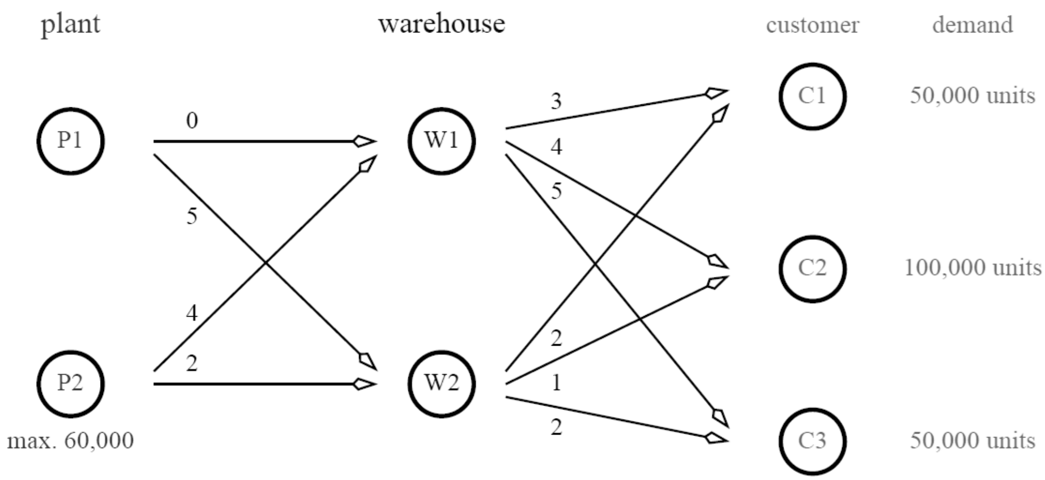
9. Linear Programming: Perspectives

Linear programs are by no means an "introductory" method that sets the table for more complicated scenarios. Although complicated optimization problems *do* exist and can be very applicable, the fact is that linear programming is the **most frequently** used method in the realm of optimization. Linear programs can be very large and impossible to handle by hand, but have huge value in their solutions. As a matter of fact, the first step toward solving a nonlinear program is to attempt to **reformulate** it as a linear program (with appropriate relaxations and assumptions!). Moreover, the first stage in the solution of truly nonlinear programs is to solve a **linear relaxation** as a starting point… So really, linear programs are absolutely everywhere!
For example, consider the following figure, which shows a simple distribution network of two manufacturing plants supplying two warehouses, with each warehouse attempting to fulfill three customer demands (all for the same product; Figure courtesy of Dr. Chachuat). The manufacturing costs for each plant are the same, and the numbers indicated on the arrows represent the shipping costs of going from one location to the other. Our goal is to choose how much of each plant to produce to satisfy demand with the lowest possible shipping costs. Not so easy, is it? Well, it is with Linear Programming!



9.1. Learning Goals

In this section, we have a few learning goals to lay out, each of which are *just* as important as the others. We are going to tackle these learning goals by attempting to understand the **geometric** interpretation and necessary **matrix calculations** to form a comprehensive knowledge base.

- Attitudes
  - An *optimum* is better than an *answer*.
  - Numbers with no perspective or understanding are useless ("It says $x_3 = 50$").
- Skills
  - Formulation of real-world problems into mathematical systems of equations.
  - Solving optimization problems.
  - Communicating the results to other engineers and those without technical backgrounds.
- Knowledge
  - When and how we can formulate optimization models.

- Know when we are dealing with "weird events" that can occur in optimization (analysis).

9.2. Definitions of Linear Programming

9.2.1. General Formulation of a Linear Program

Linear Program

An optimization model is known as a **Linear Program** if it has the following features:

1. It is comprised of *only continuous variables*
2. It has a *single linear objective function*
3. It has *only linear equality and/or inequality constraints*

A linear program has the following general formulation notation:

$$\min_{x} \phi = \boldsymbol{c}^T \boldsymbol{x} \qquad \leftarrow \text{Objective Function}$$

$$s.t. \qquad \leftarrow \text{“Subject to”}$$

$$A_h \boldsymbol{x} = \boldsymbol{b}_h \qquad \leftarrow \text{Equality Constraints}$$

$$A_g \boldsymbol{x} \leq \boldsymbol{b}_g \qquad \leftarrow \text{Inequality Constraints}$$

$$\boldsymbol{x}_{lb} \leq \boldsymbol{x} \leq \boldsymbol{x}_{ub} \qquad \leftarrow \text{Variable Bounds}$$

You should instantly notice that this is different from our "general" formulation that we came up with in the Problem Formulation Lecture. In this case, the objective function $f(\boldsymbol{x})$ has been replaced by $\boldsymbol{c}^T\boldsymbol{x}$ because we are **specifying** that the objective function is in fact a linear combination of the decision variables $\boldsymbol{x}$, or (ALL of these are equivalent… Think why!):

$$\phi = \boldsymbol{c}^T\boldsymbol{x} = \underbrace{[c_1 \;\; c_2 \; \dots \; c_n]\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{One\ notation} = \underbrace{c_1x_1 + c_2x_2 + \cdots + c_nx_n}_{Another\ notation} = \underbrace{\sum_{i=1}^{n} x_i c_i}_{ANOTHER\dots}$$

For the same reason, we have specified the structure of the constraints as $A_h\boldsymbol{x} = \boldsymbol{b}_h$ and $A_g\boldsymbol{x} \leq \boldsymbol{b}_g$ because we no longer need to constrain ourselves to the general forms $\boldsymbol{h}(\boldsymbol{x})$ and $\boldsymbol{g}(\boldsymbol{x})$. Our constraints are linear combinations as well! In summary:

- $x_j \rightarrow j^{th}$ decision variable.
- $c_j \rightarrow j^{th}$ cost coefficient for the j^{th} decision variable.
- $a_{i,j} \rightarrow$ constraint coefficient (think of systems of equations) for variable *j* in constraint *i*.
- $b_i \rightarrow$ RHS coefficient for constraint *i* (we no longer need RHS = 0 since we are linear!).
- A_h and A_g both have *n* columns (number of variables) and m_h and m_g rows, respectively (number of constraints of each type)

9.2.2. Interior, Boundary, and Extreme Points

Point Classifications in a Linear Program

A **feasible solution** (NOT necessarily an optimum) to a linear program is classified as:

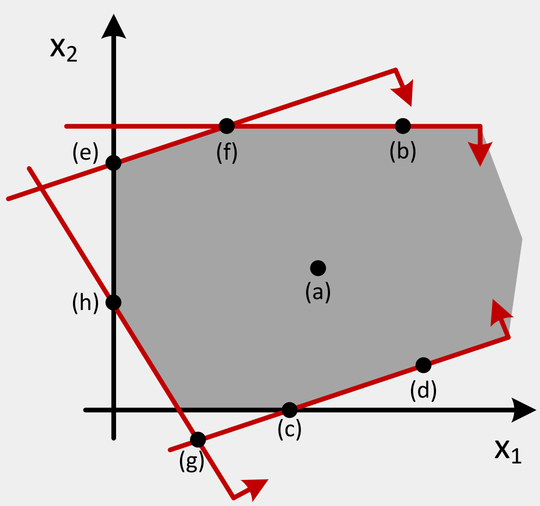
- A **boundary point** if at least one inequality constraint is satisfied as an *equality* (is active) at the given point.
- An **interior point** if no inequalities are "active".
- An **extreme point** (or a **corner point**) if every line segment in the feasible region containing it *also has that point as an endpoint*

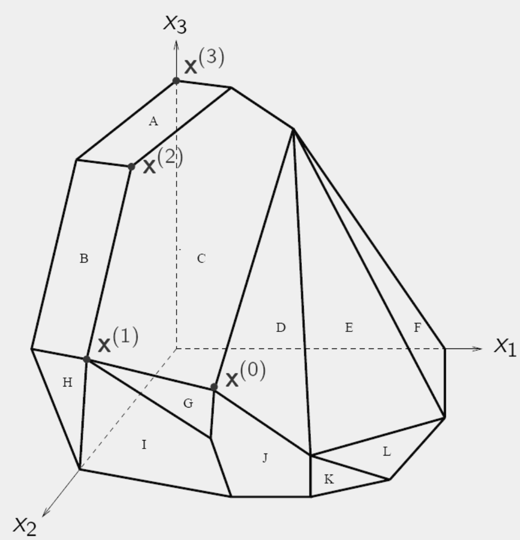
When considering extreme points, remember the following **important remarks**:

- Extreme points are called so because they are "jagged" extrema and thus "stick out" of the feasible region (they are pointy/cornered due to the convexity of linear constraints).
- Every extreme point is determined by the intersection of **at least n linearly independent constraints** that are active ONLY at that solution. For example, in three dimensions ($n = 3$), the intersection of three (or possibly more!) constraints define each extreme point.

