SMM 641

Revenue Management and Pricing

Week 4: Network Revenue Management Part 2

Introduction to Linear Programming

Solving LPs with R

Sensitivity Analysis

Shadow (Bid) Prices

Linear programming (LP) for network management

Network management is, in essence, a problem of allocating scarce resources to several different uses, each of which yields a different return.

Linear programming is well suited for solving problems of this type.

Hence, one approach to network revenue management is to model it as a linear program.

Generic form of optimization problems

Decision Variables

Objective function

Constraints

$$\begin{aligned} &\textit{maximize (minimize)} \quad f(x) \\ &\textit{subject to} \quad g_1(x) \le \ (=, or \ge) \ A \\ &\qquad\qquad\quad g_2(x) \ge \ (=, or \ge) \ B \end{aligned}$$

Optimization problems have three components.

1. Decision variables (e.g. quantities for product i)	$x = (x_1, x_2, x_3, \ldots, x_n)$
2. Objective function (e.g. profit, cost)	$f(x) = f(x_1, x_2, \ldots, x_n)$
3. Constraints (e.g. restriction of available resource)	$g_1(x) = g_1(x_1, x_2, \ldots, x_n) \le A$ $g_2(x) = g_2(x_1, x_2, \ldots, x_n) \ge B$

Examples of constraints:

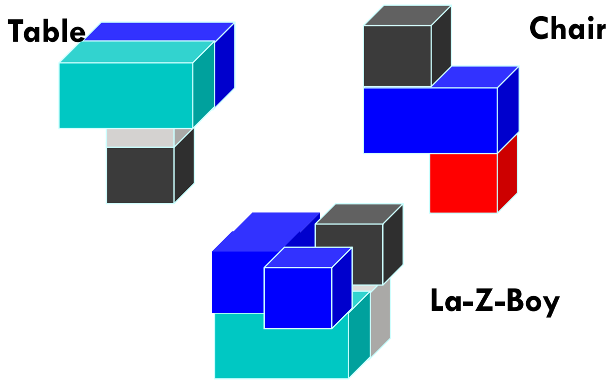
- Room allocations cannot exceed demand.
- Total room allocation cannot exceed hotel capacity.
- Total seat allocation cannot exceed plane capacity.

Example: Lego Chair, Table, and La-Z-Boy

Make tables, chairs, and La-Z-Boys to maximize profits.

Profit:

\$ 35 for each Table, \$18 for each Chair, \$48 for each La-Z-Boy



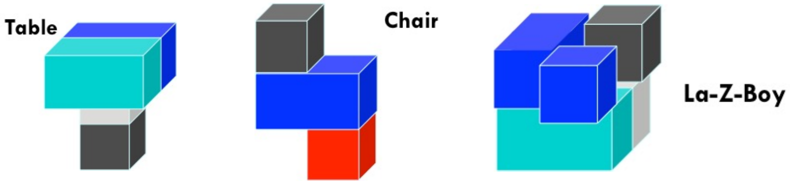
Example: Lego Chair, Table, and La-Z-Boy

Make tables, chairs, and La-Z-Boys to maximize profits.

Profit:
$ 35 for each Table, $18 for each Chair, $48 for each La-Z-Boy

Each table: uses 2 large blocks and 2 small blocks
Each chair: uses 1 large block and 2 small blocks
Each La-Z-Boy: uses 3 large blocks and 2 small blocks

You are limited by the availability of material. You only have 10 large blocks and 8 small blocks.
How many tables, chairs, and La-Z-Boys should you make to maximize profit?



Optimal Solution of Lego Game

of tables:

# of chairs:

# of La-Z-Boys:

Total profit:

Optimization

- Decision variables - What you change to optimize your objective
- Objective function – Maximize Profit
- Constraints - Resources are limited

Lego game:

Decision variables: Tables (t), Chairs (c), and La-z-boys (z)

Objective function: $\$35*t + \$18*c + \$48*z$

Constraints: 10 (large legos) $2*t + 1*c + 3*z \leq 10$

8 (small legos) $2*t + 2*c + 2*z \leq 8$

$t \geq 0, c \geq 0, z \geq 0$

Sensitivity Analysis

- Suppose you can buy an additional large block for $10. Should you buy it?
- Suppose you can buy an additional large block for $15. Should you buy it?
- How much would you be willing to pay for an additional large block?

Sensitivity Analysis

How much would you be willing to pay for an additional large block?

# of tables:

# of chairs:

# of La-Z-Boys:

Total profit:

What is Linear programming?

Linear programming: An optimization problem whose objective function and constraints are linear.

1. So, what is a **linear relationship**?
2. The function is a **sum of terms**
3. Each **term** of the function has **at most one decision variable** (multiplied by a constant).
4. Each variable is raised to the **first power**

Examples:	Linear?
$f(x_1, x_2) = 100x_1 + 200x_2$	Yes
$g(x_1, x_2, x_3) = 4x_1 + 5x_2 - 7x_3 \leq 400$	Yes
$f(p) = (p-2)(20-0.5p)$	No, because it expands to $-0.5p^2 + 21p - 40$

What is Linear programming?

