Analysis of Linear, Integer, and Binary Programming and their Applications

By Yusuke Motozawa

Fall, 2009

In Partial Fulfillment of Math 4395 – Senior Project

Department of Computer and Mathematical Sciences
University of Houston - Downtown

Faculty Advisor:
Dr. Tim Redl ________________________

Committee member:
Dr. Carol Vobach ________________________

Committee member:
Dr. Yunwei Cui ________________________

Department Chairman:
Dr. Dennis Rodriguez ________________________

Table of Contents

- **Figures and Tables**

- **Abstract**

My research mainly focuses on the field of operations research. Operations research is a mathematical or scientific analysis and its application of its methods and techniques used in making decisions. Operations research is also used to optimize the utility of limited resources. The objective is to select the best alternative; that is, the one leading to the best result. [2]

Operations research includes using mathematical programming models, linear programming models, integer programming models, and binary integer programming model (bi-integer programming models) to optimize the solution. We will define those models, introduce various types of application examples and explain the variables and the solutions. Then we will look at different solution methods for LP, IP, and BIP. Lastly, we will briefly look at the traveling salesman problem.

- **Acknowledgement**

I would like to give acknowledgements to my advisor, Dr. Tim Redl, and both of my committee members, Dr. Carol Vobach and Dr. Yunwei Cui for all of the help that they have provided me with. I also like to thank my professor, Dr. Xie, and classmates in my senior project class for giving me many inputs and opinions to help me improve my project and presentation.

When I arrived to UHD, I have already completed 144 credit hours of pure math at UH Main campus. Upon transferring from the main campus to the down town campus, I had to change my major from pure math to applied math. At that time, I felt like I already knew it all as if there was not a field left in the undergraduate mathematics for me to cover. Little that I knew, I was so wrong. After taking classes from Dr. Redl and Dr. Vobach, I discovered the fun and my passion for the applied mathematics field. I grew a great interest in the field. Then I decided that this is what my project will be about.

This change had also made me realize so much about the power of applied mathematics. I realized that this is the best tool that I can apply my knowledge to the real world.

But those are not the only reasons for me to choose this topic. Another reason is Dr. Tim Redl. I consider him as my mentor. Without him, probably I would not be able to enjoy what I study today. After learning so much from him, I wanted to end my student career with something that involves him in it.

- **Introduction**

Suppose we have a manufacturer of two different types of games to sell to department stores. Each type of game has a different amount of labor required to be a finished product, and each brings a different amount of profit. Is it possible to come up with an optimal number of games of each type to produce to maximize profit? If so, is it possible to come up with the optimal model using mathematics? The answer is yes. By using operations research to determine the solutions for such problems, we can determine an optimal solution for each specific problem.

But in order to do so, we need to look at the situation as a decision-making problem whose solution requires answering three questions:

1. What are the decision alternatives?
2. Under what restrictions is the decision made?
3. What is an appropriate objective criterion for evaluating the alternatives? [2]

Once we have a new perspective and understanding for the problem, now we look at the definition of programming models.

- **Definitions**

Within operations research, there are different types of programming models which one must choose, based on the nature of the specific problem.

- Mathematical programming model

Generally, *a mathematical program* is an optimization problem in which the objective and constraints are given as mathematical functions and functional relationships. Mathematical programs with *n* variables and *m* constraints have the form of

Maximize or minimize: $z = f(x_1, x_2, \ldots, x_n)$

Subject to: $g_1(x_1, x_2, \ldots, x_n)$	$\leq$ or $=$ or $\geq$	b_1
$g_2(x_1, x_2, \ldots, x_n)$	$\leq$ or $=$ or $\geq$	b_2
$g_3(x_1, x_2, \ldots, x_n)$	$\leq$ or $=$ or $\geq$	b_3
………………..	$\leq$ or $=$ or $\geq$	…..
$g_m(x_1, x_2, \ldots, x_n)$	$\leq$ or $=$ or $\geq$	b_m

and $x_{1\ldots\ldots}\, x_m$ are non-negative

Each of the m constraint relationships involves one of the three signs $\leq, =, \geq$. [2]

There are three types of programming models within the realm of mathematical programming: *linear programming models*, *integer programming models*, and *binary-integer programming models*.