Example - Classifying Feasible/Extreme Points

1. Determine if each of the points in the figure below to the left represent interior, boundary, and/or extreme points for the LP feasible region.
2. Identify all sets of 3 constraints that determine the extreme points in the figure below to the right.





Solution – Classifying Feasible/Extreme Points

(a) Interior	(e) Extreme	$\boldsymbol{x}^{(0)}$ Any 3 of C, D, G, J
(b) Boundary	(f) Extreme	$\boldsymbol{x}^{(1)}$ Any 3 of B, C, G, H, I
(c) Extreme	(g) Infeasible	$\boldsymbol{x}^{(2)}$ {A, B, C}
(d) Boundary	(h) Extreme	$\boldsymbol{x}^{(3)}$ {A, $x_1 \geq 0$, $x_2 \geq 0$}

The classification of boundary points versus interior points might seem rather arbitrary at first, but it actually leads us to a **powerful and useful set of results**.

Optimal Points in Linear Programs

Fundamental Result 1

EVERY solution to an LP with a non-constant objective function will be at a *boundary point* of its feasible region.

Fundamental Result 2

If an LP has a unique optimum, it *MUST* occur at an *EXTREME POINT* of the feasible region.

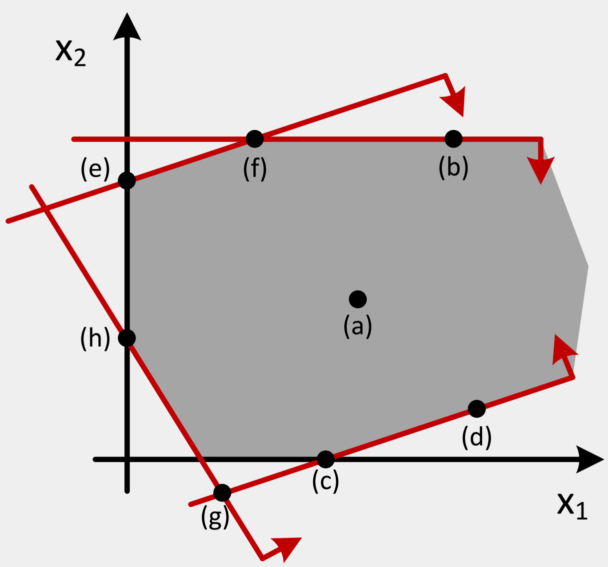
Moreover, if an LP has multiple (equivalent) optima, *one of them MUST* occur at an extreme point of the feasible region.

Fundamental Result 3

EVERY local optimum for an LP is also a global optimum because *ALL LPs are CONVEX*

Example – Identifying Optimal Points

For the following linear program, identify which of the labeled points are potential optima, potential unique optima, or neither:



Example Solution – Classifying Feasible/Extreme Points

(a) neither
(b) potential optima
(c) potential unique optima
(d) potential optima
(e) potential unique optima
(f) potential unique optima
(g) neither (infeasible)
(h) potential unique optima

The final piece of the puzzle that we require before being able to solve these types of problems are the definitions of **edges** and **adjacent points**. Once we get this out of the way, we will be able to derive a solution method that exploits the geometric nature of all linear programs. So… Why wait! They are defined right here:

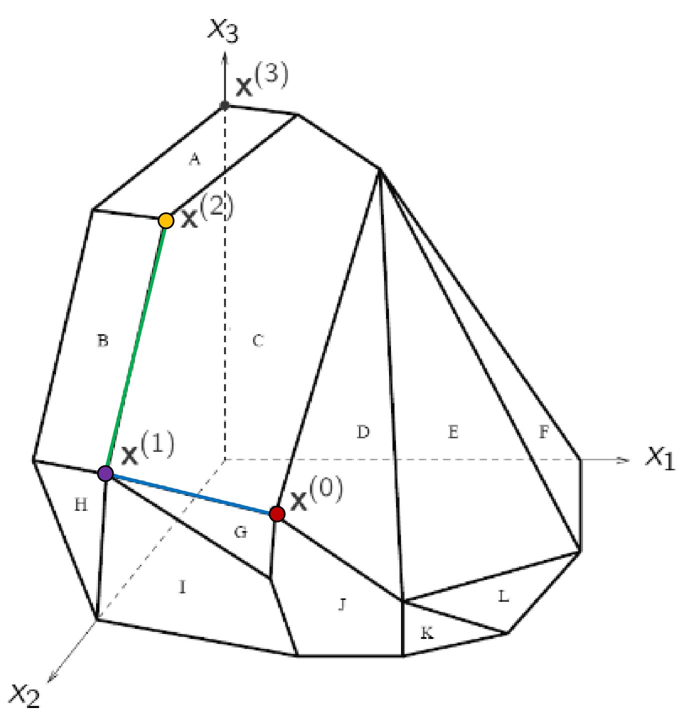
Adjacent Extreme Points and Edges

Two extreme points of an LP feasible region are **adjacent** if they are determined by an active set of constraints that differ by only one element (variable). An **edge** of the feasible region is a 1-D set of feasible points along a line determined by the collection of active constraints.

- Every edge is determined as the intersection of **exactly $\boldsymbol{n-1}$ independent, active constraints**.
- Adjacent extreme points are **joined** by the edge that has all of the active points those extreme points have in common (think why?).

The geometric interpretation of active versus extreme points can be visually interpreted by looking at the faceted 3-D feasible region in one of our previous example. In the figure below, we have the following scenario:

- There are $n = 3$ variables (x_1, x_2, x_3)
- $\boldsymbol{x}^{(0)}$ and $\boldsymbol{x}^{(1)}$ are **adjacent extreme points**. They are connected by the **edge** defined by constraints G and C. Also note that G and C are $(n-1) = 2$ constraints, as required to define an edge.
- $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ are **adjacent extreme points**. They are connected by the **edge** defined by constraints C and B. Also note that C and B are $(n-1) = 2$ constraints, as required to define an edge.



Congratulations! You now have all the terminology and tools you need to develop an algorithm to solve linear programs! So why don't we just go right on ahead and do that!

9.3. Introduction to Linear Solution Algorithms

9.3.1. Brute Force Method

OK, so we have a very convenient interpretation of linear programs. We know that we can find the solution to **any LP** as one of its extreme points, AND we know that the extreme points are classified as the intersection of at least n linearly independent constraints. OK, so how about this idea:

> **LP Algorithm Idea (1)**
> The optimum to any LP can be found by determining the set of *all* extreme points, evaluating the objective function at each of them, and keeping the extreme point with the **best objective value**.

This is actually not a terrible idea! However, for the sake of argument let's list some **advantages** and **disadvantages** of this method:

- **AD –** Guaranteed to find the global optimum.
- **AD** – Simple to implement.
- **DISAD** – Not very elegant (if you're into that sort of thing).
- **DISAD** – Might have to visit *A LOT* of different points to make sure we have the answer.

Now, when I say we have to visit *a lot* of points. How many is that, exactly? Recall that not every constraint combination ends up giving us an extreme point, since a good deal of them intersect outside of our feasible region. However, when considering a solution algorithm we must always plan for the worst-case scenario. As it turns out, the **maximum number of extreme points** $\mathcal{N}_E$ for a given LP with n variables and m independent constraints is defined as:

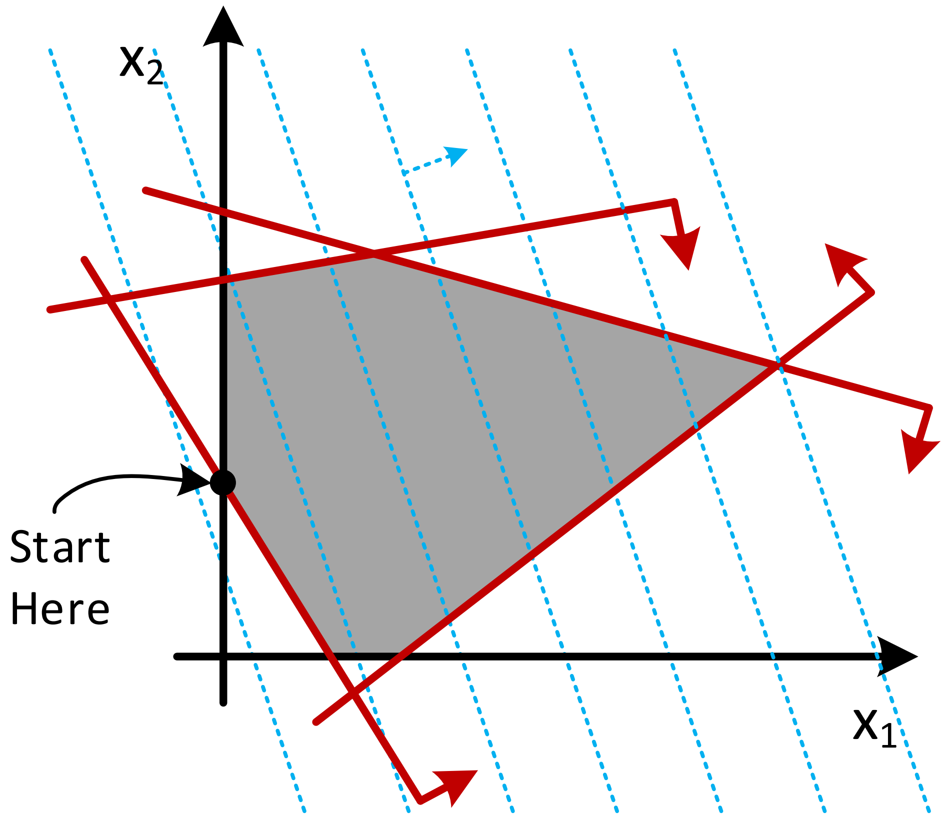
$$\mathcal{N}_E = \frac{n!}{(n-m)!\,m!}$$