	Linear Function?
(1) $f(x_1, x_2, x_3) = 4x_1 - 7x_2 + 10x_3 + 7$	Yes
(2) $-3x_2 + 14.5x_1 \leq 30$	Yes
(3) $f(x_1, x_2, x_3) = x_1 + 10x_2 - 2.5x_2^3$	No, because of x_2^3
(4) $7x_1x_2 + x_2 \leq 100$	No, because of x_1x_2
(5) $f(x_2) = 1/x_2$	No, because of $1/x_2$

Implications of linear relationships

- Constant contribution of every decision variable.
- The contribution of each decision variable is additive

Because of these reasons, computers can solve big problems fast if they are linear programs.

Linear programming: An introductory example

- The Savoury Sauce (S2) Company owns a small shop that **produces** two popular types of salsa sauce, the **Red Ripe** and the **Sassy Spicy**. Two basic ingredients, **tomatoes** and **pepper** are used to produce the sauces.
- The maximum availability of tomatoes is 8 kgs a day; that of pepper is 6 kgs a day. The requirements of ingredients per kg of the Red Ripe and the Sassy Spicy are summarized in the following table.

	Requirements (kg/kg)		
	Red Ripe	Sassy Spicy	Maximum availability (kg)
Peppers	1	2	6
Tomatoes	2	1	8

- In order to have a diverse product line, Sassy Spicy production cannot exceed Red Ripe production by more than 1 kg.
- A market survey also showed that only up to 2 kgs of Sassy Spicy can be sold daily.
- The profit margin is £ 3 for Red Ripe per kg and £ 2 for Sassy Spicy per kg.
- What should the Savoury Sauce Company produce?

Linear programming: An introductory example

Steps in Formulation:

1. Understanding the Problem.
   - How many kgs each of the Red Ripe and the Sassy Spicy sauce should the company produce in order to maximize their profit, while using no more than 8 kgs of tomatoes and 6 kgs of pepper?
2. Identify the Decision Variables.
   - How many kgs of Red Ripe and of Sassy Spicy should be produced?

     X_1: Amount of Red Ripe sauce to produce

     X_2: Amount of Sassy Spicy sauce to produce
3. State the objective function as a linear combination of the decision variables
   - The objective of maximizing the profit the company earns is stated mathematically as:

     Max $3\ X_1 + 2\ X_2$

Linear programming: An introductory example

Steps in Formulation:

4. State the constraints as linear combinations of decision variables.

   In our example, what are the four major constraints the Company faces?

   - There are only 6 kgs of peppers available. Each kg of Red Ripe requires 1 kg of peppers and each kg of Sassy Spicy requires 2 kg of peppers. The total amount of peppers used must not exceed the total availability:

     $1\ X_1 + 2\ X_2 \leq 6$
   - There are only 8 kgs of tomatoes available. Each kg of Red Ripe requires 2 kg of tomatoes and each kg of Sassy Spicy requires 1 kg of tomatoes. The total amount of tomatoes used must not exceed the total availability:

     $2\ X_1 + 1\ X_2 \leq 8$

Linear programming: An introductory example

Steps in Formulation:

4. State the constraints as linear combinations of decision variables.

   In our example, what are the four major constraints the Company faces?

   - There is a production diversity requirement. The production of Sassy Spicy cannot exceed the production of Red Ripe by more than 1 kg:

     $-1\ X_1 + 1\ X_2 \leq 1$ (i.e., $X_2 - X_1 \leq 1$)
   - There is a demand limit. A market survey showed that only up to 2 kgs of Sassy Spicy can be sold daily so the production of Sassy Spicy should not exceed 2kgs:

     $X_2 \leq 2$
5. Identify whether any decision variables should be nonnegative.
   - In our example, it is impossible to produce a negative amount of sauce.

     $X_1 \geq 0$ and $X_2 \geq 0$

Linear programming: An introductory example

Decision Variables: What do we want to determine/control/set values?

X_1: Amount of Red Ripe to produce, X_2: Amount of Sassy Spicy to produce.

Objective function:

- Profit of each product is unit profit contribution * amount produced

  Maximize $3X_1 + 2X_2$

Constraints:

1. Can not use more than 6 kgs of peppers.

   $X_1 + 2\ X_2 \leq 6$
2. Can not use more than 8 kgs of tomatoes.

   $2\ X_1 + X_2 \leq 8$
3. Sassy Spicy production cannot exceed Red Ripe production by more than 1kg

   $X_2 \leq X_1 + 1$, or equivalently, $-X_1 + X_2 \leq 1$
4. The maximum amount of Sassy Spicy the company can sell is 2 kgs per day.

   $X_2 \leq 2$
5. Can decision variables be negative?

   $X_1 \geq 0$, $X_2 \geq 0$

Savoury Sauce Company

Decision variables

Define:

$X_1 =$ Amount of Red Ripe to be produced

$X_2 =$ Amount of Sassy Spicy to be produced

Objective function

Maximize $z = 3X_1 + 2X_2$

Subject to:

Constraints

$X_1 + 2X_2 \leq 6$	(1)	(Availability of Peppers)
$2X_1 + X_2 \leq 8$	(2)	(Availability of Tomatoes)
$-X_1 + X_2 \leq 1$	(3)	(Diversity restriction in product line)
$X_2 \leq 2$	(4)	(Demand Restriction)
$X_1 \geq 0$		(Non-negativity)
$X_2 \geq 0$		(Non-negativity)

Visualizing an LP

Visualizing a two variable linear program:

1. Finding the (range of) values of the decision variables for which each constraint is met.
2. Determining the feasible region of the solution space, the set of decision variables satisfying all constraints.
3. Determining the optimal solution.

