OPERATIONS RESEARCH LECTURE NOTES

Y. İlker Topcu, Ph.D.

Acknowledgements:
We would like to acknowledge Prof. W.L. Winston's "Operations Research: Applications and Algorithms" and Prof. J.E. Beasley's lecture notes which greatly influence these notes...
We retain responsibility for all errors and would love to hear from visitors of this site!
Istanbul Technical University OR/MS team

www.isl.itu.edu.tr/ya

CONTENTS

1. INTRODUCTION TO OR

1.1 TERMINOLOGY

The British/Europeans refer to "operational research", the Americans to "operations research" - but both are often shortened to just "OR" (which is the term we will use).

Another term which is used for this field is "management science" ("MS"). The Americans sometimes combine the terms OR and MS together and say "OR/MS" or "ORMS".

Yet other terms sometimes used are "industrial engineering" ("IE"), "decision science" ("DS"), and “problem solving”.

In recent years there has been a move towards a standardization upon a single term for the field, namely the term "OR".

1.2 THE METHODOLOGY OF OR

When OR is used to solve a problem of an organization, the following seven step procedure should be followed:

<u>Step 1. Formulate the Problem</u>

OR analyst first defines the organization's problem. Defining the problem includes specifying the organization's objectives and the parts of the organization (or system) that must be studied before the problem can be solved.

<u>Step 2. Observe the System</u>

Next, the analyst collects data to estimate the values of parameters that affect the organization's problem. These estimates are used to develop (in Step 3) and evaluate (in Step 4) a mathematical model of the organization's problem.

<u>Step 3. Formulate a Mathematical Model of the Problem</u>

The analyst, then, develops a mathematical model (in other words an idealized representation) of the problem. In this class, we describe many mathematical techniques that can be used to model systems.

Step 4. Verify the Model and Use the Model for Prediction

The analyst now tries to determine if the mathematical model developed in Step 3 is an accurate representation of reality. To determine how well the model fits reality, one determines how valid the model is for the current situation.

Step 5. Select a Suitable Alternative

Given a model and a set of alternatives, the analyst chooses the alternative (if there is one) that best meets the organization's objectives.

Sometimes the set of alternatives is subject to certain restrictions and constraints. In many situations, the best alternative may be impossible or too costly to determine.

Step 6. Present the Results and Conclusions of the Study

In this step, the analyst presents the model and the recommendations from Step 5 to the decision making individual or group. In some situations, one might present several alternatives and let the organization choose the decision maker(s) choose the one that best meets her/his/their needs.

After presenting the results of the OR study to the decision maker(s), the analyst may find that s/he does not (or they do not) approve of the recommendations. This may result from incorrect definition of the problem on hand or from failure to involve decision maker(s) from the start of the project. In this case, the analyst should return to Step 1, 2, or 3.

Step 7. Implement and Evaluate Recommendation

If the decision maker(s) has accepted the study, the analyst aids in implementing the recommendations. The system must be constantly monitored (and updated dynamically as the environment changes) to ensure that the recommendations are enabling decision maker(s) to meet her/his/their objectives.

1.3 HISTORY OF OR

(Prof. Beasley’s lecture notes)

OR is a relatively new discipline. Whereas 70 years ago it would have been possible to study mathematics, physics or engineering (for example) at university it would not have been possible to study OR, indeed the term OR did not exist then. It was only really in the late 1930's that operational research began in a systematic fashion, and it started in the UK.

Early in 1936 the British Air Ministry established Bawdsey Research Station, on the east coast, near Felixstowe, Suffolk, as the centre where all pre-war radar experiments for both the Air Force and the Army would be carried out. Experimental radar equipment was brought up to a high state of reliability and ranges of over 100 miles on aircraft were obtained.

It was also in 1936 that Royal Air Force (RAF) Fighter Command, charged specifically with the air defense of Britain, was first created. It lacked however any effective fighter aircraft - no Hurricanes or Spitfires had come into service - and no radar data was yet fed into its very elementary warning and control system.

It had become clear that radar would create a whole new series of problems in fighter direction and control so in late 1936 some experiments started at Biggin Hill in Kent into the effective use of such data. This early work, attempting to integrate radar data with ground based observer data for fighter interception, was the start of OR.

The first of three major pre-war air-defense exercises was carried out in the summer of 1937. The experimental radar station at Bawdsey Research Station was brought into operation and the information derived from it was fed into the general air-defense warning and control system. From the early warning point of view this exercise was encouraging, but the tracking information obtained from radar, after filtering and transmission through the control and display network, was not very satisfactory.

In July 1938 a second major air-defense exercise was carried out. Four additional radar stations had been installed along the coast and it was hoped that Britain now had an aircraft location and control system greatly improved both in coverage and effectiveness. Not so! The exercise revealed, rather, that a new and serious problem had arisen. This was the need to coordinate and correlate the additional, and often conflicting, information received from the additional radar stations. With the out-break of war apparently imminent, it was obvious that something new - drastic if necessary - had to be attempted. Some new approach was needed.

Accordingly, on the termination of the exercise, the Superintendent of Bawdsey Research Station, A.P. Rowe, announced that although the exercise had again demonstrated the technical feasibility of the radar system for detecting aircraft, its operational achievements still fell far short of requirements. He therefore proposed that a crash program of research into the operational - as opposed to the technical - aspects of the system should begin immediately. The term "operational research" [RESEARCH into (military) OPERATIONS] was coined as a suitable description of this new branch of applied science. The first team was selected from amongst the scientists of the radar research group the same day.

In the summer of 1939 Britain held what was to be its last pre-war air defense exercise. It involved some 33,000 men, 1,300 aircraft, 110 antiaircraft guns, 700 searchlights, and 100 barrage balloons. This exercise showed a great improvement in the operation of the air defense warning and control system. The contribution made by the OR teams was so apparent that the Air Officer Commander-in-Chief RAF Fighter Command (Air Chief Marshal Sir Hugh Dowding) requested that, on the outbreak of war, they should be attached to his headquarters at Stanmore.

On May 15th 1940, with German forces advancing rapidly in France, Stanmore Research Section was asked to analyze a French request for ten additional fighter squadrons (12 aircraft a squadron) when losses were running at some three squadrons every two days. They prepared graphs for Winston Churchill (the British Prime Minister of the time), based upon a study of current daily losses and replacement rates, indicating how rapidly such a move would deplete fighter strength. No aircraft were sent and most of those currently in France were recalled.

This is held by some to be the most strategic contribution to the course of the war made by OR (as the aircraft and pilots saved were consequently available for the successful air defense of Britain, the Battle of Britain).

In 1941 an Operational Research Section (ORS) was established in Coastal Command which was to carry out some of the most well-known OR work in World War II.

Although scientists had (plainly) been involved in the hardware side of warfare (designing better planes, bombs, tanks, etc) scientific analysis of the operational use of military resources had never taken place in a systematic fashion before the Second World War. Military personnel, often by no means stupid, were simply not trained to undertake such analysis.

These early OR workers came from many different disciplines, one group consisted of a physicist, two physiologists, two mathematical physicists and a surveyor. What such people brought to their work were "scientifically trained" minds, used to querying assumptions, logic, exploring hypotheses, devising experiments, collecting data, analyzing numbers, etc. Many too were of high intellectual caliber (at least four wartime OR personnel were later to win Nobel prizes when they returned to their peacetime disciplines).

By the end of the war OR was well established in the armed services both in the UK and in the USA.

OR started just before World War II in Britain with the establishment of teams of scientists to study the strategic and tactical problems involved in military operations. The objective was to find the most effective utilization of limited military resources by the use of quantitative techniques.

Following the end of the war OR spread, although it spread in different ways in the UK and USA.

You should be clear that the growth of OR since it began (and especially in the last 30 years) is, to a large extent, the result of the increasing power and widespread availability of computers. Most (though not all) OR involves carrying out a large number of numeric calculations. Without computers this would simply not be possible.

2. BASIC OR CONCEPTS

"OR is the representation of real-world systems by mathematical models together with the use of quantitative methods (algorithms) for solving such models, with a view to optimizing."

We can also define a mathematical model as consisting of:

- *Decision variables*, which are the unknowns to be determined by the solution to the model.
- *Constraints* to represent the physical limitations of the system
- An *objective* function
- An *optimal solution* to the model is the identification of a set of variable values which are feasible (satisfy all the constraints) and which lead to the optimal value of the objective function.

In general terms we can regard OR as being the application of scientific methods / thinking to decision making.

Underlying OR is the philosophy that:

- decisions have to be made; and
- using a quantitative (explicit, articulated) approach will lead to better decisions than using non-quantitative (implicit, unarticulated) approaches.

Indeed it can be argued that although OR is imperfect it offers the *best* available approach to making a particular decision in many instances (which is *not* to say that using OR will produce the *right* decision).

Two Mines Example

The Two Mines Company own two different mines that produce an ore which, after being crushed, is graded into three classes: high, medium and low-grade. The company has contracted to provide a smelting plant with 12 tons of high-grade, 8 tons of medium-grade and 24 tons of low-grade ore per week. The two mines have different operating characteristics as detailed below.

Mine	Cost per day (£'000)	Production (tons/day)		
		High	Medium	Low
X	180	6	3	4
Y	160	1	1	6

How many days per week should each mine be operated to fulfill the smelting plant contract?

Guessing

To explore the Two Mines problem further we might simply guess (i.e. use our judgment) how many days per week to work and see how they turn out.

- work one day a week on X, one day a week on Y

This does not seem like a good guess as it results in only 7 tones a day of high-grade, insufficient to meet the contract requirement for 12 tones of high-grade a day. We say that such a solution is *infeasible*.

- work 4 days a week on X, 3 days a week on Y

This seems like a better guess as it results in sufficient ore to meet the contract. We say that such a solution is *feasible*. However it is quite expensive (costly).

We would like a solution which supplies what is necessary under the contract at minimum cost. Logically such a minimum cost solution to this decision problem must exist. However even if we keep guessing we can never be sure whether we have found this minimum cost solution or not. Fortunately our structured approach will enable us to find the minimum cost solution.

Solution

What we have is a verbal description of the Two Mines problem. What we need to do is to translate that verbal description into an *equivalent* mathematical description.

In dealing with problems of this kind we often do best to consider them in the order:

- Variables
- Constraints
- Objective

This process is often called *formulating* the problem (or more strictly formulating a mathematical representation of the problem).

Variables

These represent the "decisions that have to be made" or the "unknowns".

Let

x = number of days per week mine X is operated

y = number of days per week mine Y is operated

Note here that x >= 0 and y >= 0.

Constraints

It is best to first put each constraint into words and then express it in a mathematical form.

ore production constraints - balance the amount produced with the quantity required under the smelting plant contract

Ore

High 6x + 1y >= 12

Medium 3x + 1y >= 8

Low 4x + 6y >= 24

days per week constraint - we cannot work more than a certain maximum number of days a week e.g. for a 5 day week we have

x <= 5

y <= 5

Inequality constraints

Note we have an inequality here rather than an equality. This implies that we may produce more of some grade of ore than we need. In fact we have the general rule: given a choice between an equality and an inequality choose the inequality

For example - if we choose an equality for the ore production constraints we have the three equations 6x+y=12, 3x+y=8 and 4x+6y=24 and there are no values of x and y which satisfy all three equations (the problem is therefore said to be "over-constrained"). For example the values of x and y which satisfy 6x+y=12 and 3x+y=8 are x=4/3 and y=4, but these values do not satisfy 4x+6y=24.

The reason for this general rule is that choosing an inequality rather than an equality gives us more flexibility in optimizing (maximizing or minimizing) the objective (deciding values for the decision variables that optimize the objective).

Implicit constraints

Constraints such as days per week constraint are often called implicit constraints because they are implicit in the definition of the variables.

<u>Objective</u>

Again in words our objective is (presumably) to minimize cost which is given by 180x + 160y

Hence we have the **complete mathematical representation** of the problem:

minimize
    180x + 160y
subject to
    6x + y >= 12
    3x + y >= 8
    4x + 6y >= 24
    x <= 5
    y <= 5
    x,y >= 0

Some notes

The mathematical problem given above has the form

- all variables continuous (i.e. can take fractional values)
- a single objective (maximize or minimize)
- the objective and constraints are linear i.e. any term is either a constant or a constant multiplied by an unknown (e.g. 24, 4x, 6y are linear terms but xy is a non-linear term)

Any formulation which satisfies these three conditions is called a *linear program* (LP).

We have (implicitly) assumed that it is permissible to work in fractions of days - problems where this is not permissible and variables must take integer values will be dealt with under *Integer Programming* (IP).

Discussion

This problem was a *decision problem.*

We have taken a real-world situation and constructed an equivalent mathematical representation - such a representation is often called a mathematical *model* of the real-world situation (and the process by which the model is obtained is called *formulating* the model).

Just to confuse things the mathematical model of the problem is sometimes called the *formulation* of the problem.

Having obtained our mathematical model we (hopefully) have some quantitative method which will enable us to numerically solve the model (i.e. obtain a numerical solution) - such a quantitative method is often called an *algorithm* for solving the model.

Essentially an algorithm (for a particular model) is a set of instructions which, when followed in a step-by-step fashion, will produce a numerical solution to that model.

Our model has an *objective*, that is something which we are trying to *optimize*.

Having obtained the numerical solution of our model we have to translate that solution back into the real-world situation.

"OR is the representation of real-world systems by mathematical models together with the use of quantitative methods (algorithms) for solving such models, with a view to optimizing."

3. LINEAR PROGRAMMING

It can be recalled from the Two Mines example that the conditions for a mathematical model to be a linear program (LP) were:

- all variables continuous (i.e. can take fractional values)
- a single objective (minimize or maximize)
- the objective and constraints are linear i.e. any term is either a constant or a constant multiplied by an unknown.

LP's are important - this is because:

- many practical problems can be formulated as LP's
- there exists an algorithm (called the *simplex* algorithm) which enables us to solve LP's numerically relatively easily

We will return later to the simplex algorithm for solving LP's but for the moment we will concentrate upon formulating LP's.

Some of the major application areas to which LP can be applied are:

- Work scheduling
- Production planning & Production process
- Capital budgeting
- Financial planning
- Blending (e.g. Oil refinery management)
- Farm planning
- Distribution
- Multi-period decision problems
    - Inventory model
    - Financial models
    - Work scheduling

Note that the key to formulating LP's is practice. However a useful hint is that common objectives for LP's are maximize profit/minimize cost.

There are four basic assumptions in LP:

- Proportionality
    - The contribution to the objective function from each decision variable is proportional to the value of the decision variable (The contribution to the objective function from making four soldiers (4×\$3=\$12) is exactly four times the contribution to the objective function from making one soldier (\$3))
    - The contribution of each decision variable to the LHS of each constraint is proportional to the value of the decision variable (It takes exactly three times as many finishing hours (2hrs×3=6hrs) to manufacture three soldiers as it takes to manufacture one soldier (2 hrs))
- Additivity
    - The contribution to the objective function for any decision variable is independent of the values of the other decision variables (No matter what the value of train (x_2), the manufacture of soldier (x_1) will always contribute $3x_1$ dollars to the objective function)
    - The contribution of a decision variable to LHS of each constraint is independent of the values of other decision variables (No matter what the value of x1, the manufacture of x2 uses x2 finishing hours and x2 carpentry hours)

    1st implication: The value of objective function is the sum of the contributions from each decision variables.

    2nd implication: LHS of each constraint is the sum of the contributions from each decision variables.
- Divisibility
    - Each decision variable is allowed to assume fractional values. If we actually can not produce a fractional number of decision variables, we use IP (It is acceptable to produce 1.69 trains)
- Certainty
    - Each parameter is known with certainty

3.1 FORMULATING LP

3.1.1 Giapetto Example

(Winston 3.1, p. 49)

Giapetto's wooden soldiers and trains. Each soldier sells for $27, uses $10 of raw materials and takes $14 of labor & overhead costs. Each train sells for $21, uses $9 of raw materials, and takes $10 of overhead costs. Each soldier needs 2 hours finishing and 1 hour carpentry; each train needs 1 hour finishing and 1 hour carpentry. Raw materials are unlimited, but only 100 hours of finishing and 80 hours of carpentry are available each week. Demand for trains is unlimited; but at most 40 soldiers can be sold each week. How many of each toy should be made each week to maximize profits.

<u>**Answer**</u>

Decision variables completely describe the decisions to be made (in this case, by Giapetto). Giapetto must decide how many soldiers and trains should be manufactured each week. With this in mind, we define:

x_1 = the number of soldiers produced per week,

x_2 = the number of trains produced per week,

Objective function is the function of the decision variables that the decision maker wants to maximize (revenue or profit) or minimize (costs). Giapetto can concentrate on maximizing the total weekly profit (z).

Here profit equals to (weekly revenues) – (raw material purchase cost) – (other variable costs). Hence Giapetto's objective function is:

Maximize $z = 3x_1 + 2x_2$

Constraints show the restrictions on the values of the decision variables. Without constraints Giapetto could make a large profit by choosing decision variables to be very large. Here there are three constraints:

Finishing time per week

Carpentry time per week

Weekly demand for soldiers

Sign restrictions are added if the decision variables can only assume nonnegative values (Giapetto can not manufacture negative number of soldiers or trains!)

All these characteristics explored above give the following **Linear Programming** (LP) problem

$$\max z = 3x_1 + 2x_2 \qquad \text{(The Objective function)}$$

$$\begin{aligned} \text{s.t.} \quad 2x_1 + x_2 &\leq 100 && \text{(Finishing constraint)} \\ x_1 + x_2 &\leq 80 && \text{(Carpentry constraint)} \\ x_1 &\leq 40 && \text{(Constraint on demand for soldiers)} \\ x_1, x_2 &\geq 0 && \text{(Sign restrictions)} \end{aligned}$$

A value of (x_1,x_2) is in the **feasible region** if it satisfies all the constraints and sign restrictions.