So, if we consider a rather **small** problem of $n = 20$ variables and $m = 10$ independent constraints, we have a grand total of $\mathcal{N}_E = 184{,}756$ extreme points to worry about. Depending on the computation cost of a single extreme point, this could be done either very quickly or take quite some time! And, of course, as we extend beyond 20 variables and 10 constraints, things escalate very, very quickly. We can do **better**.

9.3.2. A Better Method

Consider the situation where we have a linear program like the one shown in the figure below. It is our task to start at the starting point and find the optimum (which we can see in this example… Where is it?) as efficiently as possible.

> **Class Workshop – Developing an LP Algorithm**
> Consider the linear program given in the figure below. Develop an algorithm that combines basic numerical methods with our new **geometric insight** for LPs that will find the optimum. Be sure to generalize your approach so that it extends to higher dimensional problems that we can't visualize. Then, draw/highlight how you would *expect* your algorithm to proceed for the given 2-D problem.



Workshop Solution – Developing an LP Algorithm

1. Start at an extreme point
2. Move along the edge that yields the greatest improvement (i.e. relax one constraint)
3. Continue moving until we hit another extreme point (i.e. search until 1 new constraint is activated)
4. Check again if there is another edge that will yield an improvement in the objective function. If so, continue, else, stop.

The algorithm that **you** just came up with is the basis for what is known as the ***Simplex Search***. We are just scratching the surface of the theory behind optimization, but for the purposes of this course, this is the extent of theory you will need to know. However, if you are interested in optimization, we highly suggest you take a course on optimization in your upper year (such as CHEMENG 4G03 where this material was taken from).

9.3.3. Linear Programs in Standard Form

Standard form will allow us to use matrices as a part of our Simplex algorithm. Linear Programs in standard form have the following characteristics:

- Only *equality* constraints (except for non-negativity).

- Only *positive* variables.
- The objective function and all constraints have like-terms collected so that each variable only appears on the LHS of any equation *once* and a single constant term (possibly zero) on the RHS.

Standard Linear Program

An optimization model is known as a **Standard Linear Program** if it obeys:

$\min_{\boldsymbol{x}} \phi = \boldsymbol{c}^T\boldsymbol{x}$	← Objective Function
$s.t.$	← "Subject to"
$A\boldsymbol{x} = \boldsymbol{b}$	← ONLY Equality Constraints
$\boldsymbol{x} \geq 0$	← Variable Bounds

We want to convert our system to standard form because **systems of linear equations (even under-defined systems) are easy to work with**. Moreover, *all linear programs can be converted to standard form*. We can convert to standard form by performing the following:

- Convert any ≤ and ≥ inequalities to equalities.
- Convert all non-positive variables to non-negative.
- Convert any unrestricted (URS) variables to non-negative.

9.3.3.1. Converting Inequalities to Equalities

A **less-than inequality** (≤) may be converted to an equality by the addition of a *slack variable*. A **slack variable (s^-)** is an **invented** variable that is non-negative. It does not affect the actual solution to the problem:

$$a_1x_1 + \cdots + a_nx_n \leq b$$
$$a_1x_1 + \cdots + a_nx_n + s^- = b$$

In this case, any combination of $x_1 \ldots x_n$ that would have given us a value less than b in the above constraint can have the variable s^- ADDED to it so that it now *equals* b. A greater-than inequality is handled in the same way, except now we subtract a **surplus variable (s^+)**:

$$a_1x_1 + \cdots + a_nx_n \geq b$$
$$a_1x_1 + \cdots + a_nx_n - s^+ = b$$

You can see that if we would have had a combination of variables resulting in a number greater than b, subtracting the (non-negative) surplus will give us exactly b. NOW… Ask yourself: Why can we be sure that each inequality we change to an equality does not run the risk of giving us an over-specified system?

9.3.3.2. Converting Non-Positive and URS Variables to Non-Negative

Perhaps even simpler than converting inequalities to equalities, we can convert non-positive variables (say – conversion of a reactant in a chemical reaction) to positive ones **via a simple substitution**:

$$x \leq 0 \Rightarrow y = -x$$
$$\therefore y \geq 0$$

There are actually two ways of ridding ourselves of unrestricted (URS) variables. The first is to perform a substitution in which we *eliminate* the unrestricted variable by re-writing it in terms of the others. That is not very much fun. Especially if the system of equations is quite large. Let's not do that. INSTEAD, we can replace an URS variable with **the difference between two fictitious positive variables**:

$$x \in \mathbb{R} \Rightarrow x = y_1 - y_2$$
$$y_1, y_2 \geq 0$$

Now if I wanted x in the equation above to be negative, it corresponds to a solution where $y_2 > y_1$, and vice-versa. Now, you may be asking yourself a couple of questions, like:

1. There are now infinitely many combinations of y_1 and y_2 that would give the same x. How do we know we are choosing the right ones?
2. Does the addition of slack, surplus and substitution variables have any impact on our solution?

The answers for which are:

1. This is true but remember that the *only* real variable here is x, which means that even if the other variables are random numbers, it is the combination of those numbers that I truly care about!
2. It has a*bsolutely no impact* on the solution to the problem, but any variables changed by substitution will have to be re-computed at the end to ensure that they may be interpreted correctly. For example, a solution that reports $y_1 = 4$ and $y_2 = 6$ is *actually* saying that $x = y_1 - y_2 = -2$.

9.4. Solving Linear Programing Problems with Software

Now that we have a better understanding of linear programs can be solved, we will look at how to use software such as Excel to solve linear programs. You can use any software or programming language such as Python of MATLAB to solve linear programs, but for the lecture notes, we will use Excel.

9.4.1. Linear Programming Example 1: Fireball Inc.

You own a company, Fireball Inc, that sells two devices, device A and device B. The cost to make device A is \$300, and the cost to make device B is \$400. The price you can sell device A is \$500 and the price you can sell device B is \$650. The demand for device A is 2700 units and for device B is 950 units. With your current manufacturing capacity, you can only make a total of 3000 devices (e.g. A+B). The maximum number of device A you can make is 3000 and the maximum number of device B you can make is 1000. How many of each device should you make the maximize the profit?

The information can be summarized by the following tables:

Device	Cost (c)	Sale Price (p)	Demand	# to make
A	\$300	\$500	2700	x_1
B	\$400	\$650	950	x_2

We can write our linear program as:

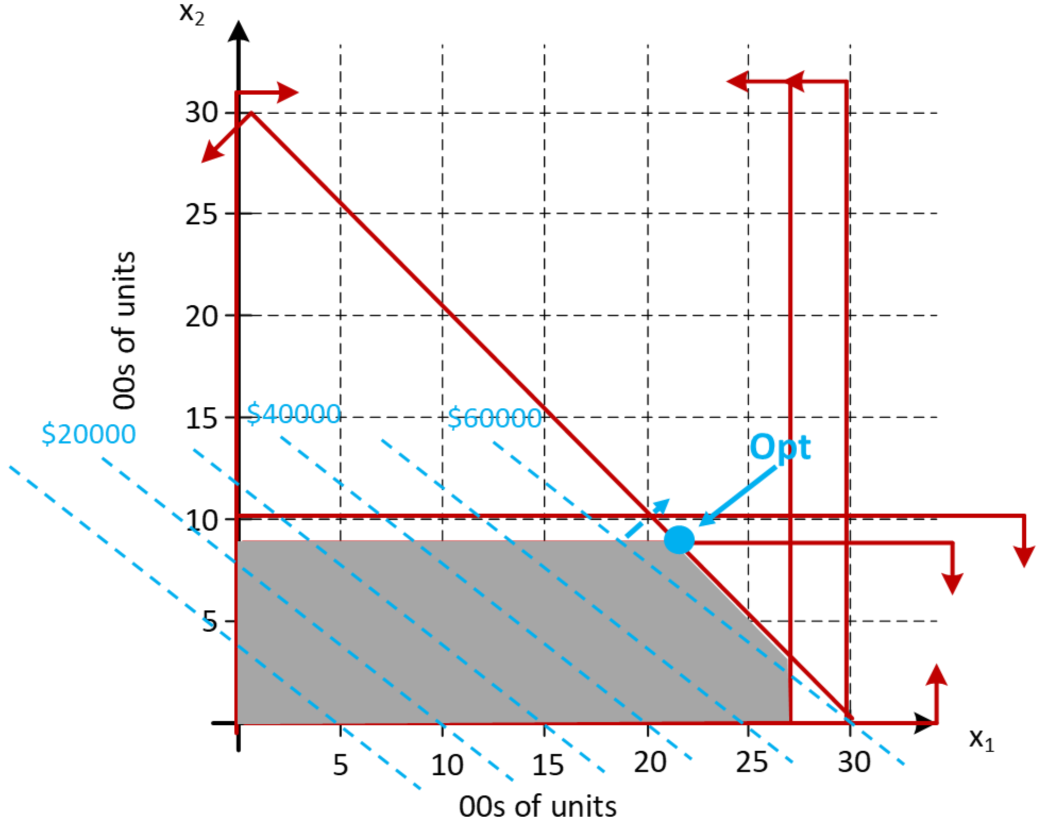
$$\max_{x} \phi = \sum_{1}^{2} (x_i p_i - x_i c_i)$$