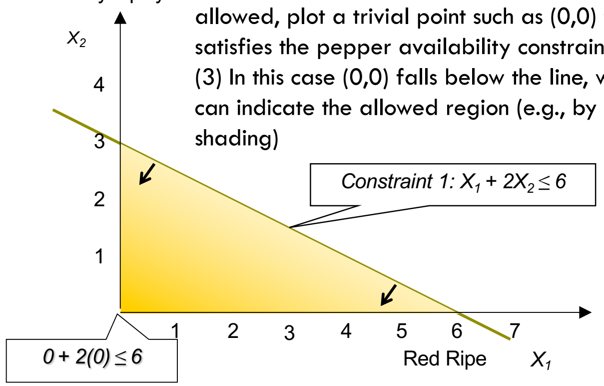
Step 1. Find the values of the decision variables for which each constraint is met

Plot the decision variables satisfying each constraint.

Let's start with the Pepper availability constraint (order not important)

$X_1 + 2X_2 \leq 6$

(1) First draw the line $X_1 + 2X_2 = 6$.
(2) To identify which side of the line is allowed, plot a trivial point such as (0,0) that satisfies the pepper availability constraint.
(3) In this case (0,0) falls below the line, we can indicate the allowed region (e.g., by shading)

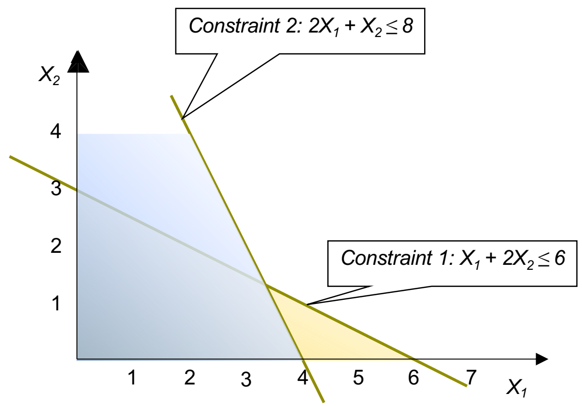


Step 1. Find the values of the decision variables for which each constraint is met

Now, add the next constraint, e.g., the Tomato availability constraint.

$X_1 + 2X_2 \leq 6$

$2X_1 + X_2 \leq 8$



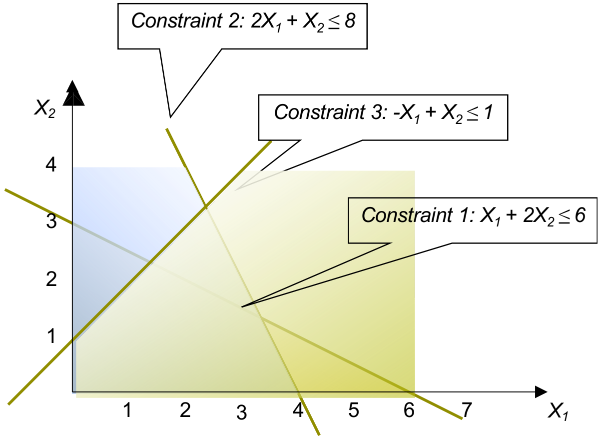
Step 1. Find the values of the decision variables for which each constraint is met

Continue with the next constraint, e.g., production diversity.

$X_1 + 2X_2 \le 6$

$2X_1 + X_2 \le 8$

$-X_1 + X_2 \le 1$



Step 1. Find the values of the decision variables for which each constraint is met

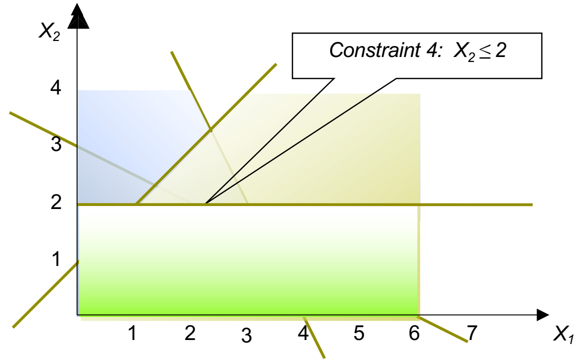
Continue with the next constraint, e.g., demand restriction.

$X_1 + 2X_2 \le 6$

$2X_1 + X_2 \le 8$

$-X_1 + X_2 \le 1$

$X_2 \le 2$



Step 1. Find the values of the decision variables for which each constraint is met

And the non-negativity constraints.