Graphically and computationally we see the solution is $(x_1,x_2) = (20,60)$ at which $z = 180$. (**Optimal solution**)

Report

The maximum profit is $180 by making 20 soldiers and 60 trains each week. Profit is limited by the carpentry and finishing labor available. Profit could be increased by buying more labor.

3.1.2 Advertisement Example

(Winston 3.2, p.61)

Dorian makes luxury cars and jeeps for high-income men and women. It wishes to advertise with 1 minute spots in comedy shows and football games. Each comedy spot costs $50K and is seen by 7M high-income women and 2M high-income men. Each football spot costs $100K and is seen by 2M high-income women and 12M high-income men. How can Dorian reach 28M high-income women and 24M high-income men at the least cost.

Answer: The decision variables are

$$x_1 = \text{the number of comedy spots}$$
$$x_2 = \text{the number of football spots.}$$

Giving the problem

$$\begin{array}{lrcrcr}
\min\ z = & 50x_1 & + & 100x_2 & & \\
\text{st} & 7x_1 & + & 2x_2 & \geq & 28 \\
 & 2x_1 & + & 12x_2 & \geq & 24 \\
 & & x_1, & x_2 & \geq & 0
\end{array}$$

The graphical solution is $z = 320$ when $(x_1,x_2) = (3.6,1.4)$. From the graph, in this problem rounding up to $(x_1,x_2) = (4,2)$ gives the best *integer* solution.

Report: The minimum cost of reaching the target audience is $400K, with 4 comedy spots and 2 football slots. The model is dubious as it does not allow for saturation after repeated viewings.

3.1.3 Diet Example

(Winston 3.4., p. 70)

Ms. Fidan's diet requires that all the food she eats come from one of the four "basic food groups". At present, the following four foods are available for consumption: brownies, chocolate ice cream, cola, and pineapple cheesecake. Each brownie costs 0.5$, each scoop of chocolate ice cream costs 0.2$, each bottle of cola costs 0.3$, and each pineapple cheesecake costs 0.8$. Each day, she must ingest at least 500 calories, 6 oz of chocolate, 10 oz of sugar, and 8 oz of fat. The nutritional content per unit of each food is shown in Table. Formulate an LP model that can be used to satisfy her daily nutritional requirements at minimum cost.

	Calories	Chocolate (ounces)	Sugar (ounces)	Fat (ounces)
Brownie	400	3	2	2
Choc. ice cream (1 scoop)	200	2	2	4
Cola (1 bottle)	150	0	4	1
Pineapple cheesecake (1 piece)	500	0	4	5

Answer

The decision variables:

x_1: number of brownies eaten daily

x_2: number of scoops of chocolate ice cream eaten daily

x_3: bottles of cola drunk daily

x_4: pieces of pineapple cheesecake eaten daily

The objective function (the total cost of the diet in cents):

$$\min w = 50 x_1 + 20 x_2 + 30 x_3 + 80 x_4$$

Constraints:

$$400 x_1 + 200 x_2 + 150 x_3 + 500 x_4 \geq 500 \quad \text{(daily calorie intake)}$$

$$3 x_1 + 2 x_2 \geq 6 \quad \text{(daily chocolate intake)}$$

$$2 x_1 + 2 x_2 + 4 x_3 + 4 x_4 \geq 10 \quad \text{(daily sugar intake)}$$

$$2 x_1 + 4 x_2 + x_3 + 5 x_4 \geq 8 \quad \text{(daily fat intake)}$$

$$x_i \geq 0,\ i = 1, 2, 3, 4 \quad \text{(Sign restrictions!)}$$

Report:

The minimum cost diet incurs a daily cost of 90 cents by eating 3 scoops of chocolate and drinking 1 bottle of cola ($w=90$, $x_2=3$, $x_3=1$)

3.1.4 Post Office Example

(Winston 3.5, p.74)

A PO requires different numbers of employees on different days of the week. Union rules state each employee must work 5 consecutive days and then receive two days off. Find the minimum number of employees needed.

	Mon	Tue	Wed	Thur	Fri	Sat	Sun
Staff Needed	17	13	15	19	14	16	11

Answer: The decision variables are x_i (# of employees starting on day i)

Mathematically we must

$$
\begin{array}{rlllllllll}
\min\ z = & x_1 & +x_2 & +x_3 & +x_4 & +x_5 & +x_6 & +x_7 & & \\
 & x_1 & & & +x_4 & +x_5 & +x_6 & +x_7 & \geq & 17 \\
 & x_1 & +x_2 & & & +x_5 & +x_6 & +x_7 & \geq & 13 \\
 & x_1 & +x_2 & +x_3 & & & +x_6 & +x_7 & \geq & 15 \\
 & x_1 & +x_2 & +x_3 & +x_4 & & & +x_7 & \geq & 19 \\
 & x_1 & +x_2 & +x_3 & +x_4 & +x_5 & & & \geq & 14 \\
 & & +x_2 & +x_3 & +x_4 & +x_5 & +x_6 & & \geq & 16 \\
 & & & +x_3 & +x_4 & +x_5 & +x_6 & +x_7 & \geq & 11
\end{array}
$$

The solution is (x_i) = (4/3,10/3,2,22/3,0,10/3,5) giving z = 67/3. We could round this up to (x_i) = (2,4,2,8,0,4,5) giving z = 25. However restricting the decision var.s to be integers and using Lindo again gives (x_i) = (4,4,2,6,0,4,3) giving z = 23

3.1.5 Sailco Example

(Winston 3.10, p. 99)

Sailco must determine how many sailboats to produce in the next 4 quarters The demand is known to be 40, 60, 75, and 25 boats. Sailco must meet its demands. At the beginning of the 1st Sailco starts with 10 boats in inventory. Sailco can produce up to 40 boats with regular time labor at \$400 per boat, or additional boats at \$450 with overtime labor. Boats made in a quarter can be used to meet that quarter's demand or held in inventory for the next quarter at an extra cost of \$20.00 per boat.

Answer:

The decision variables are for t = 1,2,3,4

x_t = # of boats in quarter t built in regular time

y_t = # of boats in quarter t built in overtime

For convenience, introduce

i_t = # of boats in inventory at the end quarter t

d_t = demand in quarter t

We are given that	$x_t \leq 40, \forall t$
By logic	$i_t = i_{t-1} + x_t + y_t - d_t, \forall t.$
Demand is met iff	$i_t \geq 0, \forall t$
(Sign restrictions	x_t and $y_t \geq 0, \forall t$)

We need to minimize total cost z subject to these three sets of conditions where

$$z = 400(x_1+x_2+x_3+x_4) + 450(y_1+y_2+y_3+y_4) + 20(i_1+i_2+i_3+i_4)$$

Report:

Lindo reveals the solution to be $(x_1, x_2, x_3, x_4) = (40, 40, 40, 25)$ and $(y_1, y_2, y_3, y_4) = (0, 10, 35, 0)$ and the minimum cost of \$78450.00 is achieved by the schedule

		Q1	Q2	Q3	Q4
Regular time (x_t)		40	40	40	25
Overtime (y_t)		0	10	35	0
Inventory (i_t)	10	10	0	0	0
Demand (d_t)		40	60	75	25

3.1.6 Customer Service Level Example

(Winston 3.12, p. 108)

CSL services computers. Its demand (hours) for the time of skilled technicians in the next 5 months is

t	Jan	Feb	Mar	Apr	May
d_t	6000	7000	8000	9500	11000

It starts with 50 skilled technicians at the beginning of January. Each technician can work 160 hrs/month. To train a new technician they must be supervised for 50 hrs by an experienced technician. Each experienced technician is paid $2K/mth and a trainee is paid $1K/mth. Each month 5% of the skilled technicians leave. CSL needs to meet demand and minimize costs.

Answer:

The decision variable is

x_t = # to be trained in month t

We must minimize the total cost. For convenience let

y_t = # experienced tech. at start of t^{th} month

d_t = demand during month t

Then we must

min $z = 2000(y_1+...+y_5)+1000(x_1+...+x_5)$

subject to

$$160y_t-50x_t \geq d_t \quad \text{for } t = 1,...5$$

$$y_1 = 50$$

$$y_t = .95y_{t-1}+x_{t-1} \text{ for } t = 2,3,4,5$$

$x_t,\ y_t \geq 0$

3.2 SOLVING LP

3.2.1 LP Solutions: Four Cases

When an LP is solved, one of the following four cases will occur:

1. The LP has a **unique optimal solution**.
2. The LP has **alternative (multiple) optimal solutions**. It has more than one (actually an infinite number of) optimal solutions
3. The LP is **infeasible**. It has no feasible solutions (The feasible region contains no points).
4. The LP is **unbounded**. In the feasible region there are points with arbitrarily large (in a max problem) objective function values.

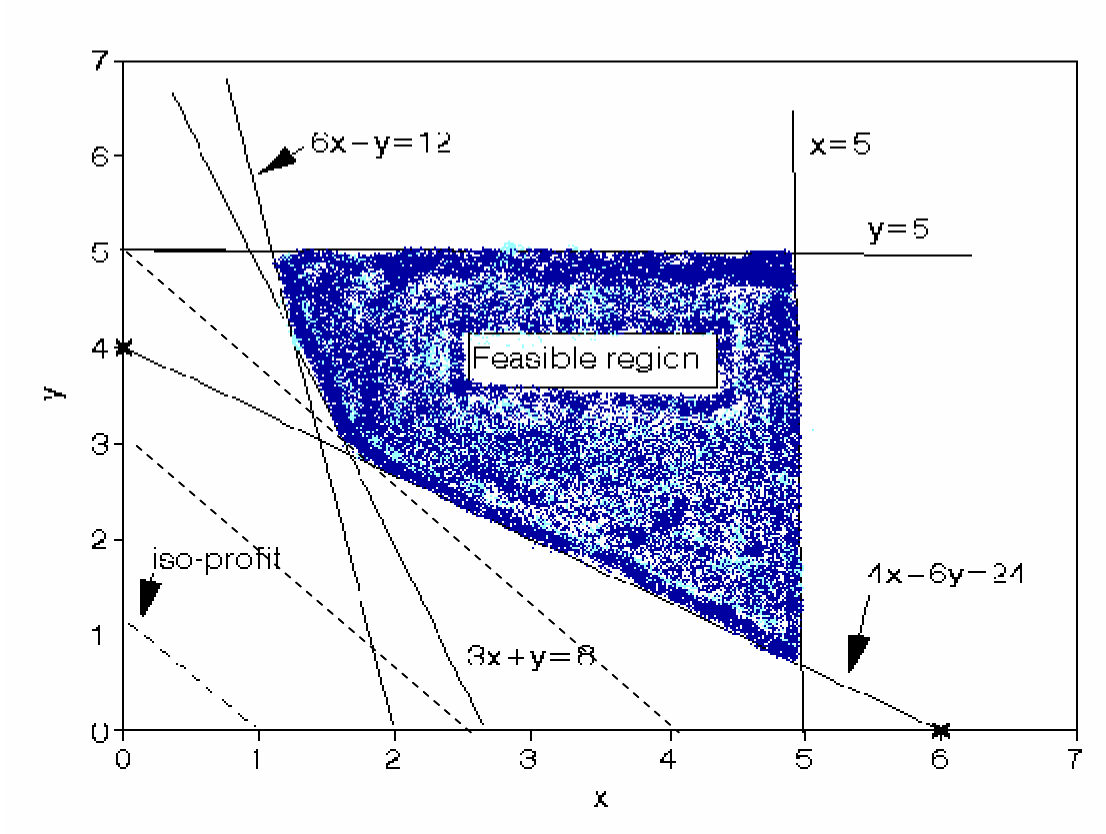
Graphically,

- In the unique optimal solution case, isoprofit line last hits a point (vertex - corner) before leaving the feasible region.
- In the alternative optimal solutions case, isoprofit line last hits an entire line (representing one of the constraints) before leaving the feasible region.
- In the infeasible case, there is no feasible region.
- In the unbounded case, isoprofit line never lose contact with the feasible region

3.2.2 The Graphical Solution (Minimization)

min 180x + 160y
st 6x + y >= 12
3x + y >= 8
4x + 6y >= 24
x <= 5
y <= 5
x,y >= 0

Since there are only two variables in this LP problem we have the graphical representation of the LP given below with the **feasible region** (region of feasible solutions to the constraints associated with the LP) outlined.



We determine the optimal solution to the LP by plotting 180x + 160y = K (K constant) for varying K values (**isoprofit lines**). One such line (180x + 160y = 180) is shown dotted on the diagram. The smallest value of K (remember we are considering a minimization problem) such that 180x + 160y = K goes through a point in the feasible region is the value of the **optimal solution** to the LP (and the corresponding point gives the optimal values of the variables). Hence we can see that the optimal solution to the LP occurs at the vertex of the feasible region formed by the intersection of 3x + y = 8 and 4x + 6y = 24.

Optimal sol'n is 765.71 (1.71 days mine x and 2.86 days mine y are operated)

3.2.3 The Graphical Solution (Maximization)

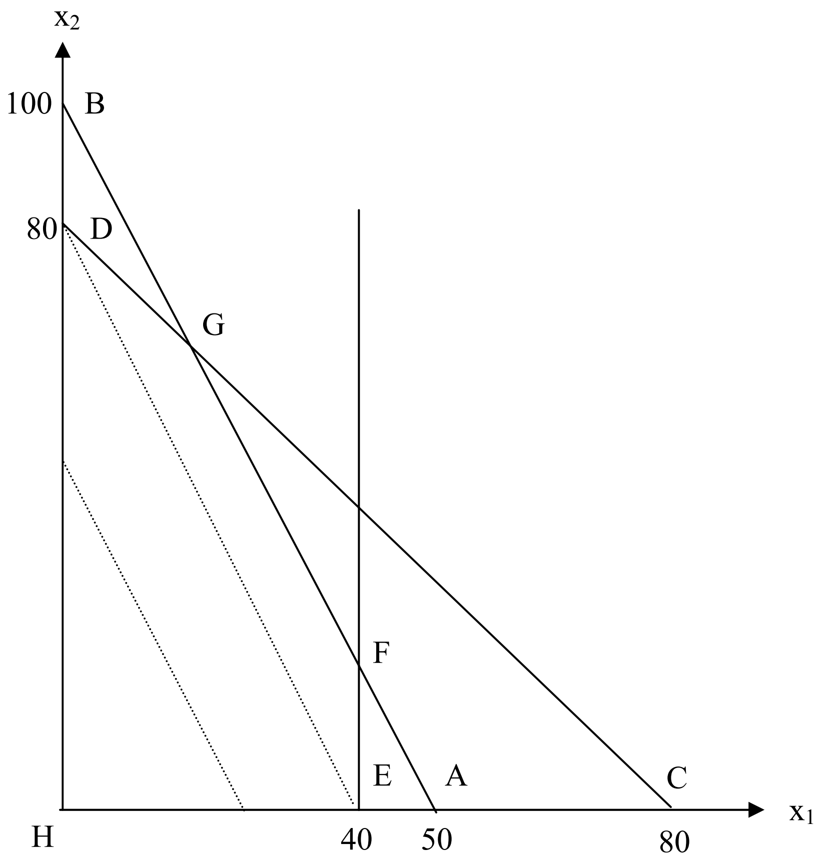
Max $z = 4x_1 + 2x_2$ (The Objective function)

s.t. $2x_1 + x_2 \leq 100$ (Finishing constraint)

$x_1 + x_2 \leq 80$ (Carpentry constraint)

$x_1 \leq 40$ (Constraint on demand for soldiers)

$x_1 + x_2 \geq 0$ (Sign restrictions)



Points on the line between points G (20, 60) and F (40, 20) are the **alternative optimal solutions**. Thus, for $0 \leq c \leq 1$,

$c\ [20\ 60] + (1-c)\ [40\ 20] = [40-20c\ \ 20+40c]$

will be optimal

****

Add constraint $x_2 \geq 90$ (Constraint on demand for trains)

No feasible region: **Infeasible LP**

****

Only use constraint $x_2 \geq 90$

Isoprofit line never lose contact with the feasible region: **Unbounded LP**

3.2.4 The Simplex Algorithm

Note that in the example considered at the graphical solution (min) section, the optimal solution to the LP occurred at a vertex (corner) of the feasible region. In fact it is true that for *any* LP the optimal solution occurs at a vertex of the feasible region. This fact is the key to the simplex algorithm for solving LP's.

Essentially the simplex algorithm starts at one vertex of the feasible region and moves (at each iteration) to another (adjacent) vertex, improving (or leaving unchanged) the objective function as it does so, until it reaches the vertex corresponding to the optimal LP solution.

The simplex algorithm for solving linear programs (LP's) was developed by Dantzig in the late 1940's and since then a number of different versions of the algorithm have been developed. One of these later versions, called the *revised simplex* algorithm (sometimes known as the "product form of the inverse" simplex algorithm) forms the basis of most modern computer packages for solving LP's.

Steps

1. Convert the LP to standard form
2. Obtain a basic feasible solution (bfs) from the standard form
3. Determine whether the current bfs is optimal. If it is optimal, stop.
4. If the current bfs is not optimal, determine which nonbasic variable should become a basic variable and which basic variable should become a nonbasic variable to find a new bfs with a better objective function value
5. Go back to Step 3.

Related concepts:

- Standard form: all constraints are equations and all variables are nonnegative
- bfs: any basic solution where all variables are nonnegative
- Nonbasic variable: a chosen set of variables where variables equal to 0
- Basic variable: the remaining variables that satisfy the system of equations at the standard form

3.2.5 Example illustrating Simplex (Dakota Furniture)

(Winston 4.3, p. 134)

Dakota Furniture makes desks, tables, and chairs. Each product needs the limited resources of lumber, carpentry and finishing; as described in the table. At most 5 tables can be sold per week. Maximize weekly revenue.

