Integer Programming

For ARE 521 QUANTITATIVE TECHNIQUES

Offered by Prof. G. D’Souza

Co-Instructors: Anura Amarasinghe and Dr. Tanya Borisova

Linear Programming: *Assumptions*

- Proportionality
  - a change in a variable result in a proportionate change in that variable's contribution to the value of the function
- Additivity
  - the function value is the sum of the contributions of each term
- Divisibility
  - The decision variables can be divided into non-integer values, taking on fractional values

Linear Programming Assumptions

- Proportionality
  - a change in a variable result in a proportionate change in that variable's contribution to the value of the function
- Additivity
  - the function value is the sum of the contributions of each term
- Divisibility
  - The decision variables can be divided into non-integer values, taking on fractional values

Integer programming is used if the divisibility assumption does not hold

Integer Programming

- optimization of a linear objective function
- subject to linear constraints, nonnegativity conditions
- integer value conditions
- *(Can handle non-linear problems)*

IP Problem Classification

- Pure integer IP problem – in which all variables are integer
- Mixed-integer IP problem – involves some integer and some continuous variables
- Zero-one IP problem - in which the integer variables are restricted to either zero or one
  - pure zero-one IP problems – all variables are zero-one
  - mixed zero-one IP problems - containing both zero-one and continuous variables.

IP Problem: *General Formulation*

$$\text{Max } C_1 W + C_2 X + C_3 Y$$

$$\text{s.t. } A_1 W + A_2 X + A_3 Y \leq b$$

$$W \geq 0$$

$$X \geq 0 \text{ and integer}$$

$$Y = 0 \text{ or } 1$$

Why Integer Programming

- Allows to depict discontinuous decision variables
  - acquisition of machines, hired labor or animals
- Allows modeling of fixed costs, logical conditions, and discrete levels of resources

Why Integer Programming

- Allows to depict discontinuous decision variables
  - acquisition of machines, hired labor or animals
- Allows modeling of **fixed costs**, logical conditions, and discrete levels of resources

Fixed Cost

- X - continuous number of units of a good produced;
- Y - zero-one variable indicating whether or not fixed costs are incurred;
- C - per unit revenue from producing X;
- F - fixed cost incurred when producing a nonzero quantity of X;
- M denote a large number.

$$\text{Max} \quad C\,X - F\,Y$$

$$\text{S.t.} \quad X - M\,Y \le 0$$

$$X \ge 0$$

$$Y = 0 \text{ or } 1$$

Fixed Cost (cont.)

$$\text{Max} \quad C\,X - F\,Y$$

$$\text{S.t.} \quad X - M\,Y \le 0$$

$$X \ge 0$$

$$Y = 0 \text{ or } 1$$

What if $X = 0$? the constraint relating X and Y allows Y to be 0 or 1. Given F > 0 then the objective function would cause Y to equal 0

What if $0 < X \le M$? Y must equal to 1

Thus, any non-zero production level for X causes the fixed cost (F) to be incurred. The parameter M is an upper bound on the production of X (a capacity limit).

The fixed cost of equipment investment may be modeled similarly (see McCarl and Spreen 1997).

Why Integer Programming

- Allows to depict discontinuous decision variables
  - acquisition of machines, hired labor or animals
- Allows modeling of fixed costs, **logical conditions**, and discrete levels of resources

Logical Conditions

- Conditional Use
  - A warehouse can be used only if constructed.
- Complementary Products
  - If any of product *A* is produced, then a minimum quantity of product *B* must be produced.
- Complementary Investment
  - If a particular class of equipment is purchased then only complementary equipment can be acquired.
- Sequencing
  - Operation A must be entirely complete before operation B starts.

Logical Conditions (cont.)