$$s.t.$$

$$x_1 + x_2 \leq 3000$$
$$0 \leq x_1 \leq 2700$$
$$0 \leq x_2 \leq 950$$

Notice how the variable bound does not include the individual device manufacturing capacity. This is because the demand the individual device is smaller than the manufacturing capacity for the individual device. Constraints that are not involved in formatting the feasible set are called *redundant* constraints. It is good to keep them even though the are redundant, since if the situation changes, they may become non-redundant. For example, if the demand increases, then the individual device manufacturing capacity may become relevant to the problem.

We can solve this problem graphically as shown below, which gives the optimal solution as 2050 device A and 950 device B for a profit of $647,500.



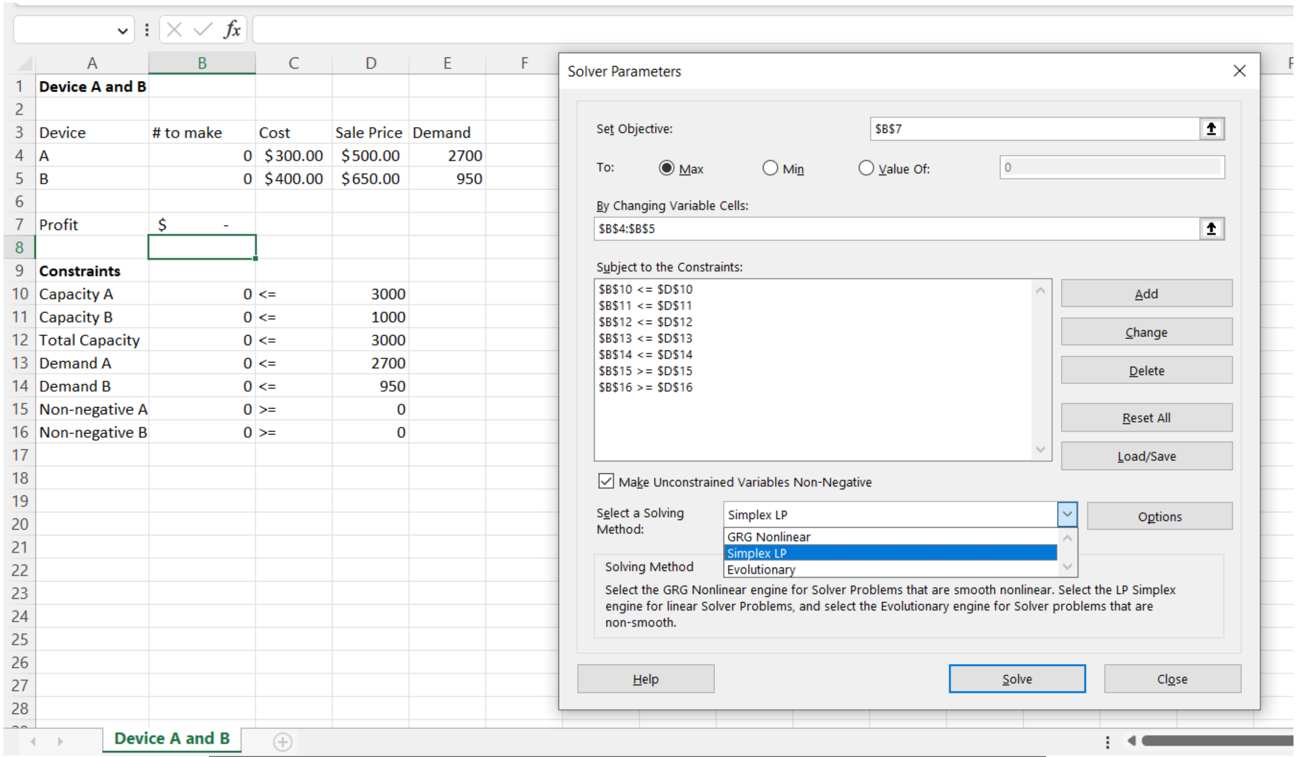
We can also solve this problem using Excel as shown below.

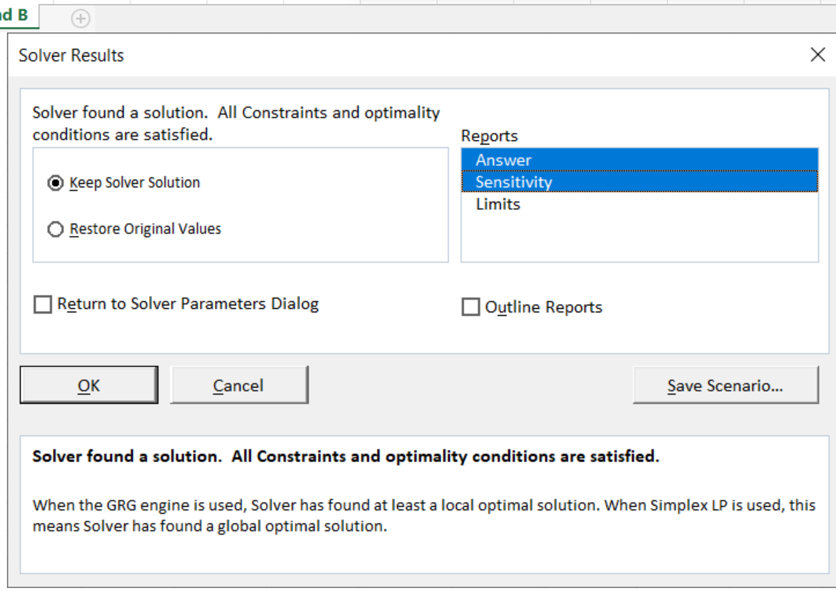
E16

	A	B	C	D	E	F
1	**Device A and B**					
2						
3	Device	# to make	Cost	Sale Price	Demand	
4	A	0	$300.00	$500.00	2700	
5	B	0	$400.00	$650.00	950	
6						
7	Profit	$ -				
8						
9	**Constraints**					
10	Capacity A	0	<=	3000		
11	Capacity B	0	<=	1000		
12	Total Capacity	0	<=	3000		
13	Demand A	0	<=	2700		
14	Demand B	0	<=	950		
15	Non-negative A	0	>=	0		
16	Non-negative B	0	>=	0		
17						

The formula for the profit cell (B7) is "=SUMPRODUCT(B4:B5,D4:D5)-SUMPRODUCT(B4:B5,C4:C5)"

Note that we do not need to list the constraints as we have shown above, but it is helpful to write out the constraints so that it is clear how we have defined the problem.
After setting up the Excel sheet, we can use Solver (Data -> Analyze -> Solver) to solve the linear program. In solver, we set the objective to maximize profit ($B$7). The under "By Changing Variable Cells:" we select the decision variable vector ($B$4:$B$5). We then add all the constraints of the linear program. Note that we did not have to add the non-negative constraints explicitly, we can just check the "Make Unconstrained Variables Non-Negative" box. After all the constraints are entered, we can select the solving method. In our case, we will choose "Simplex LP". This will perform a Simplex Search to solve our linear program. Click solve and select the Answer and Sensitivity Reports.





The solution returned by Solver is 2050 device A and 950 device B for a profit of $647,500. Which is the same as our graphical solution.