Resource	Desk	Table	Chair	Max Avail.
Lumber (board ft.)	8	6	1	48
Finishing hours	4	2	1.5	20
Carpentry hours	2	1.5	.5	8
Max Demand	unlimited	5	unlimited	
Price ($)	60	30	20	

LP Model:

Let x_1, x_2, x_3 be the number of desks, tables and chairs produced. Let the weekly profit be $z. Then, we must

$$
\begin{aligned}
\max z &= 60x_1+30x_2+20x_3 \\
\text{s.t.} \quad & 8x_1+6x_2+x_3 \le 48 \\
& 4x_1+2x_2+1.5x_3 \le 20 \\
& 2x_1+1.5x_2+.5x_3 \le 8 \\
& x_2 \le 5 \\
& x_1,x_2,x_3 \ge 0
\end{aligned}
$$

Solution with Simplex Algorithm

1. First introduce slack variables and convert the LP to the standard form and write a canonical form

$$
\begin{array}{lrcrcrcrrrrl}
\max\ z = & 60x_1 & + & 30x_2 & + & 20x_3 & & & & & \\
\text{s.t.} & 8x_1 & + & 6x_2 & + & x_3 & +s_1 & & & & = 48 \\
 & 4x_1 & + & 2x_2 & + & 1.5x_3 & & +s_2 & & & = 20 \\
 & 2x_1 & + & 1.5x_2 & + & .5x_3 & & & +s_3 & & = 8 \\
 & & & x_2 & & & & & & +s_4 & = 5
\end{array}
$$

$$x_1,x_2,x_3,s_1,s_2,s_3,s_4 \geq 0.$$

$$
\begin{array}{llllllllll}
R_0 & z & -60x_1 & -30x_2 & -20x_3 & & & & & = 0 \\
R_1 & & 8x_1 & +\ 6x_2 & +\ x_3 & +s_1 & & & & = 48 \\
R_2 & & 4x_1 & +\ 2x_2 & +1.5x_3 & & +s_2 & & & = 20 \\
R_3 & & 2x_1 & +1.5x_2 & +\ .5x_3 & & & +s_3 & & = 8 \\
R_4 & & & x_2 & & & & & +s_4 & = 5
\end{array}
$$

2. Obtain a starting bfs.

As $(x_1, x_2, x_3) = 0$ is feasible for the original problem, the below given point where three of the variables equal 0 (the ***non-basic variables***) and the four other variables (the ***basic variables***) are determined by the four equalities is an obvious bfs:

$x_1 = x_2 = x_3 = 0,\ s_1 = 48,\ s_2 = 20,\ s_3 = 8,\ s_4 = 5.$

.

3. Determine whether the current bfs is optimal.

Determine whether there is any way that z can be increased by increasing some nonbasic variable.

If each nonbasic variable has a nonnegative coefficient in the objective function row (***row 0***), current bfs is optimal.

However, here all nonbasic variables have negative coefficients: It is not optimal.

4. Find a new bfs

- z increases most rapidly when x_1 is made non-zero; i.e. x_1 is the ***entering variable***.

- Examining R_1, x_1 can be increased only to 6. More than 6 makes $s_1 < 0$. Similarly R_2, R_3, and R_4, give limits of 5, 4, and no limit for x_1 (***ratio test***). The smallest ratio is the largest value of the entering variable that will keep all the current basic variables nonnegative. Thus by R_3, x_1 can only increase to $x_1 = 4$ when s_3 becomes 0. (I.e. look for the least positive of the ratios 48/8 = 6, 20/4 = 5, 8/2 = 4, 5/0 = ∞). We say s_3 is the ***leaving variable*** and R_3 is the ***pivot equation***.
- Now we must rewrite the system so the values of the basic variables can be read off.

The new *pivot equation* ($R_3/2$) is

$$R_3^{'}: \qquad x_1+.75x_2+.25x_3+ \qquad .5s_3 \quad = 4$$

Then use $R_3^{'}$ to eliminate x_1 in all the other rows.

$R_0^{'}$	$z \quad +15x_2 \quad -5x_3 \quad +30s_3 = 240$	$z = 240$	$(R_0+60R_3^{'})$
$R_1^{'}$	$-x_3 \quad +s_1 \quad -4s_3 \quad = 16$	$s_1 = 16$	$(R_1-8R_3^{'})$
$R_2^{'}$	$-x_2 \quad +.5x_3 \quad + \quad s_2-2s_3 \quad = 4$	$s_2 = 4$	$(R_2-4R_3^{'})$
$R_3^{'}$	$x_1+.75x_2 \quad +.25x_3 \quad +.5s_3 \quad = 4$	$x_1 = 4$	$(R_3^{'})$
$R_4^{'}$	$x_2 \quad + \quad s_4 = 5$	$s_4 = 5$	(R_4)

The new bfs is $x_2 = x_3 = s_3 = 0$, $x_1 = 4$, $s_1 = 16$, $s_2 = 4$, $s_4 = 5$ making $z = 240$.

5. Go back to Step 3. Repeat steps until an optimal solution is reached

- We increase z fastest by making x_3 non-zero (i.e. x_3 enters).
- x_3 can be increased to at most $x_3 = 8$, when $s_2 = 0$ (i.e. s_2 leaves.)

Rearranging the pivot equation gives

$$R_2^{''} \qquad -2x_2+x_3+2s_2-4s_3 = 8 \quad (R_2^{'}\times 2).$$

Row operations with $R_2^{''}$ eliminate x_3 to give the new system

R_0''	$z + 5x_2 + 10s_2 + 10s_3 = 280$	$z = 280$	$(R_0' + 5R_2'')$
R_1''	$-2x_2 + s_1 + 2s_2 - 8s_3 = 24$	$s_1 = 24$	$(R_1' + R_2'')$
R_2''	$-2x_2 + x_3 + 2s_2 - 4s_3 = 8$	$x_3 = 8$	(R_2'')
R_3''	$x_1 + 1.25x_2 - .5s_2 + 1.5s_3 = 2$	$x_1 = 2$	$(R_3' - .5R_2'')$
R_4''	$x_2 + s_4 = 5$	$s_4 = 5$	(R_4')

The bfs is now $x_2 = s_2 = s_3 = 0$, $x_1 = 2$, $x_3 = 8$, $s_1 = 24$, $s_4 = 5$ making $z = 280$.

Each nonbasic variable has a nonnegative coefficient in row 0

THE CURRENT SOLUTION IS OPTIMAL

Report: Dakota furniture's optimum weekly profit would be 280$ if they produce 2 desks and 8 chairs.

This was once written as a ***tableau***.

(***Use tableau format for each operation in all HW and exams!!!***)

Initial tableau:

z	x_1	x_2	x_3	s_1	s_2	s_3	s_4	RHS	BV
1	-60	-30	-20	0	0	0	0	0	z = 0
	8	6	1	1	0	0	0	48	s_1 = 48
	4	2	1.5	0	1	0	0	20	s_2 = 20
	2	1.5	.5	0	0	1	0	8	s_3 = 8
	0	1	0	0	0	0	1	5	s_4 = 5

First tableau:

z	x_1	x_2	x_3	s_1	s_2	s_3	s_4	RHS	BV
1		15	-5			30		240	z = 240
			-1	1		-4		16	s_1 = 16
		-1	.5		1	-2		4	s_2 = 4
	1	.75	.25			.5		4	x_1 = 4
		1					1	5	s_4 = 5

Second and optimal tableau:

z	x_1	x_2	x_3	s_1	s_2	s_3	s_4	RHS	BV
1	0	5	0	0	10	10	0	280	z = 280
	0	-2	0	1	2	-8	0	24	s_1 = 24
	0	-2	1	0	2	-4	0	8	x_3 = 8
	1	1.25	0	0	-.5	1.5	0	2	x_1 = 2
	0	1	0	0	0	0	1	5	s_4 = 5

3.2.6 Other Solution Cases by Simplex

Alternative Optimal Solutions

Dakota example is modified: $35/table

new $z = 60\ x_1 + 35\ x_2 + 20\ x_3$

Second and optimal tableau for the modified problem:

z	x_1	x_2 ⇓	x_3	s_1	s_2	s_3	s_4	RHS	BV	Ratio	
1	0	**0**	0	0	10	10	0	280	z=280		
0	0	-2	0	1	2	-8	0	24	s_1=24	-	
0	0	-2	1	0	2	-4	0	8	x_3=8	-	
0	1	1.25	0	0	-.5	1.5	0	2	x_1=2	2/1.25	⇒
0	0	1	0	0	0	0	1	5	s_4=5	5/1	

Another optimal tableau for the modified problem:

z	x_1	x_2	x_3	s_1	s_2	s_3	s_4	RHS	BV
1	0	0	0	0	10	10	0	280	z=280
0	1.6	0	0	1	1.2	-5.6	0	27.2	s_1=27.2
0	1.6	0	1	0	1.2	-1.6	0	11.2	x_3=11.2
0	0.8	1	0	0	-0.4	1.2	0	1.6	x_2=1.6
0	-0.8	0	0	0	0.4	-1.2	1	3.4	s_4=3.4

Therefore the optimal solution is as follows:

z = 280 and for $0 \leq c \leq 1$

$$\begin{vmatrix} x_1 \\ x_2 \\ x_3 \end{vmatrix} = c \begin{vmatrix} 2 \\ 0 \\ 8 \end{vmatrix} + (1 - c) \begin{vmatrix} 0 \\ 1.6 \\ 11.2 \end{vmatrix} = \begin{vmatrix} 2c \\ 1.6 - 1.6c \\ 11.2 - 3.2c \end{vmatrix}$$

Unbounded LPs

z	x_1	x_2	x_3 ⇓	x_4	s_1	s_2	RHS	BV	Ratio
1	0	2	**-9**	0	12	4	100	z=100	
0	0	1	-6	1	6	-1	20	x_4=20	None
0	1	1	-1	0	1	0	5	x_1=5	None

Since ratio test fails, the LP under consideration is an unbounded LP.

3.2.7 The Big M Method

If an LP has any $\geq$ or = constraints, a starting bfs may not be readily apparent.

When a bfs is not readily apparent, the Big M method or the two-phase simplex method may be used to solve the problem.

The Big M method is a version of the Simplex Algorithm that first finds a bfs by adding "artificial" variables to the problem. The objective function of the original LP must, of course, be modified to ensure that the artificial variables are all equal to 0 at the conclusion of the simplex algorithm.

Steps

1. Modify the constraints so that the RHS of each constraint is nonnegative (This requires that each constraint with a negative RHS be multiplied by -1. Remember that if you multiply an inequality by any negative number, the direction of the inequality is reversed!). After modification, identify each constraint as a $\leq$, $\geq$, or = constraint.
2. Convert each inequality constraint to standard form (If constraint i is a $\leq$ constraint, we add a slack variable s_i; and if constraint i is a $\geq$ constraint, we subtract an excess variable e_i).
3. Add an artificial variable a_i to the constraints identified as $\geq$ or = constraints at the end of Step 1. Also add the sign restriction $a_i \geq 0$.
4. Let M denote a very large positive number. If the LP is a min problem, add (for each artificial variable) Ma_i to the objective function. If the LP is a max problem, add (for each artificial variable) $-Ma_i$ to the objective function.
5. Since each artificial variable will be in the starting basis, all artificial variables must be eliminated from row 0 before beginning the simplex. Now solve the transformed problem by the simplex (In choosing the entering variable, remember that M is a very large positive number!).

If all artificial variables are equal to zero in the optimal solution, we have found the optimal solution to the original problem.

If any artificial variables are positive in the optimal solution, the original problem is **infeasible**!!!

3.2.8 Example illustrating Big M (Oranj Juice)

(Winston 4.10, p. 164)

Bevco manufactures an orange flavored soft drink called Oranj by combining orange soda and orange juice. Each ounce of orange soda contains 0.5 oz of sugar and 1 mg of vitamin C. Each ounce of orange juice contains 0.25 oz of sugar and 3 mg of vitamin C. It costs Bevco 2¢ to produce an ounce of orange soda and 3¢ to produce an ounce of orange juice. Marketing department has decided that each 10 oz bottle of Oranj must contain at least 20 mg of vitamin C and at most 4 oz of sugar. Use LP to determine how Bevco can meet marketing dept.'s requirements at minimum cost.

LP Model:

Let x_1 and x_2 be the quantity of ounces of orange soda and orange juice (respectively) in a bottle of Oranj.

$$
\begin{array}{lll}
\min z = 2x_1+3x_2 & & \\
\text{s.t.} \quad 0.5\,x_1 + 0.25\,x_2 \le 4 & & \text{(sugar const.)} \\
\quad x_1 + 3\,x_2 \ge 20 & & \text{(vit. C const.)} \\
\quad x_1 + x_2 = 10 & & \text{(10 oz in bottle)} \\
x_1, x_2 \ge 0 & &
\end{array}
$$

Solving Oranj Example with Big M Method

<u>1. Modify the constraints so that the RHS of each constraint is nonnegative</u>

The RHS of each constraint is nonnegative

<u>2. Convert each inequality constraint to standard form</u>

$$
\begin{array}{rcl}
z - 2\,x_1 - 3\,x_2 & = & 0 \\
0.5\,x_1 + 0.25\,x_2 + s_1 & = & 4 \\
x_1 + 3\,x_2 - e_2 & = & 20 \\
x_1 + x_2 & = & 10
\end{array}
$$

all variables nonnegative

3. Add a_i to the constraints identified as > or = const.s

$z - 2x_1 - 3x_2$		$= 0$	Row 0
$0.5x_1 + 0.25x_2 + s_1$		$= 4$	Row 1
$x_1 + 3x_2$	$- e_2 + a_2$	$= 20$	Row 2
$x_1 + x_2$	$+ a_3$	$= 10$	Row 3

all variables nonnegative

4. Add Ma_i to the objective function (min problem)

$$\min z = 2x_1 + 3x_2 + M a_2 + M a_3$$

Row 0 will change to

$$z - 2x_1 - 3x_2 - M a_2 - M a_3 = 0$$

5. Since each artificial variable are in our starting bfs, they must be eliminated from row 0

New Row 0 = Row 0 + M * Row 2 + M * Row 3 $\Rightarrow$

$z + (2M-2)x_1 + (4M-3)x_2 - M e_2 = 30M$ New Row 0

Initial tableau:

z	x_1	x_2 ⇓	s_1	e_2	a_2	a_3	RHS	BV	Ratio
1	2M-2	**4M-3**	0	-M	0	0	30M	z=30M	
0	0.5	0.25	1	0	0	0	4	s_1=4	16
0	1	$\boxed{3}$	0	-1	1	0	20	a_2=20	**20/3***
0	1	1	0	0	0	1	10	a_3=10	10

In a min problem, entering variable is the variable that has the "most positive" coefficient in row 0!

First tableau:

z	x_1 ⇓	x_2	s_1	e_2	a_2	a_3	RHS	BV	Ratio
1	**(2M-3)/3**	0	0	(M-3)/3	(3-4M)/3	0	20+3.3M	z	
0	5/12	0	1	1/12	-1/12	0	7/3	s_1	28/5
0	1/3	1	0	-1/3	1/3	0	20/3	x_2	20
0	$\boxed{2/3}$	0	0	1/3	-1/3	1	10/3	a_3	**5***

Optimal tableau:

z	x_1	x_2	s_1	e_2	a_2	a_3	RHS	BV
1	0	0	0	-1/2	(1-2M)/2	(3-2M)/2	25	$z=25$
0	0	0	1	-1/8	1/8	-5/8	1/4	$s_1=1/4$
0	0	1	0	-1/2	1/2	-1/2	5	$x_2=5$
0	1	0	0	1/2	-1/2	3/2	5	$x_1=5$

Report:

In a bottle of Oranj, there should be 5 oz orange soda and 5 oz orange juice. In this case the cost would be 25¢.

3.3 SENSITIVITY ANALYSIS

3.3.1 Reduced Cost

For any nonbasic variable, the reduced cost for the variable is the amount by which the nonbasic variable's objective function coefficient must be improved before that variable will become a basic variable in some optimal solution to the LP.

If the objective function coefficient of a nonbasic variable x_k is improved by its reduced cost, then the LP will have alternative optimal solutions -at least one in which x_k is a basic variable, and at least one in which x_k is not a basic variable.

If the objective function coefficient of a nonbasic variable x_k is improved by more than its reduced cost, then any optimal solution to the LP will have x_k as a basic variable and $x_k>0$.

Reduced cost of a basic variable is zero (see definition)!

3.3.2 Shadow Price

We define the shadow price for the *i*th constraint of an LP to be the amount by which the optimal z value is "improved" (increased in a max problem and decreased in a min problem) if the RHS of the *i*th constraint is increased by 1.

This definition applies only if the change in the RHS of the constraint leaves the current basis optimal!

A $\geq$ constraint will always have a nonpositive shadow price; a $\leq$ constraint will always have a nonnegative shadow price.

3.3.3 Simple Example

$$\max z = 5x1 + x2 + 10x3$$
$$x1 + x3 \leq 100$$
$$x2 \leq 1$$
All variables ≥ 0

This is a very easy LP model and can be solved manually without utilizing Simplex.
x2 = 1 (This variable does not exist in the first constraint. In this case, as the problem is a maximization problem, the optimum value of the variable equals the RHS value of the second constraint).
x1 = 0, x3 = 100 (These two variables do exist only in the first constraint and as the objective function coefficient of x3 is greater than that of x1, the optimum value of x3 equals the RHS value of the first constraint).
Hence, the optimal solution is as follows:
z = 1001, [x1, x2, x3] = [0, 1, 100]

Similarly, sensitivity analysis can be executed manually.
Reduced Cost
As x2 and x3 are in the basis, their reduced costs are 0.
In order to have x1 enter in the basis, we should make its objective function coefficient as great as that of x3. In other words, improve the coefficient as 5 (10-5). New objective function would be (max z = 10 x1 + x2 + 10 x3) and there would be at least two optimal solutions for [x1, x2, x3]: [0, 1, 100] and [100, 1, 0].
Therefore reduced cost of x1 equals 5.
If we improve the objective function coefficient of x1 more than its reduced cost, there would be a unique optimal solution: [100, 1, 0].
Shadow Price
If the RHS of the first constraint is increased by 1, new optimal solution of x3 would be 101 instead of 100. In this case, new z value would be 1011.
If we use the definition: 1011 - 1001 = 10 is the shadow price of the fist constraint.
Similarly the shadow price of the second constraint can be calculated as 1 (please find it).