- All of these conditions can be imposed using a zero-one indicator variable.
- *Example*: producing mutually exclusive products X and Z

Let Y_1 indicate whether or not X is produced, and Y_2 indicate whether or not Z is produced, M – a large number

$$
\begin{aligned}
X \quad - M Y_1 \quad &\leq 0 \\
Z \quad - M Y_2 \quad &\leq 0 \\
Y_1 + Y_2 \quad &\leq 1 \\
X, \; Z \quad &\geq 0 \\
Y_1, \; Y_2 \quad &= 0 \text{ or } 1
\end{aligned}
$$

Logical Conditions (cont.)

$$
\begin{aligned}
X \quad - M Y_1 \quad &\leq 0 \\
Z \quad - M Y_2 &\leq 0 \\
Y_1 + Y_2 &\leq 1 \\
X, \quad Z \quad &\geq 0 \\
Y_1, \quad Y_2 &= 0 \text{ or } 1
\end{aligned}
$$

where Y_1 indicates whether or not X is produced, while Y_2 indicates whether or not Z is produced, and M – a large number

Can X and Z be produced simultaneously (X>0 and Z>0)?

No. $Y_1 + Y_2 \leq 1$, in conjunction with the zero-one restriction on Y_1 and Y_2, imposes mutual exclusivity.

when $Y_1 = 1$ then X can be produced but Z cannot. Similarly, when $Y_2 = 1$ then X must be zero while $0 \leq Z \leq M$. Consequently, either X or Z can be produced, but not both.

Why Integer Programming

- Allows to depict discontinuous decision variables
  - acquisition of machines, hired labor or animals
- Allows modeling of fixed costs, logical conditions, and **discrete levels of resources**

Discrete Levels of Resources

- Farm has 3 fields.
- Each field is planted to a single crop.
- Let
  - F_1, F_2, and F_3 be field sizes (acres)
  - X_1 and X_2 – be acreage of crop 1 and crop 2
  - The variable Y_i indicates whether field *i* is planted to crop 1 or crop 2.

- Constraints:

$$X_1 - F_1 \quad Y_1 - F_2 \quad Y_2 - F_3 \quad Y_3 = 0$$

$$X_2 - F_1 (1-Y_1) - F_2 (1-Y_2) - F_3 (1-Y_3) = 0$$

$$X_k \geq 0,\ Y_i = 0 \text{ or } 1 \text{ for all } k \text{ and } i$$

Feasible Region Characteristics and Solution Difficulties

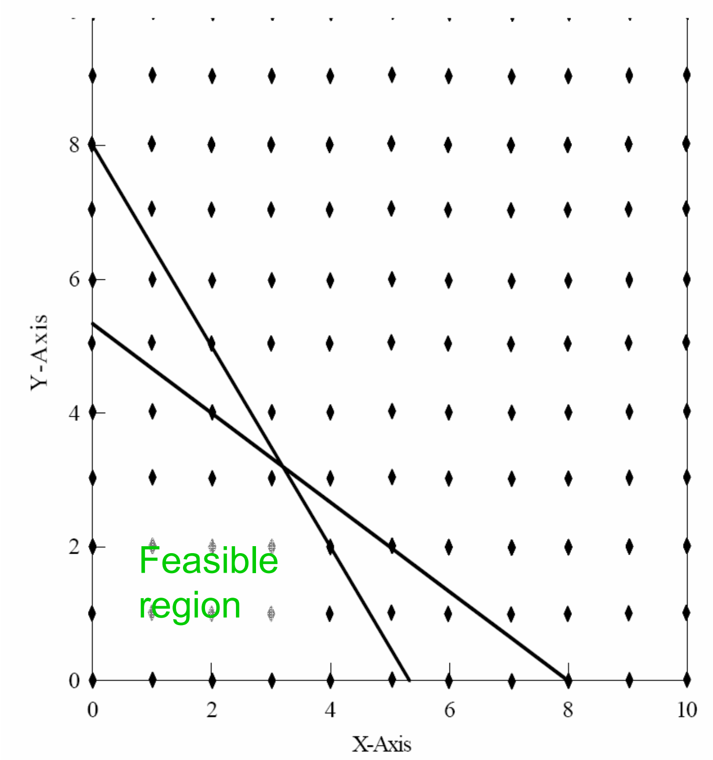
- LP:
  - Solutions at constraint intersections / corner points.
  - Given any two feasible points, all points in between will be feasible
- NLP:
  - Can apply calculus concepts (take derivatives) or Khun-Tucker theory
- IP problems are notoriously difficult to solve.
  - There is no particular location for the potential IP solutions
  - Given any two feasible solutions, all the points in between are not feasible
  - Calculus concepts (such as derivatives or Kuhn-Tucker theory) are not applicable because of discontinuous feasible region