	A	B	C	D	E
1	**Device A and B**				
2					
3	Device	# to make	Cost	Sale Price	Demand
4	A	2050	$300.00	$500.00	2700
5	B	950	$400.00	$650.00	950
6					
7	Profit	$647,500.00			
8					
9	**Constraints**				
10	Capacity A	2050	<=	3000	
11	Capacity B	950	<=	1000	
12	Total Capacity	3000	<=	3000	
13	Demand A	2050	<=	2700	
14	Demand B	950	<=	950	
15	Non-negative A	2050	>=	0	
16	Non-negative B	950	>=	0	
17					

9.4.1.1. LP Solution Analysis

Let us look at the LP reports generated by Excel. You should see two new sheets: “Answer Report 1” and “Sensitivity Report 1”. We will look at the Answer Report first.

Microsoft Excel 16.0 Answer Report
Worksheet: [Wk-9 Optimization LP.xlsx]Device A and B
Report Created: 2023-03-11 8:20:36 AM
Result: Solver found a solution. All Constraints and optimality conditions are satisfied.
Solver Engine
Engine: Simplex LP
Solution Time: 0.125 Seconds.
Iterations: 4 Subproblems: 0
Solver Options
Max Time Unlimited, Iterations Unlimited, Precision 0.000001, Use Automatic Scaling
Max Subproblems Unlimited, Max Integer Sols Unlimited, Integer Tolerance 1%, Assume NonNegative

Objective Cell (Max)

Cell	Name	Original Value	Final Value
$B$7	Profit # to make	$ -	$ 647,500.00

Variable Cells

Cell	Name	Original Value	Final Value	Integer
$B$4	A # to make	0	2050	Contin
$B$5	B # to make	0	950	Contin

Constraints

Cell	Name	Cell Value	Formula	Status	Slack
$B$10	Capacity A # to make	2050	$B$10<=$D$10	Not Binding	950
$B$11	Capacity B # to make	950	$B$11<=$D$11	Not Binding	50
$B$12	Total Capacity # to make	3000	$B$12<=$D$12	Binding	0
$B$13	Demand A # to make	2050	$B$13<=$D$13	Not Binding	650
$B$14	Demand B # to make	950	$B$14<=$D$14	Binding	0
$B$15	Non-negative A # to make	2050	$B$15>=$D$15	Not Binding	2050
$B$16	Non-negative B # to make	950	$B$16>=$D$16	Not Binding	950

At the top of the sheet, it gives us the computation details. In our case, solver took 0.125 seconds and found the optimal solution in 4 iterations.

The first table shows the objective function, which is profit in our case. The solution shows that the optimum profit is $647,500.

The second table shows the **decision variables**. The **original values** for the variables were 0. The optimal value is 2050 for device A and 950 for device B. The last column shows that the variables were treated as **continuous**. Note that technically the number of device to make is an integer, but due to the large number of device we can treat the variables as continuous. Also, fortunately the solution were integer values, so we do not have a problem. We will look at how to deal with integer variables in a later example.

The last table shows the **constraints**. It shows us the cell value for each constraint, the formula, the status, and the slack. We will focus on the **status** and the slack. The status of the constraints shows which constraints are **active (or binding)**. Since we have a 2-dimensional problem (i.e. 2 decision variables) the number of constraints that are active should be 2 or more. In our case, the total manufacturing capacity of A and B together and the demand for B are our active constraints. This means that if we want to improve our profit, we need to either increase the total manufacturing capacity of A and B together or increase the demand for B.

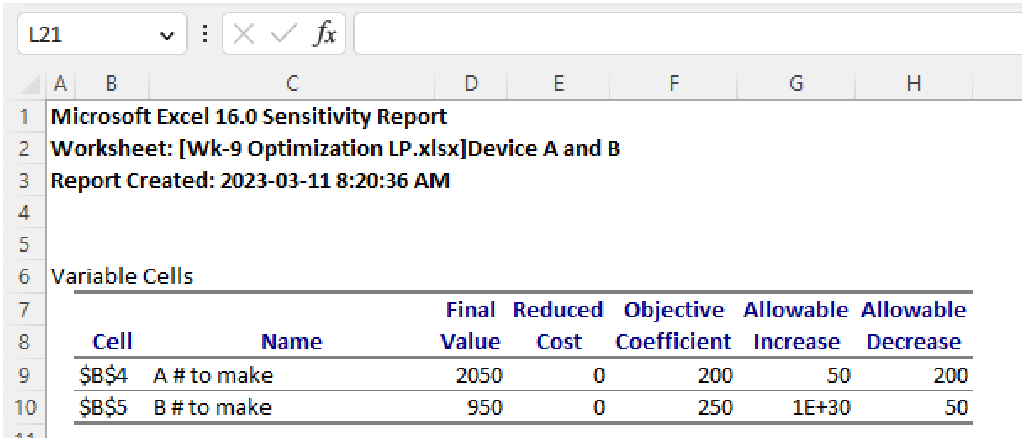
Since this was a 2-D problem, we could see this result easily on our graph. However, with higher dimension problems it is much more difficult (almost impossible) to visualize and this report is helpful to show us which constraints to focus on.

The last column shows the **slack** of the constraint. The slack of a constraint (or slack variable) shows how far the optimal solution is from the constraint.

All binding constraints will have a slack of 0 (i.e. the optimal solution is at the constraint). The manufacturing capacity of device A alone has a slack of 950. This means that the optimal solution will not change even if we decrease the manufacturing capacity of device A by 950. Similarly, the Demand for device A has a slack of 650. This means that even if the demand for device A decrease by 650, it will not change our optimal solution.

9.4.1.2. LP Sensitivity Analysis

We will now look at the Sensitivity Report generated by Solver.



Microsoft Excel 16.0 Sensitivity Report
Worksheet: [Wk-9 Optimization LP.xlsx]Device A and B
Report Created: 2023-03-11 8:20:36 AM

Variable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
$B$4	A # to make	2050	0	200	50	200
$B$5	B # to make	950	0	250	1E+30	50

A **sensitivity analysis** on a LP model is used to analyze how the model's constraints affect the optimal solution for the objective function.

In the first table (screenshot above), it shows the decision variables. The final value for the number of Device A is 2050 and the number of Device B is 950. The next column shows the **reduced cost**. The reduced cost is the amount the objective function will change if you tighten the variable bounds (i.e. increase the lower bound or decrease the upper bound). For example, if the reduced cost of a variable is -10. This means that if we increase the lower bound of the variable by 1 unit the value of the objective function will decrease by 10.

For our problem, the lower bound for both variables is 0 and the upper bound is 3000 for device A and 1000 for device B. Since the optimal solution (# of Device A=2050, # of Device B=950) is not at the variable bounds the reduced cost for both variables are 0. In general, the reduced cost of a variable is 0 if the optimal solution is not at the variable bound. The reduced cost will only be non-zero when the variable's value is equal to its upper or lower bound at the optimal solution.

The second column of the table shows the **objective coefficient.** This is the coefficients for our decision variables. Recalling the notation: $\phi = \boldsymbol{c}^T \boldsymbol{x}$**.** The objective coefficient column is the vector *c*. Notice that even though we entered the sale price and the cost separately, solver merged the two into one coefficient vector. The objective coefficient in this case would be the gross profit per device (i.e. sale price – cost).

The third and fourth columns show the allowable increase and allowable decrease for the objective coefficients. These numbers indicate how much the objective coefficient must increase or decrease before the optimal solution changes. Note that when you change the objective coefficient within the allowable increase/decrease range, the value for the objective function (i.e. profit) will change but the solution (i.e. 2050 device A and 950 device B) does not change.

For our problem, we can increase the gross profit of Device A by $50 without affecting the solution. So for example, if we increase the sale price of Device A from $500 to $550, the optimal solution is still to make 2050 of Device A and 950 of Device B. However, the profit would increase from $647,500 to $750,000.

Note that the allowable increase for Device B is 1E+30. This means that you can increase the objective coefficient for Device B by any amount (up to infinity) and the optimal solution would not change.

It is also important to note that allowable increase and allowable decrease for an objective coefficient assumes that no other coefficients are changing. For example, if we increase the objective coefficient of Device A by 30 and decrease the objective coefficient for Device B by 30, this may change our optimal solution even though both changes were within their allowable increase/decrease respectively.

If there are simultaneous changes to the objective coefficients, then we can apply the **100% rule** to determine if the optimal solution would change. The 100% rule states that if $\sum \frac{\text{Proposed Changes}}{\text{Allowable Changes}} \leq 100\%$ then the optimal solution would not change.

For example, if we increase the objective coefficient of Device A from 200 to 220 and decrease the objective coefficient of Device B from 250 to 225 would the optimal solution change? We can apply the 100% rule:

$$\sum \frac{\text{Proposed Changes}}{\text{Allowable Changes}} = \frac{20}{50} + \frac{-25}{-50} = 90\% < 100\%$$

Therefore, the optimal solution would not change even though our profit would change. It will still be best to make 2050 Device A and 950 Device B.