3.3.4 Utilizing Lindo Output for Sensitivity

NOTICE: The objective function which is regarded as Row 0 in Simplex is accepted as Row 1 in Lindo.

Therefore the first constraint of the model is always second row in Lindo!!!

```
MAX     5 X1 + X2 + 10 X3
SUBJECT TO
       2)  X1 + X3 <=   100
       3)  X2 <=   1
END


LP OPTIMUM FOUND AT STEP      1

      OBJECTIVE FUNCTION VALUE
      1)      1001.000

VARIABLE        VALUE          REDUCED COST
      X1          0.000000          5.000000
      X2          1.000000          0.000000
      X3        100.000000          0.000000

     ROW   SLACK OR SURPLUS     DUAL PRICES
      2)          0.000000           10.000000
      3)          0.000000            1.000000

RANGES IN WHICH THE BASIS IS UNCHANGED:

                      OBJ COEFFICIENT RANGES
VARIABLE         CURRENT        ALLOWABLE       ALLOWABLE
                  COEF           INCREASE        DECREASE
     X1          5.000000        5.000000        INFINITY
     X2          1.000000        INFINITY        1.000000
     X3         10.000000        INFINITY        5.000000

                      RIGHTHAND SIDE RANGES
     ROW        CURRENT        ALLOWABLE       ALLOWABLE
                  RHS           INCREASE        DECREASE
      2        100.000000       INFINITY        100.000000
      3          1.000000       INFINITY          1.000000
```

Lindo output reveals the reduced costs of x1, x2, and x3 as 5, 0, and 0 respectively. In the maximization problems, the reduced cost of a non-basic variable can also be read from the allowable increase value of that variable at obj. coefficient ranges. Here, the corresponding value of x1 is 5.

In the minimization problems, the reduced cost of a non-basic variable can also be read from the allowable decrease value of that variable at obj. coefficient ranges.

The same Lindo output reveals the shadow prices of the constraints in the "dual price" section:
Here, the shadow price of the first constraint (Row 2) equals 10.
The shadow price of the second constraint (Row 3) equals 1.

Some important equations

If the change in the RHS of the constraint leaves the current basis optimal (within the allowable RHS range), the following equations can be used to calculate new objective function value:
for maximization problems

- new obj. fn. value = old obj. fn. value + (new RHS – old RHS) × shadow price

for minimization problems

- new obj. fn. value = old obj. fn. value – (new RHS – old RHS) × shadow price

For the above example, as the allowable increases in RHS ranges are infinity for each constraint, we can increase RHS of them as much as we want. But according to allowable decreases, RHS of the first constraint can be decreased by 100 and that of second constraint by 1.

Lets assume that new RHS value of the first constraint is 60.
As the change is within allowable range, we can use the first equation (max. problem):
$z_{new} = 1001 + (60 - 100) 10 = 601.$

3.3.5 Additional Example (Utilizing Simplex for Sensitivity)

In Dakota furniture example; x_1, x_2, and x_3 were representing the number of desks, tables, and chairs produced.

The LP formulated for profit maximization:

$$\max z = 60x_1 + 30x_2 + 20x_3$$

$$\begin{array}{lllllllll}
8x_1 & + & 6x_2 & + & x_3 & +s_1 & & & = 48 \quad \text{Lumber} \\
4x_1 & + & 2x_2 & + & 1.5x_3 & & +s_2 & & = 20 \quad \text{Finishing} \\
2x_1 & + & 1.5x_2 & + & .5x_3 & & & +s_3 & = 8 \quad \text{Carpentry} \\
 & & x_2 & & & & & +s_4 & = 5 \quad \text{Demand}
\end{array}$$

The optimal solution was:

$$\begin{array}{llllllll}
z & +5x_2 & & & + 10s_2 & + 10s_3 & & = 280 \\
 & -2x_2 & & +s_1 & + 2s_2 & - 8s_3 & & = 24 \\
 & -2x_2 & +x_3 & & + 2s_2 & - 4s_3 & & = 8 \quad (1) \\
 & x_1+1.25x_2 & & & - .5s_2 & + 1.5s_3 & & = 2 \\
 & x_2 & & & & & +s_4 & = 5
\end{array}$$

Analysis 1

Suppose available finishing time changes from $20 \rightarrow 20+\delta$, then we have the system:

$$z' = 60x_1' + 30x_2' + 20x_3'$$

$$\begin{array}{lllllllll}
8x_1' & + & 6x_2' & + & x_3' & +s_1' & & & = 48 \\
4x_1' & + & 2x_2' & + & 1.5x_3' & & +s_2' & & = 20+\delta \\
2x_1' & + & 1.5x_2' & + & .5x_3' & & & +s_3' & = 8 \\
 & & x_2' & & & & & +s_4' & = 5
\end{array}$$

or equivalently:

$$\begin{array}{lllllllll}
z' = 60x_1' & + & 30x_2' & + & 20x_3' & & & & \\
8x_1' & + & 6x_2' & + & x_3' & +s_1' & & & = 48 \\
4x_1' & + & 2x_2' & + & 1.5x_3' & +(s_2'-\delta) & & & = 20 \\
2x_1' & + & 1.5x_2' & + & .5x_3' & & +s_3' & & = 8 \\
 & & x_2' & & & & & +s_4' & = 5
\end{array}$$

That is $z', x_1', x_2', x_3', x_4', s_1', s_2'-\delta, s_3', s_4'$ satisfy the original problem, and hence (1)

Substituting in:

$$\begin{array}{llllllll}
z' & +5x_2' & & & + & 10(s_2'-\delta) & + & 10s_3' & & = 280 \\
 & -2x_2' & & +s_1' & + & 2(s_2'-\delta) & - & 8s_3' & & = 24 \\
 & -2x_2' & +x_3' & & - & 2(s_2'-\delta) & - & 4s_3' & & = 8 \\
 & x_1'+1.25x_2' & & & - & .5(s_2'-\delta) & + & 1.5s_3' & & = 2 \\
 & x_2' & & & & & & & +s_4' & = 5
\end{array}$$

and thus

$$\begin{array}{llllllll}
z' & +5x_2' & & & + & 10s_2' & + & 10s_3' & & = 280+10\delta \\
 & -2x_2' & & +s_1' & + & 2s_2' & - & 8s_3' & & = 24+2\delta \\
 & -2x_2' & +x_3' & & + & 2s_2' & - & 4s_3' & & = 8+2\delta \\
 & x_1'+1.25x_2' & & & - & .5s_2' & + & 1.5s_3' & & = 2-.5\delta \\
 & x_2' & & & & & & & +s_4' & = 5
\end{array}$$

For $-4 \le \delta \le 4$, the new system maximizes z'. In this range RHS are non-negative.

As δ increases, revenue increases by 10δ and. we could pay up to \$10 per hr. for extra finishing labor and still increase revenue. Therefore, the ***shadow price*** of finishing labor is \$10 per hr. (This is valid for up to 4 extra hours or 4 fewer hours).

Analysis 2

What happens if revenue from desks changes to \$60+γ? For small γ, revenue increases by 2γ (as we are making 2 desks currently). But how large an increase is possible?

The new revenue is:

$$
\begin{aligned}
z' &= (60+\gamma)x_1+30x_2+20x_3 = z+\gamma x_1 \\
&= (280-5x_2-10s_2-10s_3)+\gamma(2-1.25x_2+.5s_2-1.5s_3) \\
&= 280+2\gamma-(5+1.25\gamma)x_2-(10-.5\gamma)s_2-(10+1.5\gamma)s_3
\end{aligned}
$$

So the top line in the final system would be:

$$z'+(5+1.25\gamma)x_2+(10-.5\gamma)s_2+(10+1.5\gamma)s_3 = 280+2\gamma$$

Provided all terms in this row are $\geq$, we are still optimal. i.e. $-4 \leq \gamma \leq 20$ the current production schedule is still optimal.

Analysis 3

If revenue from a non-basic variable changes, the revenue is

$$
\begin{aligned}
z' &= 60x_1+(30+\gamma)x_2+20x_3 = z+\gamma x_2 \\
&= 280-5x_2-10s_2-10s_3+\gamma x_2 \\
&= 280-(5-\gamma)x_2-10s_2-10s_3
\end{aligned}
$$

Thus the current solution is optimal for $\gamma \leq 5$. But when $\gamma > 5$ or the revenue per table is increased past \$35, it becomes better to produce tables. We say the **reduced cost** of tables is \$5.00.

3.4 DUALITY

3.4.1 The Dual Theorem

Linear programming theory tells us that (provided the primal LP is feasible) the optimal value of the primal LP is *equal* to the optimal value of the dual LP

3.4.2 Finding the Dual of a Normal Max Problem

For an LP with m constraints and n variables if b is a one-dimensional vector of length m, c and x are one-dimensional vectors of length n and A is a two-dimensional matrix with m rows and n columns the primal linear program (in matrix notation) is:

maximize cx

subject to Ax <= b

x >= 0

Associated with this primal linear program we have the *dual* linear program (involving a one-dimensional vector y of length m) given (in matrix notation) by:

minimize by

subject to Ay >= c

y >= 0

Table 1. Finding the dual of a problem

		max z				
min w		$x_1 \geq 0$	$x_2 \geq 0$	....	$x_n \geq 0$	
.		x_1	x_2		x_n	
$y_1 \geq 0$	y_1	a_{11}	a_{12}	....	a_{1n}	$\leq b_1$
$y_2 \geq 0$	y_2	a_{21}	a_{22}	....	a_{2n}	$\leq b_2$
....	....	....	....		....	....
$y_m \geq 0$	y_m	a_{m1}	a_{m2}	....	a_{mn}	$\leq b_m$
		$\geq c_1$	$\geq c_2$		$\geq c_n$	

A tabular approach makes it easy to find the dual of an LP. If the primal is a normal max problem, it can be *read across* (Table 1); the dual is found by *reading down* the table.

3.4.3 Finding the Dual of a Normal Min Problem

For an LP with m constraints and n variables if b is a one-dimensional vector of length m, c and x are one-dimensional vectors of length n and A is a two-dimensional matrix with m rows and n columns the primal linear program (in matrix notation) is:

minimize cx

subject to Ax >= b

x >= 0

Associated with this primal linear program we have the *dual* linear program (involving a one-dimensional vector y of length m) given (in matrix notation) by:

maximize by

subject to yA <= c

y >= 0

The tabular approach, mentioned above, can be used here similarly. If the primal is a normal min problem, we find it by *reading down* (Table 1); the dual is found by *reading across* in the table.

3.4.4 Finding the Dual of a Nonnormal Max Problem

1. Fill in Table 1 so that the primal can be read across.
2. After making the following changes, the dual can be read down in the usual fashion:

   a) If the *i*th primal constraint is a $\geq$ constraint, the corresponding dual variable y_i must satisfy $y_i \leq 0$

   b) If the *i*th primal constraint is an equality constraint, the dual variable y_i is now unrestricted in sign (urs).

   c) If the *i*th primal variable is urs, the *i*th dual constraint will be an equality constraint

3.4.5 Finding the Dual of a Nonnormal Min Problem

1. Fill in Table 1 so that the primal can be read down.
2. After making the following changes, the dual can be read across in the usual fashion:

   a) If the *i*th primal constraint is a $\leq$ constraint, the corresponding dual variable x_i must satisfy $x_i \leq 0$

b) If the *i*th primal constraint is an equality constraint, the dual variable x_i is now urs.

c) If the *i*th primal variable is urs, the *i*th dual constraint will be an equality constraint

3.4.6 An Example for Economic Interpretation (Dakota Furniture)

Let x_1, x_2, x_3 be the number of desks, tables and chairs produced. Let the weekly profit be \$z. Then, we must

$$\begin{aligned}
\max z &= 60x_1+30x_2+20x_3 \\
\text{s.t.} \quad & 8x_1+6x_2+x_3 \le 48 \text{ (Lumber constraint)} \\
& 4x_1+2x_2+1.5x_3 \le 20 \text{ (Finishing hour constraint)} \\
& 2x_1+1.5x_2+0.5x_3 \le 8 \text{ (Carpentry hour constraint)} \\
& x_1,x_2,x_3 \ge 0
\end{aligned}$$

Dual

Suppose an entrepreneur wants to purchase all of Dakota's resources.

In the dual problem y_1, y_2, y_3 are the resource prices (price paid for one board ft of lumber, one finishing hour, and one carpentry hour).

\$w is the cost of purchasing the resources.

Resource prices must be set high enough to induce Dakota to sell. i.e. total purchasing cost equals total profit.

$$\begin{aligned}
\min w &= 48y_1+20y_2+8y_3 \\
\text{s.t.} \quad & 8y_1+4y_2+2y_3 \ge 60 \text{ (Desk constraint)} \\
& 6y_1+2y_2+1.5y_3 \ge 30 \text{ (Table constraint)} \\
& y_1+1.5y_2+0.5y_3 \ge 20 \text{ (Chair constraint)} \\
& y_1,y_2,y_3 \ge 0
\end{aligned}$$

4. TRANSPORTATION PROBLEMS

In general, a transportation problem is specified by the following information:

- A set of m *supply points* form which a good is shipped. Supply point i can supply at most s_i units.
- A set of n *demand points* to which the good is shipped. Demand point j must receive at least d_j units of the shipped good.
- Each unit produced at supply point i and shipped to demand point j incurs a variable cost of c_{ij}.

The relevant data can be formulated in a *transportation tableau*:

	Demand point 1	Demand point 2	.....	Demand point n	SUPPLY
Supply point 1	c_{11}	c_{12}		c_{1n}	s_1
Supply point 2	c_{21}	c_{22}		c_{2n}	s_2
.....					
Supply point m	c_{m1}	c_{m2}		c_{mn}	s_m
DEMAND	d_1	d_2		d_n	

If total supply equals total demand then the problem is said to be a **balanced transportation problem**.

4.1 FORMULATING TRANSPORTATION PROBLEMS

4.1.1 LP Representation

Let x_{ij} = number of units shipped from supply point i to demand point j
then the general LP representation of a transportation problem is

min $\sum_i \sum_j c_{ij} x_{ij}$

s.t. $\sum_j x_{ij} \leq s_i$ (i=1,2, ..., m) Supply constraints

$\sum_i x_{ij} \geq d_j$ (j=1,2, ..., n) Demand constraints

$x_{ij} \geq 0$

If a problem has the constraints given above and is a *maximization* problem, it is still a transportation problem.

4.1.2 Powerco Example for Balanced Transportation Problem

Powerco has three electric power plants that supply the needs of four cities. Each power plant can supply the following numbers of kwh of electricity: plant 1, 35 million; plant 2, 50 million; and plant 3, 40 million. The peak power demands in these cities as follows (in kwh): city 1, 45 million; city 2, 20 million; city 3, 30 million; city 4, 30 million. The costs of sending 1 million kwh of electricity from plant to city is given in the table below. To minimize the cost of meeting each city's peak power demand, formulate a balanced transportation problem in a transportation tableau and represent the problem as a LP model.

	To			
From	City 1	City 2	City 3	City 4
Plant 1	\$8	\$6	\$10	\$9
Plant 2	\$9	\$12	\$13	\$7
Plant 3	\$14	\$9	\$16	\$5

Answer

Formulation of the transportation problem

	City 1	City 2	City 3	City 4	SUPPLY
Plant 1	8	6	10	9	35
Plant 2	9	12	13	7	50
Plant 3	14	9	16	5	40
DEMAND	45	20	30	30	125

Total supply & total demand both equal 125: "balanced transport'n problem".

Representation of the problem as a LP model

x_{ij}: number of (million) kwh produced at plant *i* and sent to city *j*.

min z = 8 x11 + 6 x12 + 10 x13 + 9 x14 + 9 x21 + 12 x22 + 13 x23 + 7 x24 + 14 x31 + 9 x32 + 16 x33 + 5 x34

s.t. x11 + x12 + x13 + x14 $\leq$ 35 (supply constraints)

x21 + x22 + x23 + x24 $\leq$ 50

x31 + x32 + x33 + x34 $\leq$ 40

x11 + x21 + x31 $\geq$ 45 (demand constraints)

x12 + x22 + x32 $\geq$ 20

x13 + x23 + x33 $\geq$ 30

x14 + x24 + x34 $\geq$ 30

$x_{ij} \geq 0$ (i = 1, 2, 3; j = 1, 2, 3, 4)

4.1.3 Balancing an Unbalanced Transportation Problem

Excess Supply

If total supply exceeds total demand, we can balance a transportation problem by creating a **dummy demand point** that has a demand equal to the amount of excess supply.

Since shipments to the dummy demand point are not real shipments, they are assigned a cost of zero.

These shipments indicate unused supply capacity.

Unmet Demand

If total supply is less than total demand, actually the problem has no feasible solution. To solve the problem it is sometimes desirable to allow the possibility of leaving some demand unmet. In such a situation, *a penalty is often associated with unmet demand.* This means that a **dummy supply point** should be introduced.

4.1.4 Modified Powerco Example for Excess Supply

Suppose that demand for city 1 is 40 million kwh. Formulate a balanced transportation problem.