- Linear programming (LP) model

    - A mathematical program is a *linear programming model*, if $f(x_1, x_2, \ldots, x_n)$ and each $g_i(x_1, x_2, \ldots, x_n)$ ($i = 1, 2, \ldots, m$) are linear – that is, if

$$f(x_1, x_2, \ldots, x_n) = c_1x_1 + c_2x_2 + \ldots + c_nx_n$$

and

$$g_i(x_1, x_2, \ldots, x_n) = a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n$$

where c_j and a_{ij} (i = 1,2,…,m; j = 1,2,…n) are known constants. [2]

    - Linear Programming (LP) model is a technique for optimization of a linear objective function, subject to linear equality and linear inequality constraints. Informally, linear programming determines the way to achieve the best outcome (such as a maximum profit or a minimum cost) given some list of requirements that are represented as linear equations or inequalities.[8]

- Integer programming (IP) model
    - An *integer programming model* is a linear programming model with the additional requirement that all of the values for the variables be integers.[8]

- Binary integer programming (BIP) model
    - A *binary integer programming model* (or *bi-integer programming model*) is an integer programming model in which each variable can only take on a value of 0 or 1. This may represent the selection or rejection of an option, the turning on or off of a switch, an answer of yes or no, or many other situations. [8]

To put this into perspective, we briefly look at the following problem. This particular example is an example of an integer programming problem.

- **An example of assignment of lawyers and cases (LP formulation)**

A legal firm has accepted five new cases, each of which can be handled adequately by any one of its five lawyers. Due to differences in experience and expertise, however, the lawyers would spend varying amounts of time on the cases. The firm has estimated the time requirements (in hours) as shown below: [2]

	Case 1	Case 2	Case 3	Case 4	Case 5
Lawyer 1	145	122	130	95	115
Lawyer 2	80	63	85	48	78
Lawyer 3	121	107	93	69	95
Lawyer 4	118	83	116	80	105
Lawyer 5	97	75	120	80	111

Table 1: Required hours for each lawyer to work each case

Determine an optional assignment of cases to lawyers such that each lawyer receives a different case and the total hours expended by the firm is minimized.

		D1	D2	D3	D4	D5	**Supply**
	Name	Case 1	Case 2	Case 3	Case 4	Case 5	
S1	Lawyer 1	145.00	122.00	130.00	95.00	115.00	1.00
S2	Lawyer 2	80.00	63.00	85.00	48.00	78.00	1.00
S3	Lawyer 3	121.00	107.00	93.00	69.00	95.00	1.00
S4	Lawyer 4	118.00	83.00	116.00	80.00	105.00	1.00
S5	Lawyer 5	97.00	75.00	120.00	80.00	111.00	1.00
Demand		1.00	1.00	1.00	1.00	1.00	

Table 2: Assignment model

This problem is an assignment model and a minimization problem. For the assignment model, the most appropriate model to use is a transportation model in which the supply for each lawyer and the demand for each case is 1. The reason this problem is a minimization problem is because we want to minimize the total hours required to resolve all five cases with the right assignment of a lawyer to a case. So as a result, we will end up with five lawyer/case pairs with the least total amount of hours being used to solve all five cases.

In Table 1 and Table 2, we see the number of hours each lawyer needs to work on each case.

From	To	Amt Shipped	Obj Coeff	Obj Contrib
S1: Lawyer 1	D5: Case 5	1	115.00	115.00
S2: Lawyer 2	D4: Case 4	1	48.00	48.00
S3: Lawyer 3	D3: Case 3	1	93.00	93.00
S4: Lawyer 4	D2: Case 2	1	83.00	83.00
S5: Lawyer 5	D1: Case 1	1	97.00	97.00

Table 3: Solution

The optimal assignment is shown in Table 3. Lawyer 1 is assigned to case 5 which requires 115 hours, lawyer 2 is assigned to case 4 which requires 48 hours, lawyer 3 is assigned to case 3 which requires 93 hours, lawyer 4 is assigned to case 2 which requires 83 hours, and lawyer 5 is assigned to case 1 which requires 97 hours. The total number of hours, when minimized, is 436 hours.