$X_1 + 2X_2 \le 6$

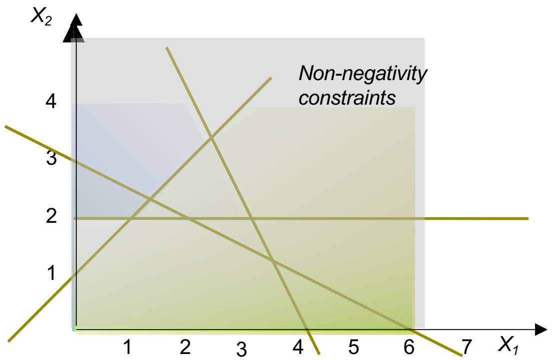
$2X_1 + X_2 \le 8$

$-X_1 + X_2 \le 1$

$X_2 \le 2$

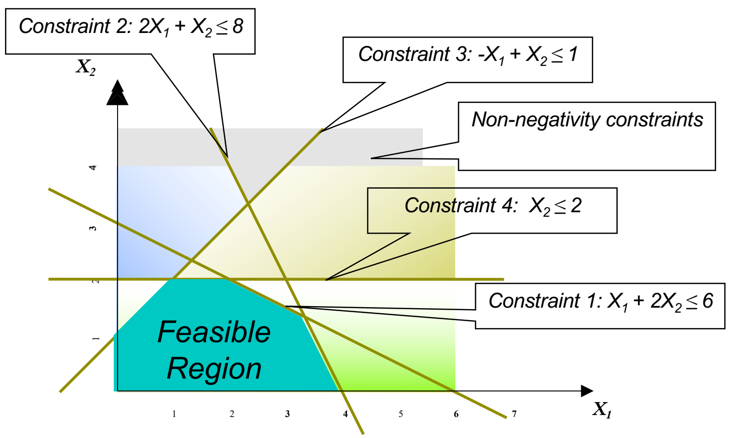
$X_1 \ge 0$

$X_2 \ge 0$



Step 2. Determine the feasible region

Feasible region is the set of decision variables satisfying all constraints.



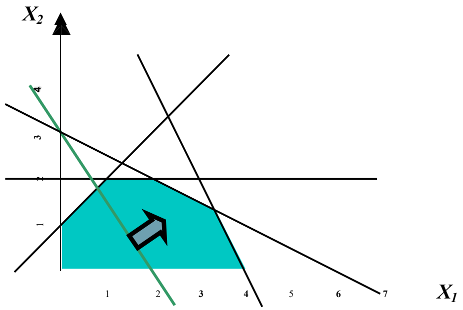
Step 3. Introduce the objective function

Write the objective function as $z = 3X_1 + 2X_2$

Start by picking a value for z that would allow you to draw the line. For example, When $z = 6$, the intercepts will be (2,0) and (0,3)

Connecting those intercepts gives us the line of equal profits of 6. Any allocation between (2,0) and (0,3) has the same profit of 6.

What happens as z increases to 9?



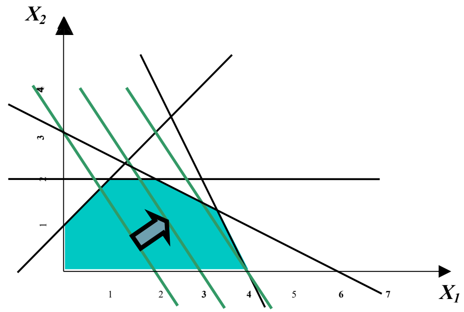
Step 3. Introduce the objective function

Write the objective function as $z= 3X_1 + 2X_2$

When $z = 9$, the intercepts will be (3,0) and (0,4.5)

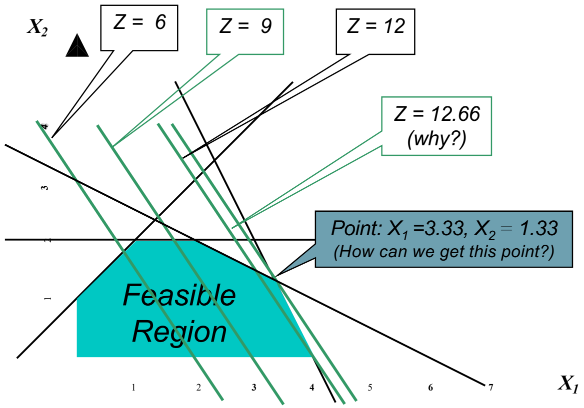
Connecting those intercepts gives us the line of equal profits of 9. Any allocation between (3,0) and (0,4.5) has the same profit of 9.

What happens as z increases further?



Step 4. Improve the objective function

Increase "Z" value as far as possible while still intersecting with the feasible region.



*An **optimal solution** is a feasible solution that has the most favorable value of the objective function.*

An optimal solution is a corner point feasible solution (why?)

Simultaneously solve:

$X_1 + 2 X_2 = 6$

$2 X_1 + X_2 = 8$

$(X_1 , X_2) =(3.333,1.333)$

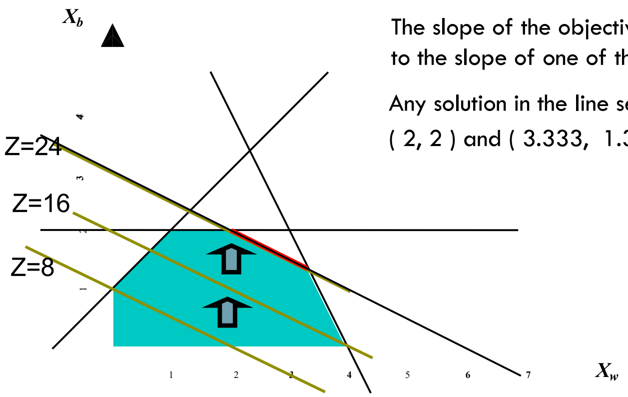
Z=3*3.333+2*1.333
= 12.66

An Important Property of Linear Programmes

- If there is exactly one optimal solution, it must be a corner point feasible solution.
- If there are two corner point optimal solutions, every point in between is also optimal.