Answer

Total demand is 120, total supply is 125.

To balance the problem, we would add a dummy demand point with a demand of 125 – 120 = 5 million kwh.

From each plant, the cost of shipping 1 million kwh to the dummy is 0.

For details see Table 4.

Table 4. Transportation Tableau for Excess Supply

	City 1	City 2	City 3	City 4	Dummy	SUPPLY
Plant 1	8	6	10	9	0	35
Plant 2	9	12	13	7	0	50
Plant 3	14	9	16	5	0	40
DEMAND	40	20	30	30	5	125

4.1.5 Modified Powerco Example for Unmet Demand

Suppose that demand for city 1 is 50 million kwh. For each million kwh of unmet demand, there is a penalty of 80$. Formulate a balanced transportation problem.

Answer

We would add a dummy supply point having a supply of 5 million kwh representing shortage.

	City 1	City 2	City 3	City 4	SUPPLY
Plant 1	8	6	10	9	35
Plant 2	9	12	13	7	50
Plant 3	14	9	16	5	40
Dummy (Shortage)	80	80	80	80	5
DEMAND	50	20	30	30	130

4.2 FINDING BFS FOR TRANSPORT'N PROBLEMS

For a balanced transportation problem, the general LP representation may be written as:

$$\min \sum_i \sum_j c_{ij} x_{ij}$$

s.t. $\sum_j x_{ij} = s_i$ (i=1,2, ..., m) Supply constraints

$\sum_i x_{ij} = d_j$ (j=1,2, ..., n) Demand constraints

$x_{ij} \geq 0$

To find a bfs to a balanced transportation problem, we need to make the following important observation:

If a set of values for the xij's satisfies all but one of the constraints of a balanced transportation problem, the values for the xij's will automatically satisfy the other constraint.

This observation shows that when we solve a balanced transportation, we may omit from consideration any one of the problem's constraints and solve an LP having m+n-1 constraints. We arbitrarily assume that the first supply constraint is omitted from consideration.

In trying to find a bfs to the remaining m+n-1 constraints, you might think that any collection of m+n-1 variables would yield a basic solution. But this is not the case:

If the m+n-1 variables yield a basic solution, the cells corresponding to a set of m+n-1 variables contain **no loop**.

An ordered sequence of at least four different cells is called a loop if

- Any two consecutives cells lie in either the same row or same column
- No three consecutive cells lie in the same row or column
- The last cell in the sequence has a row or column in common with the first cell in the sequence

There are three methods that can be used to find a bfs for a balanced transportation problem:

1. Northwest Corner method
2. Minimum cost method
3. Vogel's method

4.2.1 Northwest Corner Method

We begin in the upper left corner of the transportation tableau and set x_{11} as large as possible (clearly, x_{11} can be no larger than the smaller of s_1 and d_1).

- If $x_{11}=s_1$, cross out the first row of the tableau. Also change d_1 to d_1-s_1.
- If $x_{11}=d_1$, cross out the first column of the tableau. Change s_1 to s_1-d_1.
- If $x_{11}=s_1=d_1$, cross out either row 1 or column 1 (but not both!).
  - If you cross out row, change d_1 to 0.
  - If you cross out column, change s_1 to 0.

Continue applying this procedure to the most northwest cell in the tableau that does not lie in a crossed out row or column.

Eventually, you will come to a point where there is only one cell that can be assigned a value. Assign this cell a value equal to its row or column demand, and cross out both the cell's row or column.

A bfs has now been obtained.

Example:

For example consider a balanced transportation problem given below (We omit the costs because they are not needed to find a bfs!).

				5
				1
				3
2	4	2	1	

Total demand equals total supply (9): this is a balanced transport'n problem.

2				3
				1
				3
X	4	2	1	

2	3			X
				1
				3
X	1	2	1	

2	3			X
	1			X
				3
X	0	2	1	

2	3			X
	1			X
	0	2	1	3
X	0	2	1	

NWC method assigned values to $m+n-1$ (3+4-1 = 6) variables. The variables chosen by NWC method can not form a loop, so a bfs is obtained.

4.2.2 Minimum Cost Method

Northwest Corner method does not utilize shipping costs, so it can yield an initial bfs that has a very high shipping cost. Then determining an optimal solution may require several pivots.

To begin the minimum cost method, find the variable with the smallest shipping cost (call it x_{ij}). Then assign x_{ij} its largest possible value, min $\{s_i, d_j\}$.

As in the NWC method, cross out row *i* or column *j* and reduce the supply or demand of the noncrossed-out of row or column by the value of x_{ij}.

Continue like NWC method (instead of assigning upper left corner, the cell with the minimum cost is assigned). See Northwest Corner Method for the details!

2	3	5	6	5
2	1	3	5	10
3	8	4	6	15
12	8	4	6	

2	3	5	6	5
2	**1** 8	3	5	2
3	8	4	6	15
12	X	4	6	

2	3	5	6	5
2 2	1 8	3	5	X
3	8	4	6	15
10	X	4	6	

2 5	3	5	6	X
2 2	1 8	3	5	X
3	8	4	6	15
5	X	4	6	

2 5	3	5	6	X
2 2	1 8	3	5	X
3 5	8	4 4	6 6	10
5	X	4	6	

4.2.3 Vogel's Method

Begin by computing for each row and column a penalty equal to the difference between the two smallest costs in the row and column.

Next find the row or column with the largest penalty.

Choose as the first basic variable the variable in this row or column that has the smallest cost.

As described in the NWC method, make this variable as large as possible, cross out row or column, and change the supply or demand associated with the basic variable (See Northwest Corner Method for the details!).

Now recomputed new penalties (using only cells that do not lie in a crossed out row or column), and repeat the procedure until only one uncrossed cell remains. Set this variable equal to the supply or demand associated with the variable, and cross out the variable's row and column.

				Supply	Row penalty
	6	7	8	10	7-6=1
	15	80	78	15	78-15=63
Demand	15	5	5		
Column penalty	15-6=9	80-7=**73**	78-8=70		

				Supply	Row penalty
	6	**7** 5	8	5	8-6=2
	15	80	78	15	78-15=63
Demand	15	X	5		
Column penalty	15-6=9	-	78-8=**70**		

				Supply	Row penalty
	6	7 5	8 5	X	-
	15	80	78	15	-
Demand	15	X	0		
Column penalty	15-6=9	-	-		

	6	7 5	8 5	X
	15 15	80	78 0	15
Demand	15	X	0	

4.3 THE TRANSPORTATION SIMPLEX METHOD

4.3.1 Steps of the Method

1. If the problem is unbalanced, balance it
2. Use one of the methods to find a bfs for the problem
3. Use the fact that $u_1 = 0$ and $u_i + v_j = c_{ij}$ for all basic variables to find the u's and v's for the current bfs.
4. If $u_i + v_j - c_{ij} \leq 0$ for all nonbasic variables, then the current bfs is optimal. If this is not the case, we enter the variable with the most positive $u_i + v_j - c_{ij}$ into the basis using the *pivoting procedure*. This yields a new bfs.
5. Using the new bfs, return to Steps 3 and 4.

For a maximization problem, proceed as stated, but replace Step 4 by the following step:

If $u_i + v_j - c_{ij} \geq 0$ for all nonbasic variables, then the current bfs is optimal. Otherwise, enter the variable with the most negative $u_i + v_j - c_{ij}$ into the basis using the *pivoting procedure*. This yields a new bfs.

Pivoting procedure

1. Find the loop (there is only one possible loop!) involving the entering variable (determined at step 4 of the transport'n simplex method) and some of the basic variables.
2. Counting *only cells in the loop*, label those that are an even number (0, 2, 4, and so on) of cells away from the entering variable as *even cells*. Also label those that are an odd number of cells away from the entering variable as *odd cells*.
3. Find the odd cell whose variable assumes the smallest value. Call this value Φ. The variable corresponding to this odd cell will leave the basis. To perform the pivot, decrease the value of each odd cell by Φ and increase the value of each even cell by Φ. The values of variables not in the loop remain unchanged. The pivot is now complete. If $\Phi = 0$, the entering variable will equal 0, and odd variable that has a current value of 0 will leave the basis.

4.3.2 Solving Powerco Example by Transportation Simplex

The problem is balanced (total supply equals total demand).

When the NWC method is applied to the Powerco example, the bfs in the following table is obtained (check: there exist $m+n-1=6$ basic variables).

	City 1	City 2	City 3	City 4	SUPPLY
Plant 1	8 35	6	10	9	35
Plant 2	9 10	12 20	13 20	7	50
Plant 3	14	9	16 10	5 30	40
DEMAND	45	20	30	30	125

u1 = 0

u1 + v1 = 8 yields v1 = 8

u2 + v1 = 9 yields u2 = 1

u2 + v2 = 12 yields v2 = 11

u2 + v3 = 13 yields v3 = 12

u3 + v3 = 16 yields u3 = 4

u3 + v4 = 5 yields v4 = 1

For each nonbasic variable, we now compute $\hat{c}_{ij} = u_i + v_j - c_{ij}$

$\hat{c}_{12} = 0 + 11 - 6 = 5$

$\hat{c}_{13} = 0 + 12 - 10 = 2$

$\hat{c}_{14} = 0 + 1 - 9 = -8$

$\hat{c}_{24} = 1 + 1 - 7 = -5$

$\hat{c}_{31} = 4 + 8 - 14 = -2$

$\hat{c}_{32} = 4 + 11 - 9 = 6$

Since $\hat{c}_{32}$ is the most positive one, we would next enter x_{32} into the basis: Each unit of x_{32} that is entered into the basis will decrease Powerco's cost by \$6.

The loop involving x_{32} is (3,2)-(3,3)-(2,3)-(2,2). $\Phi = 10$ (see table)

	City 1	City 2	City 3	City 4	SUPPLY
Plant 1	8 35	6	10	9	35
Plant 2	9 10	12 **20–Φ**	13 **20+Φ**	7	50
Plant 3	14	9 Φ	16 **10–Φ**	5 30	40
DEMAND	45	20	30	30	125

x_{33} would leave the basis. New bfs is shown at the following table:

u_i/v_j	8	11	12	7	SUPPLY
0	8 35	6	10	9	35
1	9 10	12 10	13 30	7	50
-2	14	9 10	16	5 30	40
DEMAND	45	20	30	30	125

$\hat{c}_{12} = 5$, $\hat{c}_{13} = 2$, $\hat{c}_{14} = -2$, $\hat{c}_{24} = 1$, $\hat{c}_{31} = -8$, $\hat{c}_{33} = -6$

Since $\hat{c}_{12}$ is the most positive one, we would next enter x_{12} into the basis.

The loop involving x_{12} is (1,2)-(2,2)-(2,1)-(1,1). $\Phi = 10$ (see table)

	City 1	City 2	City 3	City 4	SUPPLY
Plant 1	8 **35–Φ**	6 **Φ**	10	9	35
Plant 2	9 **10+Φ**	12 **10–Φ**	13 30	7	50
Plant 3	14	9 10	16	5 30	40
DEMAND	45	20	30	30	125

x_{22} would leave the basis. New bfs is shown at the following table:

u_i/v_j	8	6	12	2	SUPPLY
0	8 25	6 10	10	9	35
1	9 20	12	13 30	7	50
3	14	9 10	16	5 30	40
DEMAND	45	20	30	30	125

$\hat{c}_{13} = 2$, $\hat{c}_{14} = -7$, $\hat{c}_{22} = -5$, $\hat{c}_{24} = -4$, $\hat{c}_{31} = -3$, $\hat{c}_{33} = -1$

Since $\hat{c}_{13}$ is the most positive one, we would next enter x_{13} into the basis.

The loop involving x_{13} is (1,3)-(2,3)-(2,1)-(1,1). Φ = 25 (see table)

	City 1		City 2		City 3		City 4		SUPPLY
Plant 1		8		6		10		9	35
	25–Φ		10		Φ				
Plant 2		9		12		13		7	50
	20+Φ				**30–Φ**				
Plant 3		14		9		16		5	40
			10				30		
DEMAND	45		20		30		30		125

x_{11} would leave the basis. New bfs is shown at the following table:

u_i/v_j	6		6		10		2		SUPPLY
0		8		6		10		9	35
			10		25				
3		9		12		13		7	50
	45				5				
3		14		9		16		5	40
			10				30		
DEMAND	45		20		30		30		125

$\hat{c}_{11}$ = -2, $\hat{c}_{14}$ = -7, $\hat{c}_{22}$ = -3, $\hat{c}_{24}$ = -2, $\hat{c}_{31}$ = -5, $\hat{c}_{33}$ = -3

Since all $\hat{c}_{ij}$'s are negative, an optimal solution has been obtained.

Report

45 million kwh of electricity would be sent from plant 2 to city 1.

10 million kwh of electricity would be sent from plant 1 to city 2. Similarly, 10 million kwh of electricity would be sent from plant 3 to city 2.

25 million kwh of electricity would be sent from plant 1 to city 3. 5 million kwh of electricity would be sent from plant 2 to city 3.

30 million kwh of electricity would be sent from plant 3 to city 4 and

Total shipping cost is:

$$z = .9\ (45) + 6\ (10) + 9\ (10) + 10\ (25) + 13\ (5) + 5\ (30) = \$\ 1020$$

4.4 TRANSSHIPMENT PROBLEMS

Sometimes a point in the shipment process can both receive goods from other points and send goods to other points. This point is called as **transshipment point** through which goods can be transshipped on their journey from a supply point to demand point.

Shipping problem with this characteristic is a transshipment problem.

The optimal solution to a transshipment problem can be found by converting this transshipment problem to a transportation problem and then solving this transportation problem.

Remark

As stated in "Formulating Transportation Problems", we define a **supply point** to be a point that can send goods to another point but cannot receive goods from any other point.

Similarly, a **demand point** is a point that can receive goods from other points but cannot send goods to any other point.

4.4.1 Steps

1. If the problem is unbalanced, balance it

   Let *s* = total available supply (or demand) for balanced problem
2. Construct a transportation tableau as follows

   A row in the tableau will be needed for each supply point and transshipment point

   A column will be needed for each demand point and transshipment point

   Each supply point will have a supply equal to its original supply

   Each demand point will have a demand equal to its original demand

   Each transshipment point will have a supply equal to "that point's original supply + *s*"

   Each transshipment point will have a demand equal to "that point's original demand + *s*"
3. Solve the transportation problem

4.4.2 Kuruoglu Example

(Based on Winston 7.6.)

Kuruoglu manufactures refrigerators at two factories, one in Malatya and one in G.Antep. The Malatya factory can produce up to 150 refrigerators per day, and the G.Antep factory can produce up to 200 refrigerators per day. Refrigerators are shipped by air to customers in Istanbul and Izmir. The customers in each city require 130 refrigerators per day. Because of the deregulation of air fares, Kuruoglu believes that it may be cheaper to first fly some refrigerators to Ankara or Eskisehir and then fly them to their final destinations. The costs of flying a refrigerator are shown at the below table. Kuruoglu wants to minimize the total cost of shipping the required refrigerators to its customers.

Table 1. Shipping Costs for Transshipment

MTL	To					
From	Malatya	G.Antep	Ankara	Eskisehir	Istanbul	Izmir
Malatya	0	-	8	13	25	28
G.Antep	-	0	15	12	26	25
Ankara	-	-	0	6	16	17
Eskisehir	-	-	6	0	14	16
Istanbul	-	-	-	-	0	-
Izmir	-	-	-	-	-	0

Answer:

In this problem Ankara and Eskisehir are *transshipment points*.

Step 1. Balancing the problem

Total supply = 150 + 200 = 350

Total demand = 130 + 130 = 260

Dummy’s demand = 350 – 260 = 90

s = 350 (total available supply or demand for balanced problem)

Step 2. Constructing a transportation tableau

Transshipment point’s demand = Its original demand + *s* = 0 + 350 = 350

Transshipment point’s supply = Its original supply + *s* = 0 + 350 = 350

Table 2. Representation of Transship't Problem as Balanced Transp'n Problem

	Ankara	Eskisehir	Istanbul	Izmir	Dummy	Supply
Malatya	8	13	25	28	0	150
G.Antep	15	12	26	25	0	200
Ankara	0	6	16	17	0	350
Eskisehir	6	0	14	16	0	350
Demand	350	350	130	130	90	

Step 3. Solving the transportation problem

Table 3. Optimal solution for Transp'n Problem

	Ankara	Eskisehir	Istanbul	Izmir	Dummy	Supply
Malatya	8 130	13	25	28	0 20	150
G.Antep	15	12	26	25 130	0 70	200
Ankara	0 220	6	16 130	17	0	350
Eskisehir	6	0 350	14	16	0	350
Demand	350	350	130	130	90	1050

Report:

Kuruoglu should produce 130 refrigerators at Malatya, ship them to Ankara, and transship them from Ankara to Istanbul.

The 130 refrigerators produced at G.Antep should be shipped directly to İzmir.

The total shipment is 6370 MTL.

Remark: In interpreting the solution to this problem, we simply ignore the shipments to the dummy and from a point itself.

4.5 ASSIGNMENT PROBLEMS

There is a special case of transportation problems where each supply point should be assigned to a demand point and each demand should be met. This certain class of problems is called as "assignment problems". For example determining which employee or machine should be assigned to which job is an assignment problem.