Let us now, look at the sensitivity report for the constraints.

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
$B$10	Capacity A # to make	2050	0	3000	1E+30	950
$B$11	Capacity B # to make	950	0	1000	1E+30	50
$B$12	Total Capacity # to make	3000	200	3000	650	2050
$B$13	Demand A # to make	2050	0	2700	1E+30	650
$B$14	Demand B # to make	950	50	950	50	650
$B$15	Non-negative A # to make	2050	0	0	2050	1E+30
$B$16	Non-negative B # to make	950	0	0	950	1E+30

The **Final Value** shows the value of the constraint at the optimal solution. This is compared to the third column (**Constraint R.H. Side**). The difference between Constraint R.H. Side and the Final Value gives the slack (or surplus, if the difference is negative) of the constraint.

The **Shadow Price** of a constraint is the marginal improvement of the objective function value if the RHS of the constraint is increased by 1 unit while holding all other constraints constant. All inactive constraints will have a shadow price of zero. In our problem, the shadow price is 200 for the total capacity constraint and 50 for the demand B constraint. This makes intuitive sense since increasing the total capacity means we can make more Device A (since device B is at a constraint already) and the gross profit per Device A is $200. Similarly, if we increase the demand for Device B by 1, it will increase our profit by $50, because the gross profit of Device B is $50 more than Device A. Therefore, if the demand for Device B increases by 1, we would make 1 more Device B and 1 less Device A for a $50 gain in profit.

The **Allowable Increase** and **Allowable Decrease** columns show how much you can change the constraint before the shadow price changes. For example, the Allowable Increase for Total Capacity is 650. This means that we can increase the total capacity by 650 units without changing the shadow price. This makes sense because if we increase the total capacity by 650 units to 2700 units, we will hit another constraint (Demand A ≤ 2700). The 100% rule also applies for the allowable changes in constraints. If $\sum \frac{\text{Proposed Changes}}{\text{Allowable Changes}} \leq 100\%$ then the shadow price will not change.

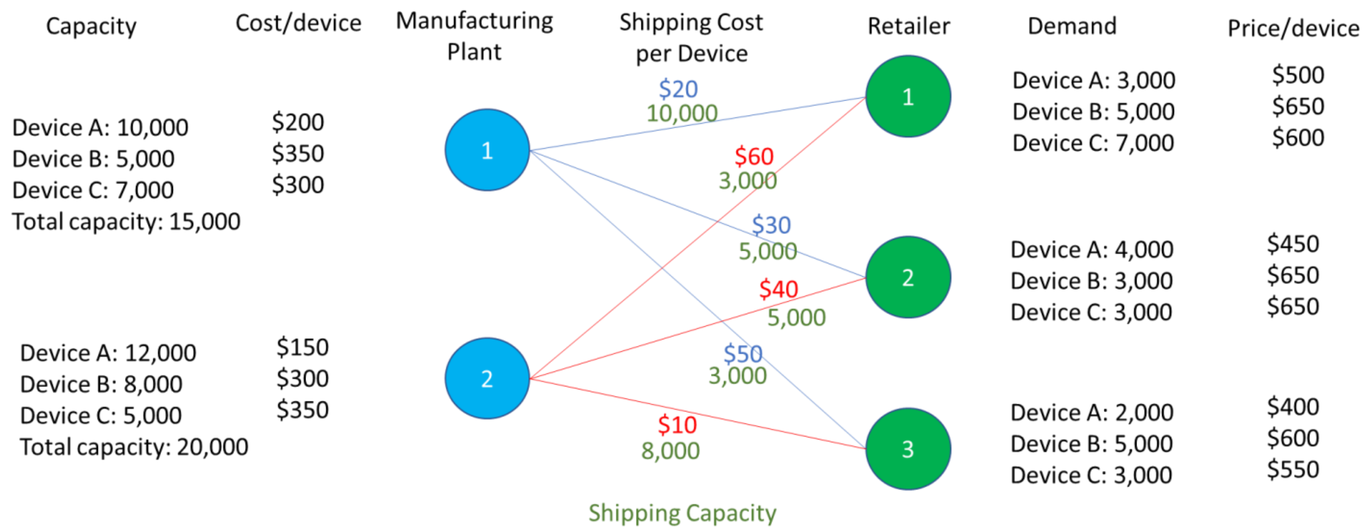
LP sensitivity analysis can be used to help us make decisions. For example, you want to know where to invest money to improve profits. The sensitivity analysis shows us that we should either increase the total capacity or the demand for B since they are the active constraints. This means we should consider investing in either more manufacturing capacity or advertising for device B.

For simplicity's sake, let's say the cost for increasing total manufacturing capacity is $100/unit capacity and the cost for advertising is $25 to increase the demand for Device B by 1. What should you do?

Based on the sensitivity analysis, we know the shadow price (or marginal gain) for total capacity is $200. This means that increasing the total capacity by 1 unit will increase the profit by $200. If it costs $100/unit to increase capacity this means that the investment in increasing capacity will return $100/unit. Similarly, for the demand of Device B, the shadow price is $50. If it costs $25 increase the demand by 1 device, then the investment return in advertisement would be $25/unit. Therefore, it would be better to invest in increasing total manufacturing capacity. In fact, you can increase the total manufacturing capacity by 650 units (allowable increase) before a change in the shadow price (marginal gain).

9.4.2. Linear Programming Example 2: Fireball Inc. Distribution

The previous example had 2 decision variables and could be easily solved graphically or even intuitively. Let us look at a more complex problem. Your company, Fireball Inc., had established itself as a major player in the market. You have expanded your product line to 3 devices (A, B and C). Also, you now have 2 manufacturing plants. Each manufacturing has its own manufacturing capacity and cost. Furthermore, your company had decided to be a wholesaler and not sell directly to retail consumers. Your company sells to 3 retailers. Each retailer purchases your devices at different prices and the delivery cost from each of your manufacturing plants to the retailers also vary. The details of the problem can be summarized by the following diagram.



Sometimes it is hard to know where to start when face with a problem that seems complicated (in reality, this is quite a small problem). Let us define our objective in words first. Our objective is to maximize our profit by determining how many of each device (A, B, and C), should be produced at each manufacturing plant (1 and 2) for each retailer (1, 2, and 3). When we define the objective this way, it is easier to determine what our decision variables are. In our case, we can call our decision variable vector X, and each decision variable is $X_{i,j,k}$, where i represents the device, j represents the manufacturing plant, and k represents the retailer. For example, $X_{\mathrm{B},2,3}$ is the number of device B manufactured at plant 2 for retailer 3. Thus, we have 18 decision variables in total. The objective function is $\max_{\boldsymbol{x}} \phi = \boldsymbol{c}^T \boldsymbol{X}$, where $\boldsymbol{c}$ is the vector of profit per device (i.e. Price – Cost – Shipping).

Now that we have defined the objective and decision variables, we need to define the constraints. Below are the constraints for this problem. We can assume that we don't need to meet the demand of the retailers (i.e. we can supply less than the demand).

Non-negative Constraints → $X_{i,j,k} \geq 0 \;\forall\; i,j,k$

Demand Constraints

$$\sum_{j=1}^{2} X_{A,j,1} \leq 3{,}000$$
$$\sum_{j=1}^{2} X_{A,j,2} \leq 4{,}000$$
$$\sum_{j=1}^{2} X_{A,j,3} \leq 2{,}000$$
$$\sum_{j=1}^{2} X_{B,j,1} \leq 5{,}000$$
$$\sum_{j=1}^{2} X_{B,j,2} \leq 3{,}000$$
$$\sum_{j=1}^{2} X_{B,j,3} \leq 5{,}000$$
$$\sum_{j=1}^{2} X_{C,j,1} \leq 7{,}000$$
$$\sum_{j=1}^{2} X_{C,j,2} \leq 3{,}000$$
$$\sum_{j=1}^{2} X_{C,j,3} \leq 3{,}000$$

Manufacturing Capacity Constraints

$$\sum_{k=1}^{3} X_{A,1,k} \leq 10{,}000$$
$$\sum_{k=1}^{3} X_{A,2,k} \leq 12{,}000$$
$$\sum_{k=1}^{3} X_{B,1,k} \leq 5{,}000$$
$$\sum_{k=1}^{3} X_{B,2,k} \leq 8{,}000$$
$$\sum_{k=1}^{3} X_{C,1,k} \leq 7{,}000$$
$$\sum_{k=1}^{3} X_{C,2,k} \leq 5{,}000$$
$$\sum_{i=A}^{C} \sum_{k=1}^{3} X_{i,1,k} \leq 15{,}000$$
$$\sum_{i=A}^{C} \sum_{k=1}^{3} X_{i,2,k} \leq 20{,}000$$

Shipping Capacity Constraints

$$\sum_{i=A}^{C} X_{i,1,1} \leq 10{,}000$$
$$\sum_{i=A}^{C} X_{i,1,2} \leq 5{,}000$$
$$\sum_{i=A}^{C} X_{i,1,3} \leq 3{,}000$$
$$\sum_{i=A}^{C} X_{i,2,1} \leq 3{,}000$$
$$\sum_{i=A}^{C} X_{i,2,2} \leq 5{,}000$$
$$\sum_{i=A}^{C} X_{i,2,3} \leq 8{,}000$$

Now that we have formulated our problem, we can set this problem up in Excel.