The optimal solution may be non-unique: Multiple optimal solutions

Max $z = 4X_1 + 8X_2$



The slope of the objective function is parallel to the slope of one of the constraints.

Any solution in the line segment between (2, 2) and (3.333, 1.333) achieves z=24.

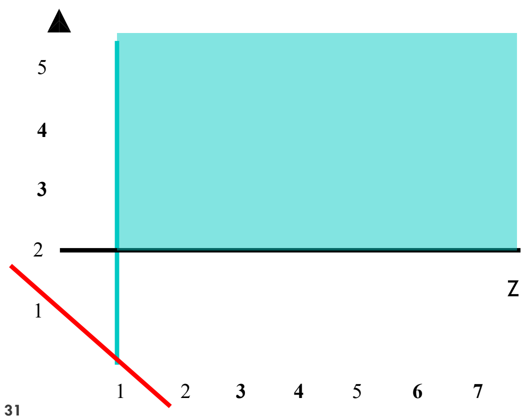
A few notes on Linear Programming

Linear Programming takes places in an ideal world where:

- (Proportionality) Contributions (consumptions) are proportional.
- (Additivity) Total contribution is the sum of individual contributions
- (Divisibility) Activities (decision variables) are divisible into fractions/small pieces. (e.g. 1.3245 chairs)
- (Certainty) All values (other than decision variables) are known.

A poor formulation may leave your solution unbounded

If the objective function can increase indefinitely, the problem is unbounded.

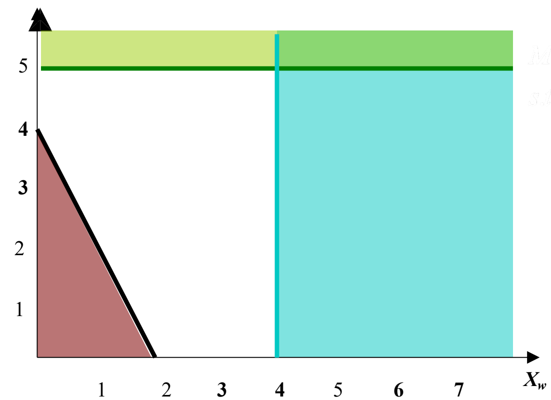


$$\begin{aligned} Max\ \ z &= X_w + X_b \\ s.t.\quad X_w &\geq 1 \\ X_b &\geq 2 \end{aligned}$$

Z keeps going and going….

Sometimes, a poor formulation makes the problem infeasible

An infeasible problem has no feasible region.
Plot the feasible region.



$$\begin{aligned} Max\ \ z &= 3X_w + 2X_b \\ s.t.\quad 4X_w + 2X_b &\leq 8 \\ X_w &\geq 4 \\ X_b &\geq 5 \end{aligned}$$

Infeasible or Unbounded LPs: Usually modeling errors

Modeling Errors

- Model could be incorrect – check constraints, max vs. min etc.
- Model could be incomplete – have all constraints been accounted for?
- Data could be incorrect – are all parameters reasonable?

Infeasibility: We could be trying to solve an impossible problem.

Unbounded: Unlikely to occur in practice.

Savoury Sauce Company

Define:

X_1 = Amount of Red Ripe to be produced

X_2 = Amount of Sassy Spicy to be produced

Decision variables

Objective function

Maximize $z = 3X_1 + 2X_2$

Subject to:

Constraints

$X_1 + 2X_2 \leq 6$	(1)	(Availability of Peppers)
$2X_1 + X_2 \leq 8$	(2)	(Availability of Tomatoes)
$-X_1 + X_2 \leq 1$	(3)	(Diversity restriction in product line)
$X_2 \leq 2$	(4)	(Demand Restriction)
$X_1 \geq 0$		(Non-negativity)
$X_2 \geq 0$		(Non-negativity)

Solving an LP through R

- Please see R Supplement

```
install.packages("lpSolve",repos = "http://cran.us.r-project.org")
```

```
library(lpSolve)
```

Solving an LP through R

```
# Setting up the LP Formulation:

# This problem has two decision variables
# x1: Amount of Red Ripe to produce
# x2: Amount of Sassy Spicy to produce

# Objective Function Coefficients
# Profit contribution of Red Ripe is 3.
# Profit contribution of Sassy Spicy is 2.
obj.fun <- c(3,2);
```