4.5.1 LP Representation

An assignment problem is characterized by knowledge of the cost of assigning each supply point to each demand point: c_{ij}

On the other hand, a 0-1 integer variable x_{ij} is defined as follows

$x_{ij} = 1$ if supply point i is assigned to meet the demands of demand point j

$x_{ij} = 0$ if supply point i is not assigned to meet the demands of point j

In this case, the general LP representation of an assignment problem is

$$\min \textstyle\sum_i \sum_j c_{ij}\, x_{ij}$$

$$\text{s.t.} \quad \textstyle\sum_j x_{ij} = 1 \; (i=1,2, ..., m) \qquad \text{Supply constraints}$$

$$\textstyle\sum_i x_{ij} = 1 \; (j=1,2, ..., n) \qquad \text{Demand constraints}$$

$$x_{ij} = 0 \text{ or } x_{ij} = 1$$

4.5.2 Hungarian Method

Since all the supplies and demands for any assignment problem are integers, all variables in optimal solution of the problem must be integers. Since the RHS of each constraint is equal to 1, each x_{ij} must be a nonnegative integer that is no larger than 1, so each x_{ij} must equal 0 or 1.

Ignoring the $x_{ij} = 0$ or $x_{ij} = 1$ restrictions at the LP representation of the assignment problem, we see that we confront with a balanced transportation problem in which each supply point has a supply of 1 and each demand point has a demand of 1.

However, the high degree of degeneracy in an assignment problem may cause the Transportation Simplex to be an inefficient way of solving assignment problems.

For this reason and the fact that the algorithm is even simpler than the Transportation Simplex, the Hungarian method is usually used to solve assignment problems.

Remarks

1. To solve an assignment problem in which the goal is to maximize the objective function, multiply the profits matrix through by –1 and solve the problem as a **minimization** problem.
2. If the number of rows and columns in the cost matrix are unequal, the assignment problem is **unbalanced**. Any assignment problem should be balanced by the addition of one or more dummy points before it is solved by the Hungarian method.

Steps

1. Find the minimum cost each row of the m*m cost matrix.
2. Construct a new matrix by subtracting from each cost the minimum cost in its row
3. For this new matrix, find the minimum cost in each column
4. Construct a new matrix (reduced cost matrix) by subtracting from each cost the minimum cost in its column
5. Draw the minimum number of lines (horizontal and/or vertical) that are needed to cover all the zeros in the reduced cost matrix. If m lines are required, an optimal solution is available among the covered zeros in the matrix. If fewer than m lines are needed, proceed to next step
6. Find the smallest nonzero cost (k) in the reduced cost matrix that is uncovered by the lines drawn in Step 5
7. Subtract k from each uncovered element of the reduced cost matrix and add k to each element that is covered by two lines. Return to Step 5

4.5.3 Flight Crew Example

(Based on Winston 7.5.)

Four captain pilots (Selçuk, Serkan, Ümit, Volkan) has evaluated four flight officers (Tuncay, Önder, Servet, Kemal) according to perfection, adaptation, morale motivation in a 1-20 scale (1: very good, 20: very bad). Evaluation grades are given in Table 1. Flight Company wants to assign each flight officer to a captain pilot according to these evaluations. Determine possible flight crews.

Table 1. Times (necessary weeks) for Papers

	Tuncay	Önder	Servet	Kemal
Selçuk	2	4	6	10
Serkan	2	12	6	5
Ümit	7	8	3	9
Volkan	14	5	8	7

Answer:

Step 1. For each row in Table 1, we find the minimum cost: 2, 2, 3, and 5 respectively

Step 2 & 3. We subtract the row minimum from each cost in the row obtaining Table 2. For this new matrix, we find the minimum cost in each column

Table 2. Cost matrix after row minimums are subtracted

	0	2	4	8
	0	10	4	3
	4	5	0	6
	9	0	3	2
Column minimum	0	0	0	2

Step 4. We now subtract the column minimum from each cost in the column obtaining reduced cost matrix in Table 3.

Table 3. Reduced cost matrix

0	2	4	6
0	10	4	1
4	5	0	4
9	0	3	0

Step 5. As shown in Table 4, lines through row 3, row 4, and column 1 cover all the zeros in the reduced cost matrix. The minimum number of lines for this operation is 3. Since fewer than four lines are required to cover all the zeros, solution is not optimal: we proceed to next step.

Table 4. Reduced cost matrix with lines covering zeros

0	2	4	6
0	10	4	1
4	5	0	4
9	0	3	0

Step 6 & 7. The smallest uncovered cost equals 1. We now subtract 1 from each uncovered cost, add 1 to each twice-covered cost, and obtain Table 6.

Table 6. Resulting matrix

0	1	3	5
0	9	3	0
5	5	0	4
10	0	3	0

Four lines are now required to cover all the zeros: An optimal s9olution is available. Observe that the only covered 0 in column 3 is x_{33}, and in column 2 is x_{42}. As row 5 can not be used again, for column 4 the remaining zero is x_{24}. Finally we choose x_{11}.

Report:

CP Selçuk should fly with FO Tuncay; CP Serkan should fly with FO Kemal; CP Ümit should fly with FO Servet; and CP Volkan should fly with FO Önder.

5. INTEGER PROGRAMMING

When formulating LP's we often found that, strictly, certain variables should have been regarded as taking integer values but, for the sake of convenience, we let them take fractional values reasoning that the variables were likely to be so large that any fractional part could be neglected. While this is acceptable in some situations, in many cases it is not, and in such cases we must find a numeric solution in which the variables take integer values.

Problems in which this is the case are called ***integer programs*** (***IP's***) and the subject of solving such programs is called ***integer programming*** (also referred to by the initials ***IP***).

IP's occur frequently because many decisions are essentially discrete (such as yes/no, do/do not) in that one or more options must be chosen from a finite set of alternatives.

An integer programming problem in which all variables are required to be integer is called a ***pure integer programming problem***.

If some variables are restricted to be integer and some are not then the problem is a ***mixed integer programming problem***.

The case where the integer variables are restricted to be 0 or 1 comes up surprising often. Such problems are called ***pure (mixed) 0-1 programming problems*** or ***pure (mixed) binary integer programming problems*****.**

5.1 FORMULATING IP

5.1.1 Budgeting Problems

5.1.1.1 Capital Budgeting Example

Suppose we wish to invest $14,000. We have identified four investment opportunities. Investment 1 requires an investment of $5,000 and has a present value (a time-discounted value) of $8,000; investment 2 requires $7,000 and has a value of $11,000; investment 3 requires $4,000 and has a value of $6,000; and investment 4 requires $3,000 and has a value of $4,000. Into which investments should we place our money so as to maximize our total present value?

Answer

As in LP, our first step is to decide on our variables. This can be much more difficult in IP because there are very clever or tricky ways to use integrality restrictions.

In this case, we will use a 0-1 variable x_j for each investment.
If x_j is 1 then we will make investment *j*.
If it is 0, we will not make the investment.
This leads to the 0-1 programming problem:

$$\begin{array}{ll} \text{Maximize} & 8\,x_1 + 11\,x_2 + 6\,x_3 + 4\,x_4 \\ \text{Subject to} & 5\,x_1 + 7\,x_2 + 4\,x_3 + 3\,x_4 \leq 14 \\ & x_j = 0 \text{ or } 1 \qquad j = 1, \ldots 4 \end{array}$$

Now, a straightforward "bang for buck" (taking ratios of objective coefficient over constraint coefficient) suggests that investment 1 is the best choice.
Ignoring integrality constraints, the optimal linear programming solution is

$x_1 = 1$, $x_2 = 1$, $x_3 = 0.5$, and $x_4 = 0$ for a value of $22,000.

Unfortunately, this solution is not integral. Rounding x_3 down to 0 gives a feasible solution with a value of $19,000.
There is a better integer solution, however, of $x_1 = 0$, $x_2 = x_3 = x_4 = 1$ for a value of $21,000.
This example shows that rounding does not necessarily give an optimal value.

5.1.1.2 Multiperiod Example

There are four possible projects, which each run for 3 years and have the following characteristics.

		Capital requirements		
Project	Return	Year1	Year2	Year3
1	0.2	0.5	0.3	0.2
2	0.3	1	0.5	0.2
3	0.5	1.5	1.5	0.3
4	0.1	0.1	0.4	0.1
Available capital		3.1	2.5	0.4

Which projects would you choose in order to maximize the total return?

Answer

x_j is 1 if we decide to do project j; x_j is 0 otherwise (i.e. not do project j).

This leads to the 0-1 programming problem:

$$\begin{array}{ll} \text{Maximize} & 0.2\,x_1 + 0.3\,x_2 + 0.5\,x_3 + 0.1\,x_4 \\ \text{Subject to} & 0.5\,x_1 + 1\quad x_2 + 1.5\,x_3 + 0.1\,x_4 \leq 3.1 \\ & 0.3\,x_1 + 0.8\,x_2 + 1.5\,x_3 + 0.4\,x_4 \leq 2.5 \\ & 0.2\,x_1 + 0.2\,x_2 + 0.3\,x_3 + 0.1\,x_4 \leq 0.4 \\ & x_j = 0 \text{ or } 1 \qquad j = 1, \ldots 4 \end{array}$$

Remarks

You will note that the objective and constraints are linear. In this course we deal only with linear integer programs (IP's with a linear objective and linear constraints). It is plain though that there do exist non-linear integer programs - these are, however, outside the scope of this course.

Whereas before in formulating LP's if we had integer variables we assumed that we could ignore any fractional parts it is clear that we cannot do so in this problem e.g. what would be the physical meaning of a numeric solution with x_1=0.4975?

5.1.1.3 Capital Budgeting Extension

There are a number of additional constraints we might want to add. Logical restrictions can be enforced using 0-1 variables.

For instance, consider the following constraints:

<u>We can only make two investments</u>

$x_1 + x_2 + x_3 + x_4 \leq 2$

Any choice of 3 or 4 investments will have $x_1 + x_2 + x_3 + x_4 \geq 3$

<u>If investment 2 is made, investment 4 must also be made</u>

$x_2 \leq x_4$ or $x_2 - x_4 \leq 0$

If x_2 is 1, then x_4 is also 1 as we desire; if x_2 is 0, then there is no restriction for x_4 (x_4 is 0 or 1)

<u>If investment 1 is made, investment 3 cannot be made</u>

$x_1 + x_3 \leq 1$

If x_1 is 1, then x_3 is 0 as we desire; if x_1 is 0, then there is no restriction for x_3 (x_3 is 0 or 1)

<u>Either investment 1 or investment 2 must be done</u>

$x_1 + x_2 = 1$

If x_1 is 1, then x_2 is 0 (only investment 1 is done); if x_1 is 0, then x_2 is 1 (only investment 2 is done)

5.1.2 Knapsack Problems

Any IP that has only one constraint is referred to as a knapsack problem.

Furthermore, the coefficients of this constraint and the objective are all non-negative.

For example, the following is a knapsack problem:

$$\begin{array}{ll} \text{Maximize} & 8\,x_1 + 11\,x_2 + 6\,x_3 + 4\,x_4 \\ \text{Subject to} & 5\,x_1 + 7\,x_2 + 4\,x_3 + 3\,x_4 \leq 14 \\ & x_j = 0 \text{ or } 1 \quad j = 1, \dots 4 \end{array}$$

The traditional story is that there is a knapsack (here of capacity 14). There are a number of items (here there are four items), each with a size and a value (here item 2 has size 7 and value 11).

The objective is to maximize the total value of the items in the knapsack.

Knapsack problems are nice because they are (usually) easy to solve.

5.1.3 Fixed Charge Problems

An important trick can be used to formulate many production and location problems as IP.

Here Gandhi example is given as a production problem and Lockbox example is given as a location problem.

5.1.3.1 Gandhi Example (Production)

(Winston 9.2., p. 470)

Gandhi Co makes shirts, shorts, and pants using the limited labor and cloth described below. In addition, the machinery to make each product must be rented.

	Shirts	Shorts	Pants	Total Avail.
Labor (hrs/wk)	3	2	6	150
Cloth (m^2/wk)	4	3	4	160
Rent for machine ($/wk)	200	150	100	
Variable unit cost	6	4	8	
Sale Price	12	8	15	

Answer

Let x_j be number of clothing produced.

Let y_j be 1 if any clothing j is manufactured and 0 otherwise.

Profit = Sales revenue – Variable Cost – Costs of renting machinery

For example the profit from shirts is $z_1 = (12 - 6)\,x_1 - 200\,y_1$.

Since supply of labor and cloth is limited, Gandhi faces two constraints.

To ensure $x_j > 0$ forces $y_j = 1$, we include the additional constraints

$$x_j \leq M_j\, y_j$$

From the cloth constraint at most 40 shirts can be produced ($M_1 = 40$), so the additional constraint for shirts is not an additional limit on x_1 (If M_1 were not chosen large (say $M_1 = 10$), then the additional constraint for shirts would unnecessarily restrict the value of x_1).

From the cloth constraint at most 53 shorts can be produced ($M_2 = 53$) and from the labor constraint at most 25 pants can be produced ($M_3 = 25$).

We thus get the mixed (binary) integer problem:

$$
\begin{array}{ll}
\max & 6x_1 + 4x_2 + 7x_3 - 200y_1 - 150y_2 - 100y_3 \\
\text{st} & 3x_1 + 2x_2 + 6x_3 \leq 150 \quad \text{(Labor constraint)} \\
& 4x_1 + 3x_2 + 4x_3 \leq 160 \quad \text{(Cloth constraint)} \\
& x_1 \leq 40y_1 \quad \text{(Shirt production and machinery constraint)} \\
& x_2 \leq 53y_2 \quad \text{(Short production and machinery constraint)} \\
& x_3 \leq 25y_3 \quad \text{(Pant production and machinery constraint)} \\
& x_1, x_2, x_3 \geq 0 \text{ and integer} \\
& y_1, y_2, y_3 = 0 \text{ or } 1
\end{array}
$$

The optimal solution to the Gandhi problem is z = \$ 75, $x_3 = 25$, $y_3 = 1$.
Thus, they should produce 25 pants and rent a pants machinery.

5.1.3.2 The Lockbox Example (Location)

(Adapted to Winston 9.2., p. 473)

Consider a national firm that receives checks from all over the United States. There is a variable delay from when the check is postmarked (and hence the customer has met her obligation) and when the check clears (the firm can use the money). It is in the firm's interest to have the check clear as quickly as possible since then the firm can use the money. To speed up this clearing, firms open offices (lockboxes) in different cities to handle the checks.

For example, suppose we receive payments from four regions (West, Midwest, East, and South). The average daily value from each region is as follows: $70,000 from the West, $50,000 from the Midwest, $60,000 from the East, and $40,000 from the South. We are considering opening lockboxes in L.A., Chicago, New York, and/or Atlanta. Operating a lockbox costs $50,000 per year. The average days from mailing to clearing is given in the table.

	LA	Chicago	NY	Atlanta
West	2	6	8	8
Midwest	6	2	5	5
East	8	5	2	5
South	8	5	5	2

Which lockboxes should we open? Formulate an IP that we can use to minimize the sum of costs due to lost interest and lockbox operations. Assume that each region must send all its money to a single city and investment rate is 20%.

Answer

First we must calculate the losses due to lost interest for each possible assignment. For example, if the West sends to New York, then on average there will be $560,000 (= 8 * $70.000) in process on any given day. Assuming an investment rate of 20%, this corresponds to a yearly loss of $112,000. We can calculate the losses for the other possibilities in a similar fashion to get the following table.

	LA	Chicago	NY	Atlanta
West	28	84	112	112
Midwest	60	20	50	50
East	96	60	24	60
South	64	40	40	16

Let y_j be a 0-1 variable that is 1 if lockbox j is opened and 0 if it is not.

Let x_{ij} be 1 if region i sends to lockbox j; and 0 otherwise.

Our objective is to minimize our total yearly costs. This is:

$28\ x_{11} + 84\ x_{12} + 112\ x_{13} + \ldots + 50\ y_1 + 50\ y_2 + 50\ y_3 + 50\ y_4$

One set of constraint is that each region must be assigned to one lockbox:

SUM [j=1 to n] $x_{ij} = 1$ for all i

(SUM[j=1 to n] should be read as "sum over all integer values of j from 1 to n inclusive")

A more difficult set of constraint is that a region can only be assigned to an open lockbox. This can be written as

$x_{1j} + x_{2j} + x_{3j} + x_{4j} \leq M\ y_j$

M is any number that should be at least 4 as there are four regions.

(Suppose we do not open LA lockbox; then y_1 is 0, so all of x_{11}, x_{21}, x_{31}, and x_{41} must also be 0. If y_1 is 1, then there is no restriction on the x values.)

For this problem we would have twenty variables (four y variables, sixteen x variables) and eight constraints. This gives the following 0-1 linear program:

$$
\begin{aligned}
\text{Min}\quad & 28\ x_{11} + 84\ x_{12} + 112\ x_{13} + 112\ x_{14} \\
& + 60\ x_{21} + 20\ x_{22} + 50\ x_{23} + 50\ x_{24} \\
& + 96\ x_{31} + 60\ x_{32} + 24\ x_{33} + 60\ x_{34} \\
& + 64\ x_{41} + 40\ x_{42} + 40\ x_{43} + 16\ x_{44} \\
& + 50\ y_1 + 50\ y_2 + 50\ y_3 + 50\ y_4 \\
\text{st}\quad & x_{11} + x_{12} + x_{13} + x_{14} = 1 \\
& x_{21} + x_{22} + x_{23} + x_{24} = 1 \\
& x_{31} + x_{32} + x_{33} + x_{34} = 1 \\
& x_{41} + x_{42} + x_{43} + x_{44} = 1 \\
& x_{11} + x_{21} + x_{31} + x_{41} \leq 4y_1 \\
& x_{12} + x_{22} + x_{32} + x_{42} \leq 4y_2 \\
& x_{13} + x_{23} + x_{33} + x_{43} \leq 4y_3 \\
& x_{14} + x_{24} + x_{34} + x_{44} \leq 4y_4 \\
& \text{All } x_{ij} \text{ and } y_j = 0 \text{ or } 1
\end{aligned}
$$

There are other formulations, however.