O18

	A	B	C	D	E	F	G	H	I	J	K
1	Variable Index	Device	Plant	Retailer	Cost	Shipping	Price	Profit/device	Devices Made		Total Profit
2	A11	A	1	1	$ 200.00	$ 20.00	$ 500.00	$ 280.00	0		0
3	A12	A	1	2	$ 200.00	$ 30.00	$ 450.00	$ 220.00	0		
4	A13	A	1	3	$ 200.00	$ 50.00	$ 400.00	$ 150.00	0		
5	A21	A	2	1	$ 150.00	$ 60.00	$ 500.00	$ 290.00	0		
6	A22	A	2	2	$ 150.00	$ 40.00	$ 450.00	$ 260.00	0		
7	A23	A	2	3	$ 150.00	$ 10.00	$ 400.00	$ 240.00	0		
8	B11	B	1	1	$ 350.00	$ 20.00	$ 650.00	$ 280.00	0		
9	B12	B	1	2	$ 350.00	$ 30.00	$ 650.00	$ 270.00	0		
10	B13	B	1	3	$ 350.00	$ 50.00	$ 600.00	$ 200.00	0		
11	B21	B	2	1	$ 300.00	$ 60.00	$ 650.00	$ 290.00	0		
12	B22	B	2	2	$ 300.00	$ 40.00	$ 650.00	$ 310.00	0		
13	B23	B	2	3	$ 300.00	$ 10.00	$ 600.00	$ 290.00	0		
14	C11	C	1	1	$ 300.00	$ 20.00	$ 600.00	$ 280.00	0		
15	C12	C	1	2	$ 300.00	$ 30.00	$ 650.00	$ 320.00	0		
16	C13	C	1	3	$ 300.00	$ 50.00	$ 550.00	$ 200.00	0		
17	C21	C	2	1	$ 350.00	$ 60.00	$ 600.00	$ 190.00	0		
18	C22	C	2	2	$ 350.00	$ 40.00	$ 650.00	$ 260.00	0		
19	C23	C	2	3	$ 350.00	$ 10.00	$ 550.00	$ 190.00	0		
20											

We have set up our variable vector ("Devices Made") and the corresponding objective coefficient vector ("Profit/device). The "Total Profit" cell is our objective function, which is the sum of the products of Profit/device and Devices Made. We can now enter our constraints in Excel.

	A	B	C	D	E	F
20						
21	Constraints					
22	Name	Value		RHS	Formula	
23	A Demand 1	0	<=	3,000	I2+I5	
24	A Demand 2	0	<=	4000	I3+I6	
25	A Demand 3	0	<=	2000	I4+I7	
26	B Demand 1	0	<=	5000	I8+I11	
27	B Demand 2	0	<=	3000	I9+I12	
28	B Demand 3	0	<=	5000	I10+I13	
29	C Demand 1	0	<=	7000	I14+I17	
30	C Demand 2	0	<=	3000	I15+I18	
31	C Demand 3	0	<=	3000	I16+I19	
32	A Capacity 1	0	<=	10000	SUM(I2:I4)	
33	A Capacity 2	0	<=	12000	SUM(I5:I7)	
34	B Capacity 1	0	<=	5000	SUM(I8:I10)	
35	B Capacity 2	0	<=	8000	SUM(I11:I13)	
36	C Capacity 1	0	<=	7000	SUM(I14:I16)	
37	C Capacity 2	0	<=	5000	SUM(I17:I19)	
38	Total Capacity 1	0	<=	15000	SUM(I2:I4,I8:I10,I14:I16)	
39	Total Capacity 2	0	<=	20000	SUM(I5:I7,I11:I13,I17:I19)	
40	Shipping Capacit	0	<=	10000	I2+I8+I14	
41	Shipping Capacit	0	<=	5000	I3+I9+I15	
42	Shipping Capacit	0	<=	3000	I4+I10+I16	
43	Shipping Capacit	0	<=	3000	I5+I11+I17	
44	Shipping Capacit	0	<=	5000	I6+I12+I18	
45	Shpping Capacity	0	<=	8000	I7+I13+I19	
46						

Using Solver to solve the problem gives the following Answer Report:

Objective Cell (Max)

Cell	Name	Original Value	Final Value
$K$2	A Total Profit	0	8540000

Variable Cells

Cell	Name	Original Value	Final Value	Integer
$I$2	A11 Devices Made	0	1000	Contin
$I$3	A12 Devices Made	0	2000	Contin
$I$4	A13 Devices Made	0	0	Contin
$I$5	A21 Devices Made	0	2000	Contin
$I$6	A22 Devices Made	0	2000	Contin
$I$7	A23 Devices Made	0	2000	Contin
$I$8	B11 Devices Made	0	5000	Contin
$I$9	B12 Devices Made	0	0	Contin
$I$10	B13 Devices Made	0	0	Contin
$I$11	B21 Devices Made	0	0	Contin
$I$12	B22 Devices Made	0	3000	Contin
$I$13	B23 Devices Made	0	5000	Contin
$I$14	C11 Devices Made	0	4000	Contin
$I$15	C12 Devices Made	0	3000	Contin
$I$16	C13 Devices Made	0	0	Contin
$I$17	C21 Devices Made	0	1000	Contin
$I$18	C22 Devices Made	0	0	Contin
$I$19	C23 Devices Made	0	1000	Contin

Constraints

Cell	Name	Cell Value	Formula	Status	Slack
$B$23	A Demand 1 Value	3000	$B$23<=$D$23	Binding	0
$B$24	A Demand 2 Value	4000	$B$24<=$D$24	Binding	0
$B$25	A Demand 3 Value	2000	$B$25<=$D$25	Binding	0
$B$26	B Demand 1 Value	5000	$B$26<=$D$26	Binding	0
$B$27	B Demand 2 Value	3000	$B$27<=$D$27	Binding	0
$B$28	B Demand 3 Value	5000	$B$28<=$D$28	Binding	0
$B$29	C Demand 1 Value	5000	$B$29<=$D$29	Not Binding	2000
$B$30	C Demand 2 Value	3000	$B$30<=$D$30	Binding	0
$B$31	C Demand 3 Value	1000	$B$31<=$D$31	Not Binding	2000
$B$32	A Capacity 1 Value	3000	$B$32<=$D$32	Not Binding	7000
$B$33	A Capacity 2 Value	6000	$B$33<=$D$33	Not Binding	6000
$B$34	B Capacity 1 Value	5000	$B$34<=$D$34	Binding	0
$B$35	B Capacity 2 Value	8000	$B$35<=$D$35	Binding	0
$B$36	C Capacity 1 Value	7000	$B$36<=$D$36	Binding	0
$B$37	C Capacity 2 Value	2000	$B$37<=$D$37	Not Binding	3000
$B$38	Total Capacity 1 Value	15000	$B$38<=$D$38	Binding	0
$B$39	Total Capacity 2 Value	16000	$B$39<=$D$39	Not Binding	4000
$B$40	Shipping Capacity 1,1 Value	10000	$B$40<=$D$40	Binding	0
$B$41	Shipping Capacity 1,2 Value	5000	$B$41<=$D$41	Binding	0
$B$42	Shipping Capacity 1,3 Value	0	$B$42<=$D$42	Not Binding	3000
$B$43	Shipping Capacity 2,1 Value	3000	$B$43<=$D$43	Binding	0
$B$44	Shipping Capacity 2,2 Value	5000	$B$44<=$D$44	Binding	0
$B$45	Shpping Capacity 2,3 Value	8000	$B$45<=$D$45	Binding	0

And the Sensitivity Report is shown below:

