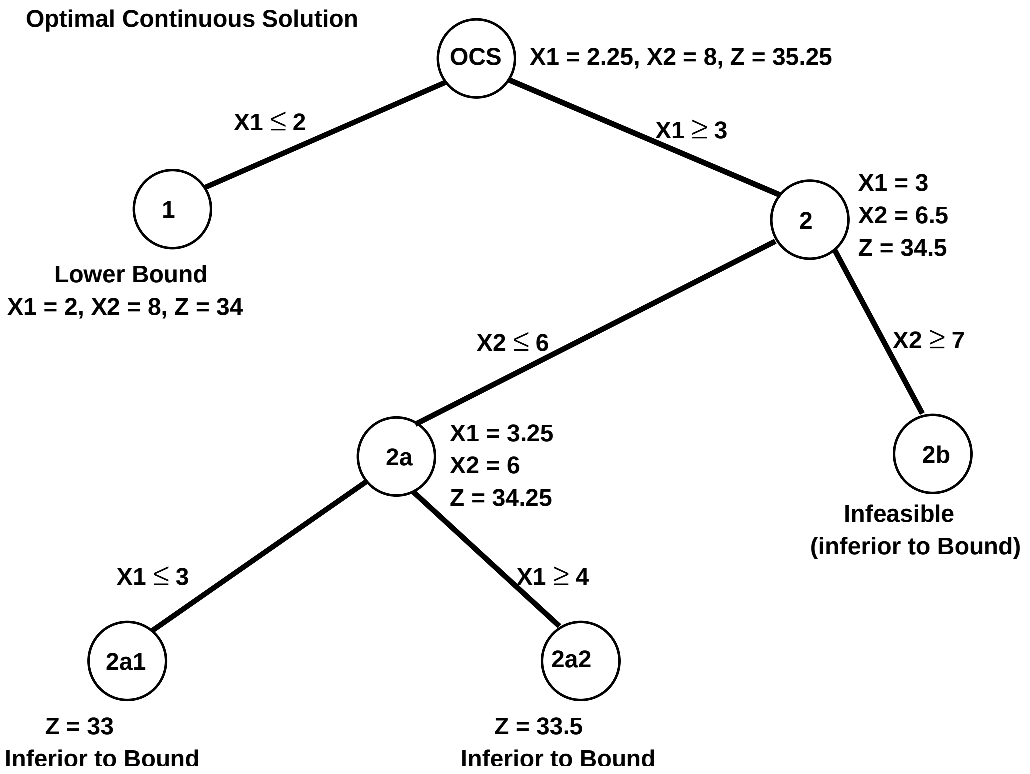
Step 2: Branching on sub-problem 2a gives the following new sub-problems:

Subproblem 2a1:	*Subproblem 2a2:*
Maximize $z = 5 X_1 + 3 X_2$	Maximize $z = 5 X_1 + 3 X_2$
Subject to: $4 X_1 + 2 X_2 \leq 25$	Subject to: $4 X_1 + 2 X_2 \leq 25$
$X_1 \leq 5$	$X_1 \leq 5$
$X_2 \leq 8$ (now redundant)	$X_2 \leq 8$ (now redundant)
$X_1 \geq 3$	$X_1 \geq 3$ (now redundant)
$X_2 \leq 6$	$X_2 \leq 6$
$X_1 \leq 3$ (new constraint)	$X_1 \geq 4$ (new constraint)

Repeating Step 3, the solutions to these new sub-problems are:
Sub-problem 2a1: $X_1 = 3$, $X_2 = 6$, $Z_{2a1} = 33$
Sub-problem 2a2: $X_1 = 4$, $X_2 = 4.5$, $Z_{2a2} = 33.5$

Steps 4 and 5: Since neither solution to the new sub-problem is superior to the bound, there is no hope for an integer solution better than the bounded solution (Z = 34). (Since further branching adds constraints, and a solution is never improved by adding constraints, new sub-problems will never produce better solutions than their parent problems.) Therefore, we can STOP, knowing the optimal solution is that of the current bound, $Z = 34$, $X_1 = 2$, $X_2 = 8$.

Graphically, in solving the example we have created the following tree from the branches, with each branch terminating when we reach an inferior solution or an upper bound.