For instance, instead of four constraints of the form

$$x_{1j} + x_{2j} + x_{3j} + x_{4j} \leq M\, y_j$$

consider the sixteen constraints of the form:

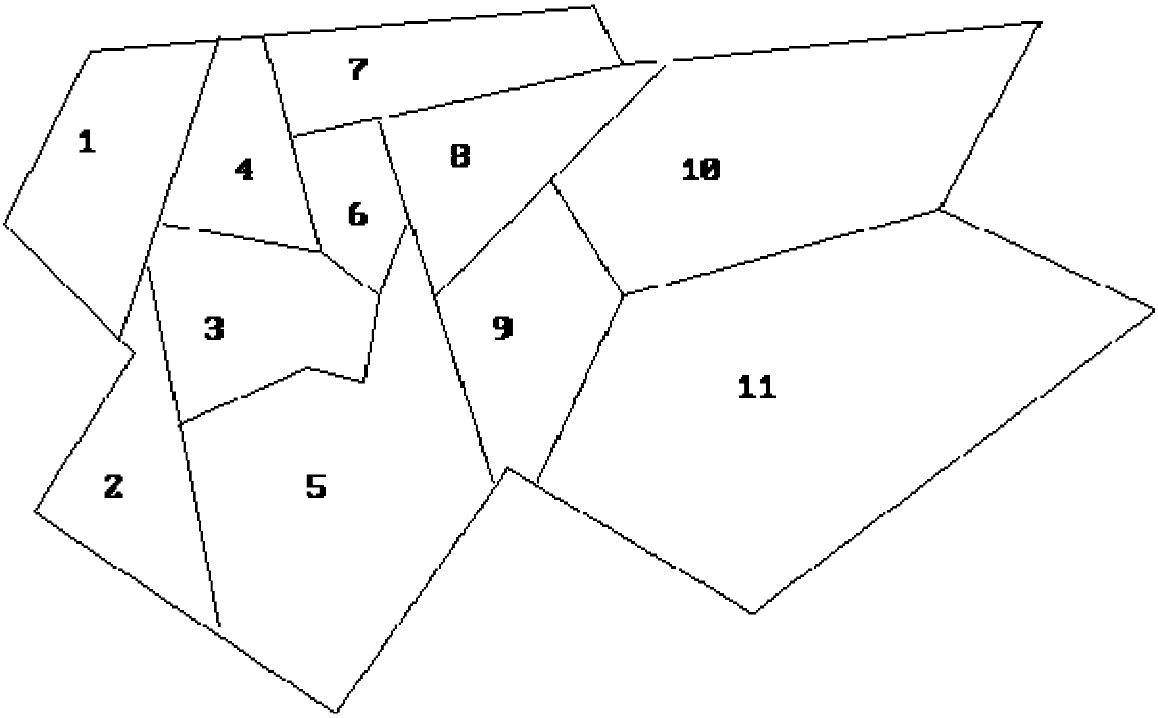
$$x_{ij} \leq y_j \qquad i = 1, 2, 3, 4;\; j = 1, 2, 3, 4$$

These constraints also force a region to only use open lockboxes.
(Suppose y_j is 0, so by using four corresponding constraints all of x_{1j}, x_{2j}, x_{3j}, and x_{4j} must also be 0. If y_j is 1, then there is no restriction on the x values.
Another point of view: If $x_{ij} = 1$, then $y_j = 1$ as desired. Also if $x_{1j} = x_{2j} = x_{3j} = x_{4j} = 0$, then there is no restriction on the y values. The act of minimizing costs will result $y_j = 0$.

It might seem that a larger formulation (twenty variables and twenty constraints) is less efficient and therefore should be avoided. This is not the case! If we solve the linear program with the above constraints, we have an integer solution as an optimal LP solution, which must therefore be optimal!

5.1.4 Set Covering Problems

To illustrate this model, consider the following location problem: A county is reviewing the location of its fire stations. The county is made up of a number of cities, as illustrated in the following figure.



A fire station can be placed in any city. It is able to handle the fires for both its city and any adjacent city (any city with a non-zero border with its home city). The objective is to minimize the number of fire stations used.

Answer

We can create one variable x_j for each city $_j$.

This variable will be 1 if we place a station in the city, and will be 0 otherwise. This leads to the following formulation

$$
\begin{array}{llr}
\text{Min} & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} & \\
\text{st} & x_1 + x_2 + x_3 + x_4 & \geq 1 \text{ (city 1)} \\
 & x_1 + x_2 + x_3 + x_5 & \geq 1 \text{ (city 2)} \\
 & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 & \geq 1 \text{ (city 3)} \\
 & x_1 + x_3 + x_4 + x_6 + x_7 & \geq 1 \text{ (city 4)} \\
 & x_2 + x_3 + x_5 + x_6 + x_8 + x_9 & \geq 1 \text{ (city 5)} \\
 & x_3 + x_4 + x_5 + x_6 + x_7 + x_8 & \geq 1 \text{ (city 6)} \\
 & x_4 + x_6 + x_7 + x_8 & \geq 1 \text{ (city 7)} \\
 & x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} & \geq 1 \text{ (city 8)} \\
 & x_5 + x_8 + x_9 + x_{10} + x_{11} & \geq 1 \text{ (city 9)} \\
 & + x_8 + x_9 + x_{10} + x_{11} & \geq 1 \text{ (city 10)} \\
 & + x_9 + x_{10} + x_{11} & \geq 1 \text{ (city 11)} \\
 & \text{All } x_j = 0 \text{ or } 1 &
\end{array}
$$

The first constraint states that there must be a station either in city 1 or in some adjacent city. Notice that the constraint coefficient a_{ij} is 1 if city i is adjacent to city j or if i=j and 0 otherwise.

The jth column of the constraint matrix represents the set of cities that can be served by a fire station in city j. We are asked to find a set of such subsets j that *covers* the set of all cities in the sense that every city appears in the service subset associated with *at least* one fire station

5.1.5 Either - Or Constraints

The following situation commonly occurs in mathematical programming problems. We are given two constraints of the form:

$f(x_1, x_2,\ldots, x_n) \leq 0$ (1)

$g(x_1, x_2,\ldots, x_n) \leq 0$ (2)

We want to ensure that ***at least*** one of (1) and (2) is satisfied, often called ***either-or*** constraints.

Adding the two constraints (1') and (2') to the formulation will ensure our aim:

$f(x_1, x_2,\ldots, x_n) \leq M\,y$ (1')

$g(x_1, x_2,\ldots, x_n) \leq M\,(1 - y)$ (2')

Here y is a 0-1 variable, and M is a number chosen large enough to ensure that $f(x_1, x_2,\ldots, x_n) \leq M$ and $g(x_1, x_2,\ldots, x_n) \leq M$ are satisfied for all values of x_j's that satisfy the other constraints in the problem.

Suppose y = 0, then (1) and possibly (2) must be satisfied.
If y = 1, then (2) and possibly (1) must be satisfied.

Example

Suppose 1.5 tons of steel and 30 hours of labor are required for production of one compact car. At present, 6.000 tons of steel and 60.000 hours of labor are available. For an economically feasible production, at least 1000 cars of compact car must be produced.

Constraint: $x_1 \leq 0$ or $x_1 \geq 1000$

$[f(x_1, x_2,\ldots, x_n) = x_1;\ g(x_1, x_2,\ldots, x_n) = 1000 - x_1]$

Sign restriction: $x_1 \geq 0$ and integer

We can replace this constraint by the following pair of linear constraints:

$$x_1 \leq M_1\,y_1$$

$$1000 - x_1 \leq M_1\,(1 - y_1)$$

$$y_1 = 0 \text{ or } 1$$

$$M_1 = \min(6.000/1.5,\ 60.000/30) = 2000$$

5.1.6 If - Then Constraints

In many mathematical programming applications, the following situation occurs.

We want to ensure that a constraint $f(x_1, x_2,\ldots, x_n) > 0$ implies the constraint $g(x_1, x_2,\ldots, x_n) \geq 0$:

If $f(x_1, x_2,\ldots, x_n) > 0$ is satisfied, then the constraint $g(x_1, x_2,\ldots, x_n) \geq 0$ must be satisfied. While if $f(x_1, x_2,\ldots, x_n) > 0$ is not satisfied, then $g(x_1, x_2,\ldots, x_n) \geq 0$ may or may not be satisfied.

To ensure this, we include the following constraints in the formulation:

$$-g(x_1, x_2,\ldots, x_n) \leq M y \qquad (1)$$

$$f(x_1, x_2,\ldots, x_n) \leq M(1 - y) \qquad (2)$$

Here y is a 0-1 variable, and M is a large positive number.

M must be chosen large enough so that $-g \leq M$ and $f \leq M$ hold for all values of x_j's that satisfy the other constraints in the problem.

Observe that if $f > 0$, then (2) can be satisfied only if $y = 0$. (1) implies $-g \leq 0$ or $g \geq 0$, which is the desired result.

Example

For the lockbox problem, suppose we add the following constraint: If customers in region 1 send their payments to city 1, no other customers may send their payments to city 1. Mathematically,

If $x_{11} = 1$, then $x_{21} = x_{31} = x_{41} = 0$

Since all variables are 0-1, we may write this implication as:

If $x_{11} > 0$, then $x_{21} + x_{31} + x_{41} \leq 0$ (or $-x_{21} - x_{31} - x_{41} \geq 0$)

If we define $f = x_{11}$ and $g = -x_{21} - x_{31} - x_{41}$ we can use (1) and (2) to express the implication by the following two constraints:

$$x_{21} + x_{31} + x_{41} \leq My$$

$$x_{11} \leq M(1 - y)$$

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

Since $-g$ and f can never exceed 3, we can choose M as 3.

5.1.7 Traveling Salesperson Problems

A salesperson must visit each of ten cities once before returning to his home. "What ordering of the cities minimizes the total distance the salesperson must travel before returning home?" problem is called the ***traveling salesperson problem (TSP)***, not surprisingly.

An IP Formulation of the TSP

Suppose there are N cities.

For $i \neq j$ let c_{ij} = distance form city i to city j and

Let c_{ii} = M (a very large number relative to actual distances)

Also define x_{ij} as a 0-1 variable as follows:

x_{ij} = 1 if the solution to TSP goes from city i to city j;

x_{ij} = 0 otherwise

The formulation of the TSP is:

Min	SUM[i] SUM[j] c_{ij} x_{ij}	
st	SUM[i] $x_{ij} = 1$	for all j
	SUM[j] $x_{ij} = 1$	for all i
	$u_i - u_j + N x_{ij} \leq N - 1$	for $i \neq j$; i = 2, 3, …, N; j = 2, 3, …, N

All x_{ij} = 0 or 1, All $u_i \geq 0$

(SUM[j] should be read as "sum over all integer values of j from 1 to n inclusive")

The first set of constraints ensures that s/he arrives once at each city.

The second set of constraints ensures that s/he leaves each city once.

The third set of constraints ensure the following:

Any set of x_{ij}'s containing a subtour will be infeasible

Any set of x_{ij}'s that forms a tour will be feasible

REMARK

The formulation of an IP whose solution will solve a TSP becomes unwieldy and inefficient for large TSPs. When using branch and bound methods to solve TSPs with many cities, large amounts of computer time may be required. For this reason, heuristics, which quickly lead to a good (but not necessarily optimal solution to a TSP, are often used.

5.2 SOLVING INTEGER PROGRAMS

We have gone through a number of examples of integer programs at the “Formulating IP Problems” section. A natural question is ``How can we get solutions to these models?''. There are two common approaches. Historically, the first method developed was based on ***cutting planes*** (adding constraints to force integrality). In the last twenty years or so, however, the most effective technique has been based on dividing the problem into a number of smaller problems in a *tree search* method called ***branch and bound***. Recently (the last ten years or so), cutting planes have made a resurgence in the form of *facets* and *polyhedral characterizations*.
Actually, all these approaches involve solving a series of LP. For solving LP’s we have *general purpose* (independent of the LP being solved) and *computationally effective* (able to solve large LP's) algorithms (simplex or interior point). For solving IP's *no* similar general purpose and computationally effective algorithms exist.

Solution methods for IP's can be categorized as:

- *General purpose* (will solve any IP) but potentially computationally ineffective (will only solve relatively small problems); or
- *Special purpose* (designed for one particular type of IP problem) but potentially computationally more effective.

Solution methods for IP's can also be categorized as:

- *Optimal*
- *Heuristic*

An optimal algorithm is one which (mathematically) *guarantees* to find the optimal solution.

It may be that we are not interested in the optimal solution:

- because the size of problem that we want to solve is beyond the computational limit of known optimal algorithms within the computer time we have available; or
- we could solve optimally but feel that this is not worth the effort (time, money, etc) we would expend in finding the optimal solution.

In such cases we can use a heuristic algorithm - that is an algorithm that should hopefully find a feasible solution that, in objective function terms, is close to the optimal solution. In fact it is often the case that a well-designed heuristic algorithm can give good quality (near-optimal) results.

Hence we have four categories that we potentially need to consider for solution algorithms:

- General Purpose, Optimal
  Enumeration, branch and bound, cutting plane
- General Purpose, Heuristic
  Running a general purpose optimal algorithm and terminating after a specified time
- Special Purpose, Optimal (beyond the scope of this course)
  Tree search approaches based upon generating bounds via dual ascent, lagrangean relaxation
- Special Purpose, Heuristic (beyond the scope of this course)
  Bound based heuristics, tabu search, simulated annealing, population heuristics (e.g. genetic algorithms), interchange

5.2.1 LP Relaxation (Relationship to LP)

For any IP we can generate an LP by taking the same objective function and same constraints but with the requirement that variables are integer replaced by appropriate continuous constraints:

“x_i = 0 or 1” can be replaced by $x_i >= 0$ and $x_i <= 1$

“$x_i \geq 0$ and integer” can be replaced by $x_i >= 0$

The LP obtained by omitting all integer and 0-1 constraints on variables is called the ***LP Relaxation of the IP***. We can then solve this linear relaxation (LR) of the original IP.

If LR is optimized by integer variables (solution is turned out to have all variables taking integer values at the optimal solution), then that solution is feasible and optimal for IP (*naturally integer LP*)

Since LR is less constrained than IP, the following are immediate:

- If IP is a maximization, the optimal objective value for LR is greater than or equal to that of IP.
- If IP is a minimization, the optimal objective value for LR is less than or equal to the optimal objective for IP.
- If LR is infeasible, then so is IP.

So solving LR does give some information: it gives a bound on the optimal value, and, if we are lucky, may give the optimal solution to IP. We saw, however, that rounding the solution of LR will not in general give the optimal solution of IP. In fact, for some problems it is difficult to round and even get a feasible solution.

In general the solution process used in many LP based IP packages is

- specify the IP: objective, constraints, integer variables (typically x_j = integer between 0 and n).
- automatically generate the LP relaxation: same objective as IP, same constraints as IP with the addition of $0<=x_j<=n$, variables as before but no longer required to be integer
- use the branch and bound to generate the optimal solution to the IP

5.2.2 Enumeration

Unlike LP (where variables took continuous values (>=0)) in IP's (where all variables are integers) each variable can only take a finite number of discrete (integer) values. Hence the obvious solution approach is simply to ***enumerate*** all these possibilities - calculating the value of the objective function at each one and choosing the (feasible) one with the optimal value.

For example for the multi-period capital budgeting problem,

Maximize	$0.2\ x_1 + 0.3\ x_2 + 0.5\ x_3 + 0.1\ x_4$
Subject to	$0.5\ x_1 + 1\quad x_2 + 1.5\ x_3 + 0.1\ x_4 \leq 3.1$
	$0.3\ x_1 + 0.8\ x_2 + 1.5\ x_3 + 0.4\ x_4 \leq 2.5$
	$0.2\ x_1 + 0.2\ x_2 + 0.3\ x_3 + 0.1\ x_4 \leq 0.4$
	$x_j = 0$ or $1 \qquad j = 1, \dots 4$

there are 2^4=16 possible solutions. These are:

x_1	x_2	x_3	x_4	
0	0	0	0	do no projects
0	0	0	1	do one project
0	0	1	0	
0	1	0	0	
1	0	0	0	
0	0	1	1	do two projects
0	1	0	1	
1	0	0	1	
0	1	1	0	
1	0	1	0	
1	1	0	0	
1	1	1	0	do three projects
1	1	0	1	
1	0	1	1	
0	1	1	1	
1	1	1	1	do four projects

Hence for our example we merely have to examine 16 possibilities before we know precisely what the best possible solution is. This example illustrates a general truth about integer programming:

What makes solving the problem easy when it is small is precisely what makes it hard very quickly as the problem size increases

This is simply illustrated: suppose we have 100 integer variables each with two possible integer values then there are 2x2x2x ... x2 = 2^{100} (approximately 10^{30}) possibilities which we have to enumerate (obviously many of these possibilities will be infeasible, but until we generate one we cannot check it against the constraints to see if it is feasible).

Be clear here - ***conceptually*** there is not a problem - simply enumerate all possibilities and choose the best one. But ***computationally*** (numerically) this is just impossible.

IP nowadays is often called "*combinatorial optimization*" indicating that we are dealing with optimization problems with an extremely large (combinatorial) increase in the number of possible solutions as the problem size increases.

5.2.3 Branch and Bound

The most effective general purpose optimal algorithm is ***LP-based tree search*** (also widely being called as ***branch and bound***).

This is *a way of systematically enumerating* feasible solutions such that the optimal integer solution is found.

Where this method differs from the enumeration method is that *not all* the feasible solutions are enumerated but only a fraction (hopefully a small fraction) of them. However we can still *guarantee* that we will find the optimal integer solution. The method was first put forward in the early 1960's by Land and Doig.