Solving an LP through R

```r
# Constraint Coefficients
constr <- matrix(c(1,2,2,1,-1,1,0,1), ncol=2, byrow=TRUE)
# Alternatively, and especially for larger problems,
# you can build the constraint coefficient matrix in parts
# and combine them afterwards.  For example,
# Pepper usage: x1+2*x2
pepper<-c(1,2)
# Tomato usage: 2*x1+x2
tomato<-c(2,1)
# Diversity: -x1+x2
diversity<-c(-1,1)
# Demand: x2
demand<-c(0,1)
constr<-rbind(pepper,tomato,diversity,demand)

# Constraint directions: Equality and/or Inequality
# For each row of constraint, indicate the
# constraint direction, "<=", ">=", or "=".
constr.dir <- c("<=", "<=", "<=", "<=")

# Constraint Right Hand Side
# For each row of constraint, indicate the
# corresponding right hand side value.
# First two are pepper and tomato availability:
# x1+2*x2 <= 6 and 2*x1+x2 <= 8
# The next is for diversity (-x1+x2 <= 1)
# The last is for demand (x2 <= 2)
rhs <- c(6,8,1,2)
```

Solving an LP through R

```r
# Solving the LP: (type "max" or "min" for first argument)
prod.sol <- lp("max", obj.fun, constr, constr.dir, rhs, compute.sens=TRUE)
```

```r
# Optimal Solution (Values for Decision Variables)
prod.sol$solution
```

```
## [1] 3.333333 1.333333
```

```r
# Optimal Objective Function Value
prod.sol$objval
```

```
## [1] 12.66667
```

Ex: Convention

- Hanson Inn is a 96 room hotel located in Cityville.
- When a convention or a special event is in town, Hanson increases its normal room rates and takes reservations based on a revenue management system.
- The Classic Car Owners Association scheduled its annual convention in Cityville for the first weekend in April. Hanson Inn agreed to make at least 50% of its rooms available for convention attendees at a special convention rate in order to be listed as a recommended hotel for the convention. (Hint: it will allocate at least 48 rooms for Friday night and Saturday night each for convention customers.)

Ex: Convention

- Although the majority of attendees at the annual meeting typically request a Friday and Saturday two-night package, some attendees may select a Friday night only or a Saturday night only reservation.
- Customers not attending the convention may also request a Friday and Saturday two-night package, or make a Friday night only or Saturday night only reservation.
- Thus, six types of reservations are possible:
  - Convention customers two-night package; convention customers Friday night only; convention customers Saturday night only; regular customers two-night package; regular customers Friday night only; and regular customers Saturday night only.

Ex: Convention

- The room rates are as follows:

	Two-Night Package	Friday Night Only	Saturday Night Only
Convention	$225	$123	$130
Regular	$295	$146	$152

- The anticipated demands are as follows:

	Two-Night Package	Friday Night Only	Saturday Night Only
Convention	40	20	15
Regular	20	30	25

- Hanson Inn would like to determine how many rooms to make available for each type of reservation in order to maximize total revenue.

Ex: Convention

Let

CT = number of convention two-night rooms

CF = number of convention Friday only rooms

CS = number of convention Saturday only rooms

RT = number of regular two-night rooms

RF = number of regular Friday only rooms

RS = number of regular Saturday only room

Ex: Convention

$$
\begin{array}{llllllll}
\text{MAX} & 225\,CT+123\,CF+130\,CS+295\,RT+146\,RF+152\,RS & & & & & & \\
\text{ST} & CT & & & & & & \le 40 \\
 & & CF & & & & & \le 20 \\
 & & & CS & & & & \le 15 \\
 & & & & RT & & & \le 20 \\
 & & & & & RF & & \le 30 \\
 & & & & & & RS & \le 25 \\
 & CT + & CF & & & & & \ge 48 \\
 & CT + & & CS & & & & \ge 48 \\
 & CT + & CF + & & RT + & RF & & \le 96 \\
 & CT + & & CS + & RT + & & RS & \le 96 \\
 & \text{All variables non-negative.} & & & & & &
\end{array}
$$

Ex: Convention

- Please see R Supplement

The optimal solution is to allocate:

36 rooms to Convention/Two-night stay

12 rooms to Convention/Friday-night only stay

15 rooms to Convention/Saturday-night only stay

20 rooms to Regular/Two-night stay

28 rooms to Regular /Friday-night only stay

25 rooms to Regular /Saturday-night only stay

for a total anticipated revenue of $25,314.00.

Next: Sensitivity Analysis

- LP gives a solution while assuming the parameters (unit cost, unit profit) are known with certainty.
- Sensitivity analysis measures the effect of parameter changes on the optimal solution.
- Why is it important?
  - It tells you a range of parameters within which the optimal solution remains the same.
  - It tells you the price of resources, such as how much one additional kg of peppers is worth to Savoury Sauce Company.