IP Feasibility Region

$$2X + 3Y \leq 16$$

$$3X + 2Y \leq 16$$

X, Y – positive integer variables



Dual Variables and IP

- Dual information (*reduced cost, shadow prices, slack / surplus*) is not well defined in the general IP problem.

- In many cases, IP shadow price information in the output should be ignored.
  - dual information is often influenced by constraints which are added during the solution process.
  - many of the shadow prices reported by IP codes are not relevant to the original problem, but rather are relevant to a transformed problem.
  - the set of transformations is not unique,

- Reliability of dual variables in mixed IP problems – no definite conclusion from literature

- Dual variables in IP may be *nonzero* for *nonbinding* constraints, because optimal solution can be in the interior of the feasible region

Solution Approaches

- None of the available algorithms have been shown to perform satisfactorily for all IP problems.

- Certain types of algorithms are good at solving certain types of problems

Rounding

- involves the solution of the problem as a LP problem followed by an attempt to round the solution to an integer one by:
  - dropping all the fractional parts; or
  - searching out satisfactory solutions wherein the variable values are adjusted to nearby larger or smaller integer values.
- Caution: One should compare the rounded and unrounded solutions to see whether after rounding:
  - the constraints are adequately satisfied; and
  - whether the difference between the optimal LP and the post rounding objective function value is reasonably small

Branch and Bound Algorithm

- Starts with LP solution
- imposes constraints to force the LP solution to become an integer solution
- constraints are upper and lower bounds on variables.

Branch and Bound Algorithm: *Example*

Max $X_1 + X_2$

$2 X_1 + 3 X_2 \leq 16$

$3 X_1 + 2 X_2 \leq 16$

$X_1, X_2 \geq 0$ and integer

LP solution: $X_1 = X_2 = 3.2$

Branch and Bound Algorithm: *Example (cont.)*

Two disjoint problems:

$$\begin{aligned} \text{Max} \quad & X_1 + X_2 \\ & 2X_1 + 3X_2 \le 16 \\ & 3X_1 + 2X_2 \le 16 \\ & X_1 \le 3 \\ & X_1, X_2 \ge 0 \end{aligned}$$

$$\begin{aligned} \text{Max} \quad & X_1 + X_2 \\ & 2X_1 + 3X_2 \le 16 \\ & 3X_1 + 2X_2 \le 16 \\ & X_1 \ge 4 \\ & X_1, X_2 \ge 0 \end{aligned}$$

Branch and Bound Algorithm *(cont.)*

- Branch and bound solution procedure generates two problems (branches) after each LP solution.
- One must decide
  - which variable to branch upon and
  - which problem to solve (branch to follow).

Branch and Bound Algorithm *(cont.)*

- Branch and bound approach is the most commonly used general purpose IP solution algorithm.
- However, its use can be expensive.
  - Often the branch and bound algorithm will come up with near optimal solutions quickly but will then spend a lot of time verifying optimality.
  - Shadow prices from the algorithm can be misleading since they include shadow prices for the bounding constraints.

IP and GAMS

- Introduce a new class of variable declaration statements
  - BINARY (zero one) or INTEGER variables.
- Invoking an IP solver.
  - "USING MIP"
- Integer Programming Examples:
  - McCarl and Spreen, Text examples implemented in GAMS in zipped format at http://agecon2.tamu.edu/people/faculty/mccarl-bruce/books.htm (e.g., WAREHOUS.gms)

Reference

- McCarl B.A. and T. H. Spreen. 1997. *Applied Mathematical Programming Using Algebraic Systems*. Chapters 15 and 16 (http://agecon2.tamu.edu/people/faculty/mccarl-bruce/books.htm)