5.2.3.1 Example

Consider our example multi-period capital budgeting problem:

Maximize	$0.2\,x_1 + 0.3\,x_2 + 0.5\,x_3 + 0.1\,x_4$
Subject to	$0.5\,x_1 + 1\,x_2 + 1.5\,x_3 + 0.1\,x_4 \leq 3.1$
	$0.3\,x_1 + 0.8\,x_2 + 1.5\,x_3 + 0.4\,x_4 \leq 2.5$
	$0.2\,x_1 + 0.2\,x_2 + 0.3\,x_3 + 0.1\,x_4 \leq 0.4$
	$x_j = 0$ or 1 $\quad j = 1, \ldots 4$

What made this problem difficult was the fact that the variables were restricted to be integers (zero or one).

If the variables had been allowed to be fractional (takes all values between zero and one for example) then we would have had an LP which we could easily solve. Suppose that we were to solve this LP relaxation of the problem [replace $x_j = 0$ or 1 $j=1,...,4$ by $0 <= x_j <= 1$ $j=1,...,4$]. Then using any LP package we get $x_2=0.5$, $x_3=1$, $x_1=x_4=0$ of value 0.65 (i.e. the objective function value of the optimal linear programming solution is 0.65).

As a result of this we now know something about the optimal integer solution, namely that it is <= 0.65, i.e. this value of 0.65 is an ***upper bound*** on the optimal integer solution.

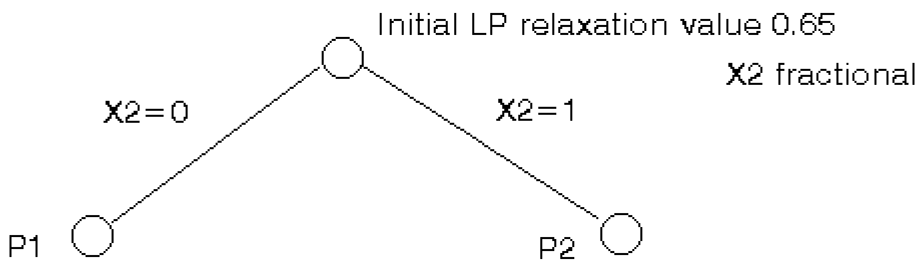
This is because when we relax the integrality constraint we (as we are maximizing) end up with a solution value at least that of the optimal integer solution (and maybe better).

Consider this LP relaxation solution. We have a variable x_2 which is fractional when we need it to be integer. To resolve this we can generate two new problems:

P1: original LP relaxation plus $x_2=0$

P2: original LP relaxation plus $x_2=1$

Then we will claim that the optimal integer solution to the original problem is contained in one of these two new problems. This process of taking a fractional variable and explicitly constraining it to each of its integer values is known as ***branching***. It can be represented diagrammatically as below (in a tree diagram, which is how the name *tree search* arises).

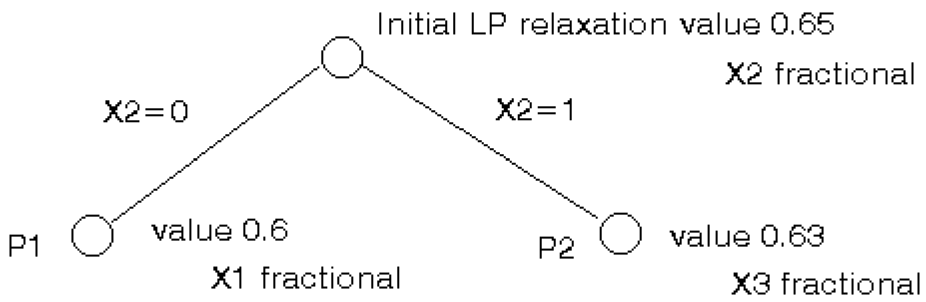


We now have two new LP relaxations to solve. If we do this we get:

P1 solution is $x_1=0.5$, $x_3=1$, $x_2=x_4=0$ of value 0.6

P2 solution is $x_2=1$, $x_3=0.67$, $x_1=x_4=0$ of value 0.63

This can be represented diagrammatically as below.



To find the optimal integer solution we just repeat the process, choosing one of these two problems, choosing one fractional variable and generating two new problems to solve.

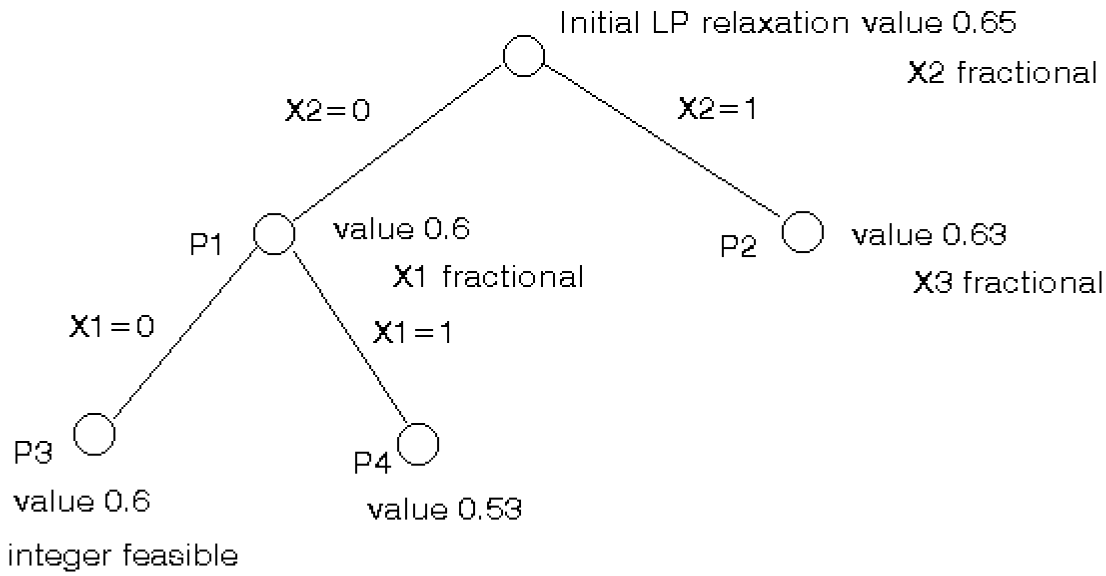
Choosing problem P1 we branch on x_1 to get our list of LP relaxations as:

P3 (P1 plus x_1=0) solution x_3=x_4=1, x_1=x_2=0 of value 0.6

P4 (P1 plus x_1=1) solution x_1=1, x_3=0.67, x_2=x_4=0 of value 0.53

P2 solution x_2=1, x_3=0.67, x_1=x_4=0 of value 0.63

This can again be represented diagrammatically as below.



At this stage we have identified an integer feasible solution of value 0.6 at P3. There are no fractional variables so no branching is necessary and P3 can be dropped from our list of LP relaxations.

Hence we now have new information about our optimal (best) integer solution, namely that it lies between 0.6 and 0.65 (inclusive).

Consider P4, it has value 0.53 and has a fractional variable (x_3). However if we were to branch on x_3 any objective function solution values we get after branching can never be better (higher) than 0.53. As we already have an integer feasible solution of value 0.6, P4 can be dropped from our list of LP relaxations since branching from it could never find an improved feasible solution. This is known as ***bounding*** - using a known feasible solution (as a ***lower bound***) to identify that some relaxations are not of any interest and can be discarded.

Hence we are just left with:

P2 solution x_2=1, x_3=0.67, x_1=x_4=0 of value 0.63

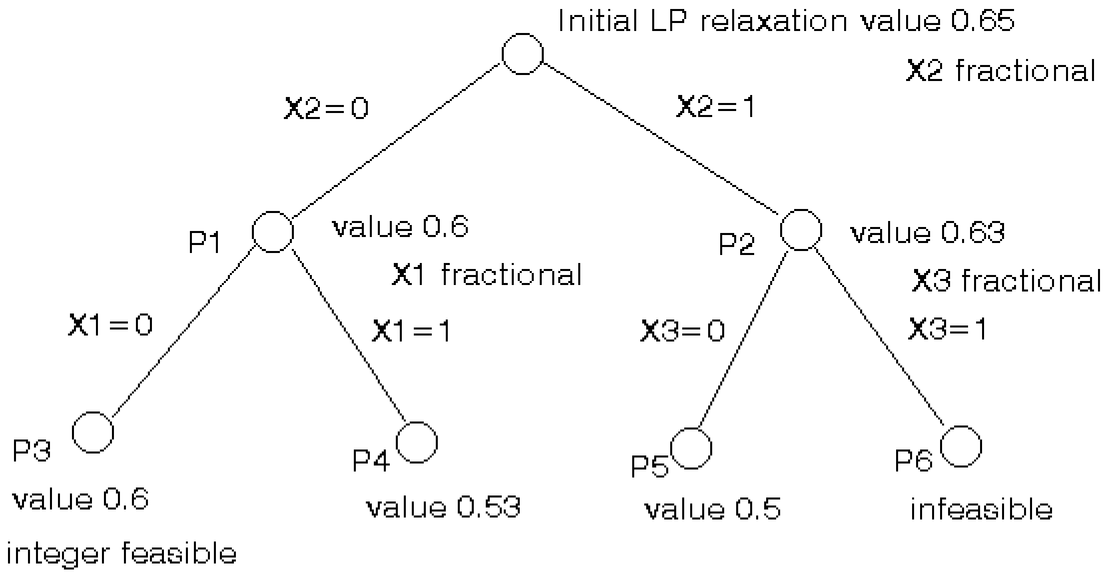
Branching on x_3 we get

P5 (P2 plus x_3=0) solution $x_1=x_2=1$, $x_3=x_4=0$ of value 0.5

P6 (P2 plus x_3=1) solution infeasible

Neither of P5 or P6 lead to further branching so we are done, we have discovered the optimal integer solution of value 0.6 corresponding to $x_3=x_4=1$, $x_1=x_2=0$.

The entire process we have gone through to discover this optimal solution (and to prove that it is optimal) is shown graphically below.



Note here that this method, like complete enumeration, also involves powers of two as we progress down the (binary) tree. However also note that we did not enumerate all possible integer solutions (of which there are 16). Instead here we solved 7 LP's. This is an important point, and indeed why tree search works at all. We do not need to examine as many LP's as there are possible solutions. While the computational efficiency of tree search differs for different problems it is this basic fact that enables us to solve problems that would be completely beyond us were we to try complete enumeration.

5.2.3.2 An Example of Branch and Bound in Graphical Solution (Winston 9.3., p. 503)

[LP Relaxation and the first two subproblems](http://www.isl.itu.edu.tr/ya/branch_and_bound_graphical_f1.jpg)

(http://www.isl.itu.edu.tr/ya/branch_and_bound_graphical_f1.jpg)

[Rest of the solution](http://www.isl.itu.edu.tr/ya/branch_and_bound_graphical_f2.jpg)

(http://www.isl.itu.edu.tr/ya/branch_and_bound_graphical_f2.jpg)

LINDO OUTPUT:

```
     MAX      8 X1 + 5 X2
     SUBJECT TO
         2)    X1 + X2 <=   6
         3)    9 X1 + 5 X2 <=   45
     END
     GIN       2

 LP OPTIMUM FOUND AT STEP 2
 OBJECTIVE VALUE = 41.2500000

 SET X1 TO <= 3 AT 1, BND= 39.00 TWIN= 41.00   9

 NEW INTEGER SOLUTION OF 39.0000000 AT BRANCH 1 PIVOT 9
 BOUND ON OPTIMUM:  41.00000
 FLIP X1 TO >= 4 AT 1 WITH BND= 41.000000
 SET X2 TO <= 1 AT 2, BND= 40.56 TWIN=-0.1000E+31   12
 SET X1 TO >= 5 AT 3, BND= 40.00 TWIN= 37.00   15

 NEW INTEGER SOLUTION OF 40.0000000 AT BRANCH 3 PIVOT 15
 BOUND ON OPTIMUM:  40.00000
 DELETE        X1 AT LEVEL     3
 DELETE        X2 AT LEVEL     2
 DELETE        X1 AT LEVEL     1
 ENUMERATION COMPLETE. BRANCHES= 3 PIVOTS= 15

 LAST INTEGER SOLUTION IS THE BEST FOUND
 RE-INSTALLING BEST SOLUTION...

        OBJECTIVE FUNCTION VALUE

        1)      40.00000

  VARIABLE         VALUE          REDUCED COST
        X1          5.000000          -8.000000
        X2          0.000000          -5.000000

       ROW   SLACK OR SURPLUS      DUAL PRICES
        2)          1.000000          0.000000
        3)          0.000000          0.000000

 NO. ITERATIONS=       15
 BRANCHES=     3 DETERM.=  1.000E    0
```

5.2.4 The Cutting Plane Algorithm

There is an alternative to branch and bound called ***cutting planes*** which can also be used to solve integer programs.

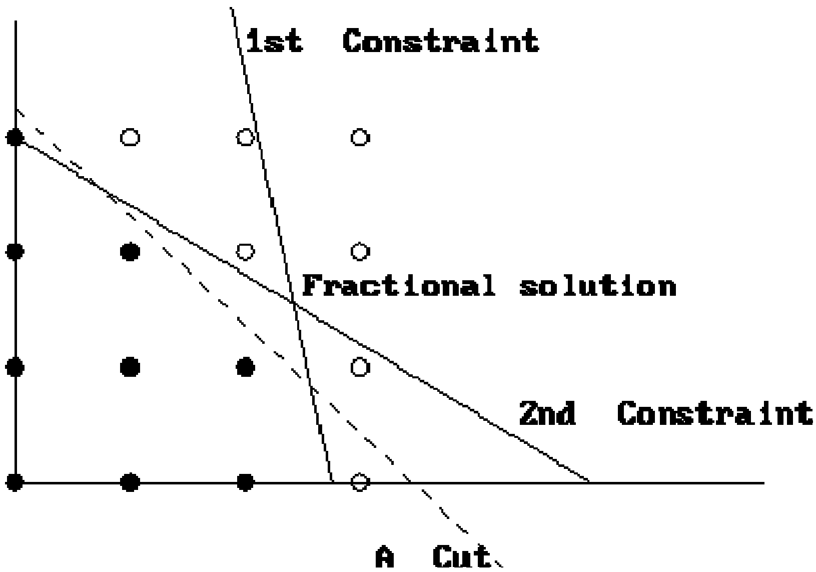
The fundamental idea behind cutting planes is to add constraints to a linear program until the optimal basic feasible solution takes on integer values. Of course, we have to be careful which constraints we add: we would not want to change the problem by adding the constraints. We will add a special type of constraint called a ***cut***.

A cut relative to a current fractional solution satisfies the following criteria:

Every feasible integer solution is feasible for the cut, and

The current fractional solution is not feasible for the cut.

This is illustrated in the figure given below.



There are two ways to generate cuts:

Gomory cuts generates cuts from any linear programming tableau. This has the advantage of “solving" any problem but has the disadvantage that the method can be very slow.

The second approach is to use the structure of the problem to generate very good cuts. The approach needs a problem-by-problem analysis, but can provide very efficient solution techniques.

5.2.4.1 Steps

1. Find the optimal tableau for the IP's LP relaxation. If all variables in the optimal solution assume integer values, we have found an optimal solution! Otherwise proceed to next step
2. Pick a constraint in the optimal tableau whose RHS has the fractional part closest to ½.
3. For the constraint identified, put all of the integer parts on the left side (round down), and all the fractional parts on the right
4. Generate the cut as: "RHS of the modified constraint" $\leq$ 0
5. Use the dual simplex to find the optimal solution to the LP relaxation, with the cut as an additional constraint. If all variables in the optimal solution assume integer values, we have found an optimal solution. Otherwise return to Step 2.

5.2.4.2 Dual Simplex Method (for a Max Problem)

We choose the most negative RHS.

BV of this pivot row leaves the basis.

For the variables that have a negative coefficient in the pivot row, we calculate the ratios (coefficient in R0 / coefficient in pivot row).

Variable with the smallest ratio (absolute value) enters basis.

5.2.4.3 Example

Consider the following integer program:

$$\begin{aligned}
&\max z = 8x1 + 5x2 \\
&\text{st} \quad x1 + x2 \le 6 \\
&\qquad 9x1 + 5x2 \le 45 \\
&x1, x2 \ge 0 \text{ and integer}
\end{aligned}$$

If we ignore integrality, we get the following optimal tableau:

z	x1	x2	s1	s2	RHS
1	0	0	1.25	0.75	41.25
0	0	1	2.25	- 0.25	2.25
0	1	0	- 1.25	0.25	3.75

Let's choose the constraint whose RHS has the fractional part closest to ½::

$x1 - 1.25\, s1 + 0.25\, s2 = 3.75$ (Arbitrarily choose the second constraint)

We can manipulate this to put all of the integer parts on the left side (round down), and all the fractional parts on the right to get:

$x1 - 2\, s1 + 0\, s2 - 3 = 0.75 - 0.75\, s1 - 0.25\, s2$

Now, note that the left hand side consists only of integers, so the right hand side must add up to an integer. It consists of some positive fraction minus a series of positive values. Therefore, the right hand side cannot be a positive value. Therefore, we have derived the following constraint:

$0.75 - 0.75\, s1 - 0.25\, s2 \le 0$

This constraint is satisfied by every feasible integer solution to our original problem. But, in our current solution, s1 and s2 both equal 0, which is infeasible to the above constraint. This means the above constraint is a cut, called the *Gomory cut* after its discoverer.

We can now add this constraint to the linear program and be guaranteed to find a different solution, one that might be integer.

z	x1	x2	**s1**	s2	s3	RHS
1	0	0	**1.25**	0.75	0	41.25
0	0	1	2.25	- 0.25	0	2.25
0	1	0	- 1.25	0.25	0	3.75
0	0	0	- 0.75	- 0.25	1	**- 0.75**

The dual simplex ratio test indicates that s1 should enter the basis instead of s3. The optimal solution is an IP solution:

$z = 40,\ x1 = 5,\ x2 = 0$