Sensitivity Analysis

X_1: Amount of Red Ripe to produce

X_2: Amount of Sassy Spicy to produce

Max $3X_1+2X_2$

subject to

$1X_1+ 2X_2 \le 6$ (1)

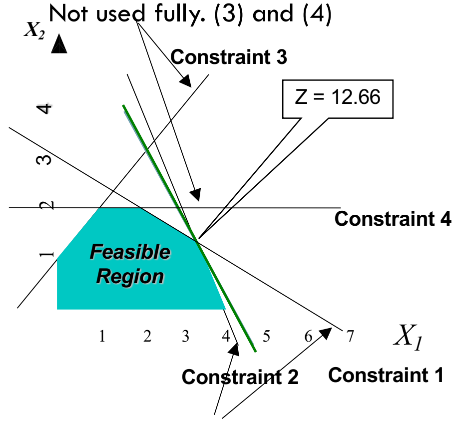
$2X_1+ 1X_2 \le 8$ (2)

$-X_1+ 1X_2 \le 1$ (3)

$X_2 \le 2$ (4)

$X_1 \ge 0, X_2 \ge 0$

Non-binding constraints: Not used fully. (3) and (4)



Binding constraints: Satisfied at equality. (Resource used fully) (1) and (2)

Case 1. Change in the objective function

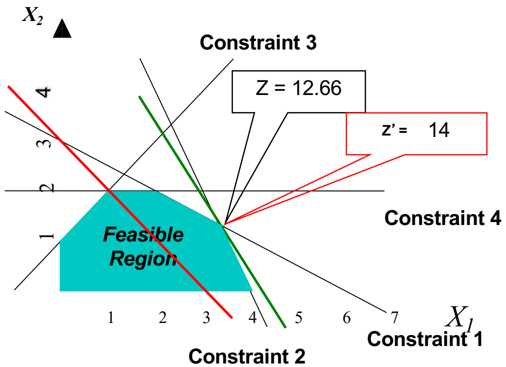
What happens if the profit of Sassy Spicy increases to 3?

Original objective function:
$Z = 3X_1+2X_2$

New objective function:
$Z' = 3X_1 + 3X_2$

Optimal solution remains the same: 3.33,1.33

Optimal Profit is now:
3*3.333+3*1.333=14



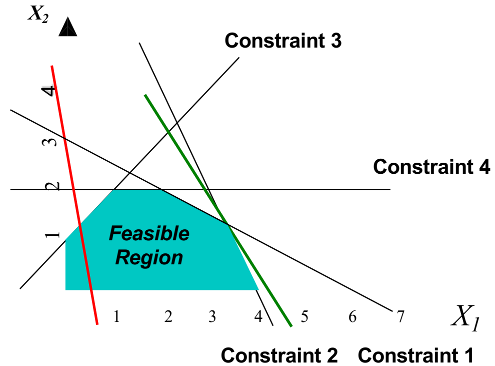
Case 1. Change in the objective function

What happens if the profit of Sassy Spicy decreases to 0.5?

New objective function:

$Z' = 3X_1 + 0.5X_2$

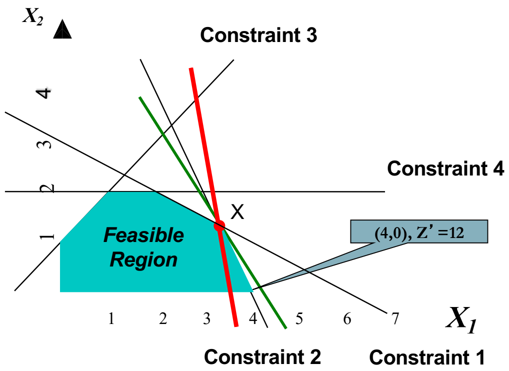
Let's increase Z value.



Case 1. Change in the objective function

Can we do better than point X?

What is the new optimal solution? $X_1 = 4$ and $X_2 = 0$. Profit = 12.



Can R Tell Us This?

- Range of parameters within which the optimal solution remains the same.

```
# Sensitivity Analysis
# Changes in Objective Function Coefficients
prod.sol$sens.coef.from
```

```
## [1] 1.0 1.5
```

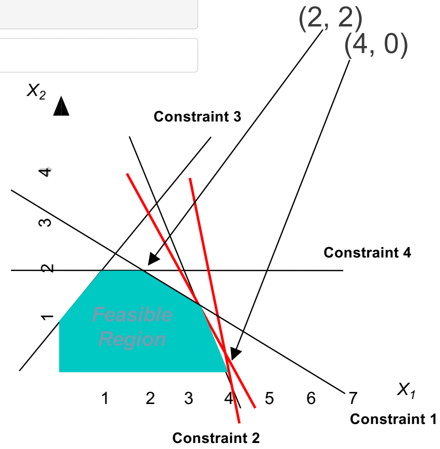
```
prod.sol$sens.coef.to
```

```
## [1] 4 6
```

When the profit margin, c_s on Sassy Spicy is within: [1,4], the optimal solution remains the same.

When $c_s > 6$, the optimal solution moves to: (2,2)

When $c_s < 1.5$, the optimal solution moves to: (4,0)

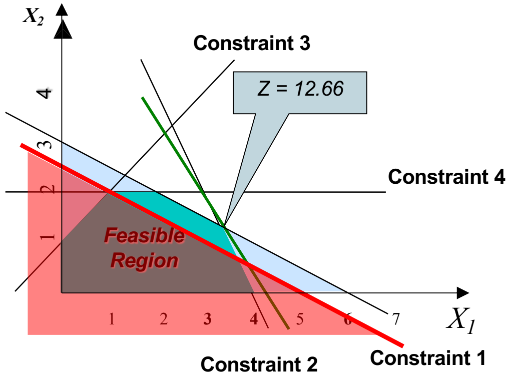


Case 2. Change in the constraint quantity

What happens if the company has 5 kgs of peppers instead of 6 kgs?