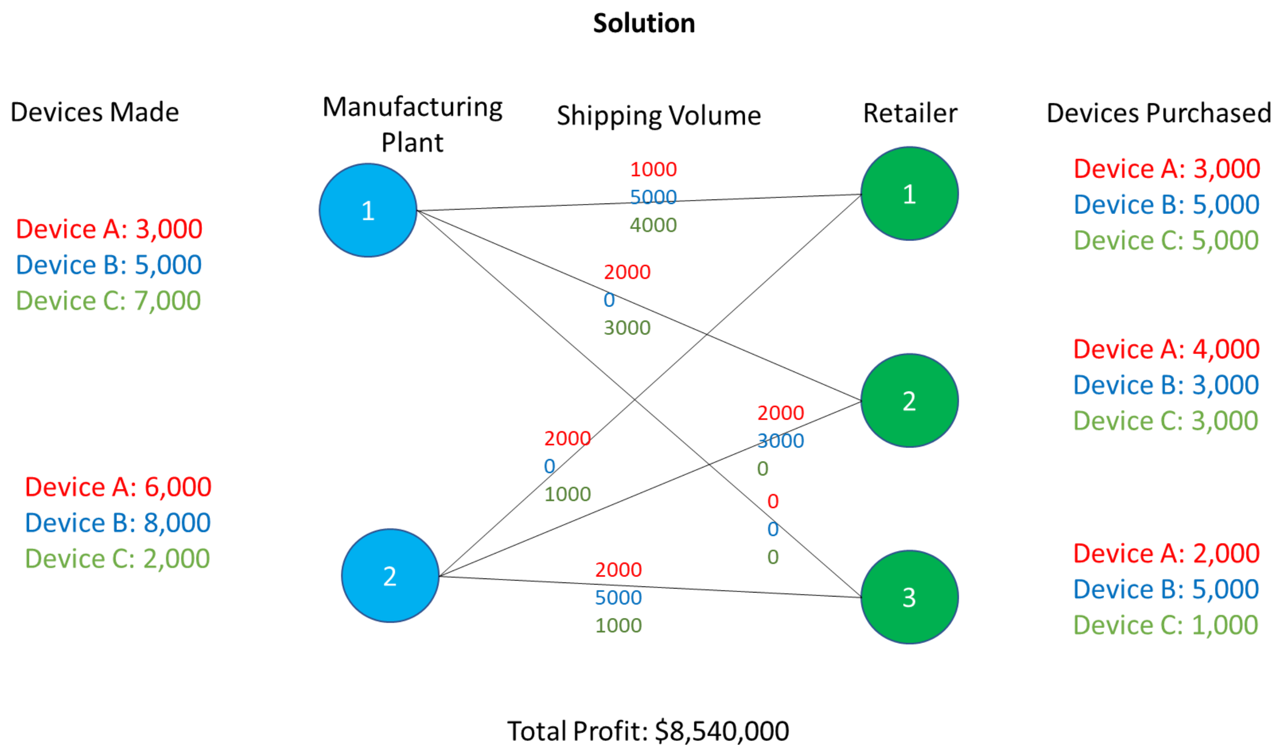
Variable Cells

	Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
9	$I$2	A11 Devices Made	1000	0	280	0	0
10	$I$3	A12 Devices Made	2000	0	220	0	0
11	$I$4	A13 Devices Made	0	0	150	0	1E+30
12	$I$5	A21 Devices Made	2000	0	290	0	0
13	$I$6	A22 Devices Made	2000	0	260	0	0
14	$I$7	A23 Devices Made	2000	0	240	1E+30	0
15	$I$8	B11 Devices Made	5000	0	280	1E+30	0
16	$I$9	B12 Devices Made	0	0	270	0	0
17	$I$10	B13 Devices Made	0	0	200	0	1E+30
18	$I$11	B21 Devices Made	0	0	290	0	1E+30
19	$I$12	B22 Devices Made	3000	0	310	0	0
20	$I$13	B23 Devices Made	5000	0	290	1E+30	0
21	$I$14	C11 Devices Made	4000	0	280	0	0
22	$I$15	C12 Devices Made	3000	0	320	0	0
23	$I$16	C13 Devices Made	0	0	200	80	0
24	$I$17	C21 Devices Made	1000	0	190	0	0
25	$I$18	C22 Devices Made	0	0	260	0	1E+30
26	$I$19	C23 Devices Made	1000	0	190	0	190

Constraints

	Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
31	$B$23	A Demand 1 Value	3000	100	3000	1000	2000
32	$B$24	A Demand 2 Value	4000	0	4000	1000	0
33	$B$25	A Demand 3 Value	2000	50	2000	1000	2000
34	$B$26	B Demand 1 Value	5000	50	5000	0	0
35	$B$27	B Demand 2 Value	3000	0	3000	1E+30	0
36	$B$28	B Demand 3 Value	5000	50	5000	0	0
37	$B$29	C Demand 1 Value	5000	0	7000	1E+30	2000
38	$B$30	C Demand 2 Value	3000	0	3000	1E+30	0
39	$B$31	C Demand 3 Value	1000	0	3000	1E+30	2000
40	$B$32	A Capacity 1 Value	3000	0	10000	1E+30	7000
41	$B$33	A Capacity 2 Value	6000	0	12000	1E+30	6000
42	$B$34	B Capacity 1 Value	5000	50	5000	0	0
43	$B$35	B Capacity 2 Value	8000	50	8000	0	0
44	$B$36	C Capacity 1 Value	7000	100	7000	1000	2000
45	$B$37	C Capacity 2 Value	2000	0	5000	1E+30	3000
46	$B$38	Total Capacity 1 Value	15000	100	15000	2000	0
47	$B$39	Total Capacity 2 Value	16000	0	20000	1E+30	4000
48	$B$40	Shipping Capacity 1,1 Value	10000	80	10000	0	2000
49	$B$41	Shipping Capacity 1,2 Value	5000	120	5000	0	2000
50	$B$42	Shipping Capacity 1,3 Value	0	0	3000	1E+30	3000
51	$B$43	Shipping Capacity 2,1 Value	3000	190	3000	2000	1000
52	$B$44	Shipping Capacity 2,2 Value	5000	260	5000	0	1000
53	$B$45	Shpping Capacity 2,3 Value	8000	190	8000	2000	1000

Another way to summarize the solution is to show it in a diagram:



From the LP analysis, we see that the optimal profit is $8,540,000. We also notice that Plant 2 is not manufacturing at full capacity and retailers 1 and 3 demand for device C is not met. So, what can we do to increase the profit?

From the sensitivity analysis, we can see that the highest shadow price is for shipping capacity from plant 2 to retailer 2 ($260), followed by shipping capacity from plant 2 to retailer 1 and to retailer 3 ($190). This provides us a good starting place to improve our operations. If we look more carefully, we can see that the allowable increase for the shipping capacity from plant 2 to retailer 2 is 0. Therefore, increasing the capacity by even 1 unit may change the shadow price. You can download the Excel file from Avenue, and you can give this a try and see what happens to the shadow price.

For shipping capacity from plant 2 to retailer 2 and retailer 3, both have a shadow price of $190 and an allowable increase of 2000. This means that we can increase the profit by $190/device for a maximum of 2000 devices if we increase the shipping capacity from plant 2 to retailer 1 or to retailer 3. Also, it is important to be mindful of the 100% rule, this means that we cannot increase the capacity of both shipping routes by 2000 and still expect the same shadow price since this would violate the 100% rule. Lastly, once you have identified a possible improvement from the model, you will need to assess if implementing the improvement makes sense. For example, how much would it cost to increase shipping capacity? If it costs less than $190/device, then it may be something that you would like to pursue. However, if it costs more than $190/device, then it is not worth it to increase the shipping capacity.

This example shows the power of linear programming as long as you formulate the problem properly. At first glance, this problem is complicated with many variables and constraints. (Side note: this problem is actually quite small compared to real life problems, nonetheless, LP can be applied to much larger problem than this.) However, using LP, we were able to identify the optimal manufacturing and distribution allocations. We have also identified a possible avenue to improve profit. Moreover, LP allows us to understand our problem better. For our case, we can see that many of the objective coefficients have 0 allowable increase and decreases. This indicates that your optimal solution will be sensitive to small changes to objective coefficients (i.e. costs/price of device). If you have time, you can play around with the costs and price of the device and see how sensitive the solution is to those changes.

9.4.3. Examples of Common LP Problems

Here is a list of examples of common LP problems to help you with formulating a LP for your project:

- Scheduling Problem: https://machinelearninggeek.com/solving-staff-scheduling-problem-using-linear-programming/#Staff_Scheduling_Problem
- CPM: https://www.linearprogramming.info/tag/cpm/
- Distribution/Transportation: See 9.4.2
- Capital Budgeting: https://zhijingeu.medium.com/optimizing-capital-budgeting-using-linear-programming-tools-dfd1dec5a64d
- Investment planning: https://www.mathworks.com/help/optim/ug/maximize-long-term-investments-using-linear-programming.html
- Blending Problems: https://www.ibm.com/docs/ja/icos/12.9.0?topic=programming-blending-problems
- Resource Allocation: https://utw11041.utweb.utexas.edu/ORMM/models/unit/linear/subunits/resource_allocation/index.html
- Inventory Optimization: https://medium.com/@gmarchetti/linear-programming-for-inventory-optimization-64aa674a13cc
- Multi-period scheduling: https://s2.smu.edu/~olinick/cse3360/lectures/l5/l5.html
- Network Flow: https://optimization.cbe.cornell.edu/index.php?title=Network_flow_problem