Optimal Continuous Solution
OCS
X1 = 2.25, X2 = 8, Z = 35.25
X1 ≤ 2
X1 ≥ 3
1
Lower Bound
X1 = 2, X2 = 8, Z = 34
2
X1 = 3
X2 = 6.5
Z = 34.5
X2 ≤ 6
X2 ≥ 7
2a
X1 = 3.25
X2 = 6
Z = 34.25
2b
Infeasible
(inferior to Bound)
X1 ≤ 3
X1 ≥ 4
2a1
Z = 33
Inferior to Bound
2a2
Z = 33.5
Inferior to Bound

Selected Bibliography

Here are some useful references for operations research practitioners. I used many of these books in putting together the notes for this class. If you read all of these books, you might discover some test or homework problems.

Books on Methods:
Sherali, Hanif D., C.M. Shetty, Mokhtar S. Bazaraa (1993), *Nonlinear Programming Theory and Algorithms*, John Wiley and Sons, N.Y.

Bertsekas, Dimitri P. (1999). *Nonlinear Programming,* Athena Publishing.

Dantzig, George B. (1963), *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ, 627 pp.
The author is the inventor of the simplex method. It is a very good, if somewhat technical book on the ins and outs of the linear programming.

Ecker, Joseph G. and Michael Kupferschmid (1988), *Introduction to Operations Research*, John Wiley and Sons, N.Y., 509 pp.
A more mathematical, and consequently briefer and more straight-forward look at the most common operations research methods.

Goicoechea, Ambrose, Don R. Hansen, and Lucien Duckstein (1982), *Multiobjective Decision Analysis with Engineering and Business Applications*, John Wiley and Sons, N.Y., 519 pp.
An OK book on multi-objective optimization and decision-making methods.

Hillier, Frederick S. and Gerald J. Lieberman (1990), *Introduction to Operations Research*, McGraw-Hill Publishing Co., N.Y., 954 pp.
An excellent book, covering many topics in operations research with a minimum of mathematics. Not quite as comprehensive as Wagner's book, but with more exhaustive explanations.

Intriligator, Michael D. (1971), *Mathematical Optimization and Economic Theory*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 508 pp.
A nice treatment of many optimization methods, with a nice introduction to optimal control theory. Written from the perspective of applied economic theory.

Taha, Hamdy A. (1992), *Operations Research: An Introduction*, MacMillan, N.Y., 822 pp.
Another fine operations research text.

Wagner, Harvey M. (1975), *Principles of Operations Research*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1039 pp. Still one of the most comprehensive books on optimization methods.

Civil Engineering Applications:
deNeufville, Richard (1990), *Applied Systems Analysis: Engineering Planning and Technology Management*, McGraw-Hill Publishing Co., N.Y., 470 pp. A nice mix of theory and applications.

Hendrickson, Chris and Tung Au (1989), *Project Management for Construction*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 537 pp.
An excellent book on project management with some interesting applications of optimization.

Loucks, Daniel P., Jery R. Stedinger, and Douglas A. Haith (1981), *Water Resources Systems Planning and Analysis*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 559 pp.
An excellent overview of applications to water resource problems.

Meredith, Dale D., Kam W. Wong, Ronald W. Woodhead, and Robert H. Wortman (1973), *Design and Planning of Engineering Systems*, Prentice-Hall, Englewood Cliffs, NJ, 393 pp.
A nice overview of civil engineering applications of systems analysis.

Ossenbruggen, Paul J. (1984), *Systems Analysis for Civil Engineers*, John Wiley and Sons, N.Y., 571 pp.
A very nice book both introducing the theory of systems analysis and some applications in civil engineering. There are a fair numbers of errors in the text, however.

Park, Chan S. and Gunter P. Sharp-Bette (1990), *Advanced Engineering Economcs*, John Wiley and Sons, N.Y., 740 pp.
Nicely integrates optimization into engineering economics.

Revelle, Charles S., Earl E. Whitlach, and Jeff R. Wright (1997), *Civil and Environmental Systems Engineering*, Prentice-Hall Upper Saddle R., NJ, 507 pp.
Excellent treatment of linear programming, its applications, and engineering economics.

Stark, Robert M. and Robert L. Nicholls (1972), *Mathematical Foundations for Design: Civil Engineering Systems*, McGraw-Hill, N.Y., 566 pp.
A fine early text with interesting examples from a broad range of civil engineering applications.