Original pepper availability constraint: $X_1 + 2X_2 \leq 6$

New pepper availability constraint: $X_1 + 2X_2 \leq 5$



Case 2. Change in the constraint quantity

What happens if the company has 5 kgs of peppers instead of 6 kgs?

Original pepper availability constraint: $X_1 + 2X_2 \leq 6$

New pepper availability constraint: $X_1 + 2X_2 \leq 5$

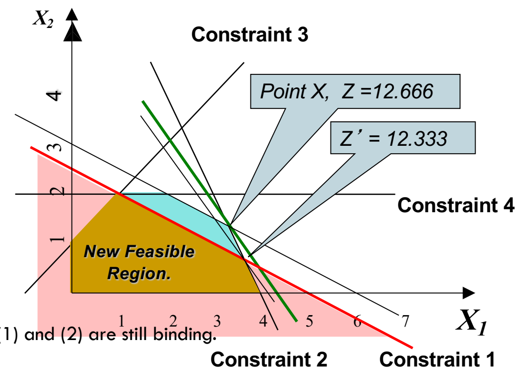
The feasible region:
(changes ~~, does not change)~~

The objective value:
(changes ~~, does not change)~~

The optimal solution
(changes ~~, do not change)~~

The binding constraints
~~(change~~, do not change)

The previous two binding constraints (1) and (2) are still binding.



Case 2. Change in the constraint quantity

What happens if the company has 7 kgs of peppers instead of 6 kgs?

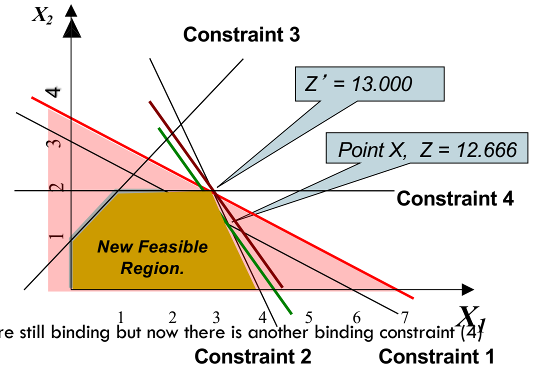
New pepper availability constraint: $X_1 + 2X_2 \leq 7$

The feasible region:
(changes ~~, does not change~~)

The objective value:
(changes ~~, does not change~~)

The optimal decision variable
(changes ~~, does not change~~)

The binding constraints
(changes ~~, does not change~~)



The two binding constraints (1) & (2) are still binding but now there is another binding constraint (4)

Case 2. Change in the constraint quantity

If the company has more than 7 kgs of peppers, does it increase profit?

If the R.H.S of the pepper availability constraint is greater than 7, it is no longer binding

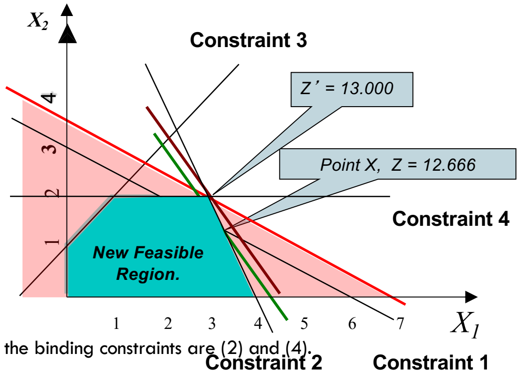
Compared with the case of 7 kgs

The feasible region:
(~~changes~~ , does not change)

The objective value:
(~~changes~~, does not change)

The optimal decision variable
(~~changes~~, does not change)

The binding constraints
(changes ~~, does not change~~)



Now constraint (1) is no longer binding, the binding constraints are (2) and (4).

Shadow Prices and Obtaining Shadow Prices in R

Shadow price: change in the objective function for a unit increase of the right-hand side of the constraint (e.g., the amount of resource available) while everything else remains the same.

Only **scarce resources are valuable**.

- *If the constraint is not binding, the shadow price is zero.*
- The shadow price is non-zero *only when a constraint is binding*

Obtaining Shadow Prices in R:

```
# Shadow (Dual) Value of Constraints
prod.sol$duals[1:length(constr.dir)]
```

```
## [1] 0.3333333 1.3333333 0.0000000 0.0000000
```

Ex. The shadow price of the pepper constraint is 0.333333

Profit: 12.667 with 6 kgs of peppers
Profit: 13.0 with 7 kgs of peppers
Profit: 12.333 with 5 kgs of peppers

The shadow price of the maximum demand constraint is 0.

Profit: 12.667 with max demand of 2 kgs.
Profit: 12.667 with max demand of 3 kgs.

Case 2. Change in the constraint quantity

- Range over which Shadow Price remains valid

```
prod.sol$duals.from[1:length(constr.dir)]
```

```
## [1]  4e+00  6e+00 -1e+30 -1e+30
```

```
prod.sol$duals.to[1:length(constr.dir)]
```

```
## [1] 7.0e+00 1.2e+01 1.0e+30 1.0e+30
```

- For example, for the peppers constraint, if the available is within 4 and 7,
  - the profit increases by 0.333 per unit increases of pepper availability.
- What happens if the change is beyond the range?
  - Need to re-solve the problem to find specifics.