Linear programming model:

Minimize:

$$Z = 145x_{11} + 122x_{12} + 130x_{13} + 95x_{14} + 115x_{15} + 80x_{21} + 63x_{22} + 85x_{23} + 48x_{24} + 78x_{25} + 121x_{31} + 107x_{32} + 93x_{33} + 69x_{34} + 95x_{35} + 118x_{41} + 83x_{42} + 116x_{43} + 80x_{44} + 105x_{45} + 97x_{51} + 75x_{52} + 120x_{53} + 80x_{54} + 111x_{55}$$

such that:

1) $x_{11} + x_{21} + x_{31} + x_{41} + x_{51} = 1$
2) $x_{12} + x_{22} + x_{32} + x_{42} + x_{52} = 1$
3) $x_{13} + x_{23} + x_{33} + x_{43} + x_{53} = 1$
4) $x_{14} + x_{24} + x_{34} + x_{44} + x_{54} = 1$
5) $x_{15} + x_{25} + x_{35} + x_{45} + x_{55} = 1$
6) $x_{11} + x_{12} + x_{13} + x_{14} + x_{15} = 1$
7) $x_{21} + x_{22} + x_{23} + x_{24} + x_{25} = 1$
8) $x_{31} + x_{32} + x_{33} + x_{34} + x_{35} = 1$
9) $x_{41} + x_{42} + x_{43} + x_{44} + x_{45} = 1$

10) $x_{51}+ x_{52}+ x_{53}+ x_{54}+ x_{55} = 1$

and:

$x_{ij} \geq 0$

Even though our model is a linear programming model, due to the nature of constraints and the objective function, it will result in a binary programming model. For the linear programming model, the objective function represents the sum of hours for the lawyers to resolve five cases. As for the variables, for any x_{ij}, i is the number for the particular lawyer and j is the number for the specific case where i equals 1,2,3,4,5, and j equals 1,2,3,4,5. For example, take x_{11}. Here, $i = 1$ and $j = 1$, which stands for lawyer 1 being assigned to case 1.

Lawyer 1 requires 145 hours to resolve case 1; 122 hours to resolve case 2; 130 hours to resolve case 3; 95 hours to resolve case 4; and 115 hours to resolve case 5. Lawyer 2 requires 80 hours to resolve case 1; 63 hours to resolve case 2; 85 hours to resolve case 3; 48 hours to resolve case 4; and 78 hours to resolve case 5.Lawyer 3 requires 121 hours to resolve case 1; 107 hours to resolve case 2; 93 hours to resolve case 3; 69 hours to resolve case 4; and 95 hours to resolve case 5.Lawyer 4 requires 118 hours to resolve case 1; 83 hours to resolve case 2; 116 hours to resolve case 3; 80 hours to resolve case 4; and 105 hours to resolve case 5.Lawyer 5 requires 97 hours to resolve case 1; 75 hours to resolve case 2; 120 hours to resolve case 3; 80 hours to resolve case 4; and 111 hours to resolve case 5.

The first constraint represents the number of lawyers that can be assigned to case 1 which equals to 1. The second constraint represents the number of lawyers that can be assigned to case 2 which equals to 1. The third constraint represents the number of lawyers that can be assigned to case 3 which equals to 1. The fourth constraint represents the number of lawyers that can be assigned to case 4 which equals to 1. The fifth constraint represents the number of lawyers that can be assigned to case 5 which equals to 1. The sixth constraint represents the number of cases that can be assigned to lawyer 1 which equals to 1. The seventh constraint represents the number of cases that can be assigned to lawyer 2 which equals to 1. The eighth constraint represents the number of cases that can be assigned to lawyer 3 which equals to 1. The ninth constraint represents the number of cases that can be assigned to lawyer 4 which equals to 1. The tenth constraint represents the number of cases that can be assigned to lawyer 5 which equals to 1.

Optimal solution:

$$\mathbf{Z = 436}$$

$$x_{15} = 1$$

$$x_{24} = 1$$

$$x_{33} = 1$$

$$x_{42} = 1$$

$$x_{51} = 1$$

All other $x_{ij} = 0$

The solution obtained using a transportation model and the solution obtained using integer programming model were the same. Lawyer 1 should be assigned to case 5, represented by $x_{15} = 1$. Lawyer 2 should be assigned to case 4, represented by $x_{24} = 1$. Lawyer 3 should be assigned to case 3, represented by $x_{33} = 1$. Lawyer 4 should be assigned to case 2, represented by $x_{42} = 1$. Lawyer 5 should be assigned to case 1, represented by $x_{51} = 1$. Total required hours are 436.

From this example, it is clear that operations research can be used to solve an assignment problem.

- **Other Application Examples (LP, IP, BIP)**

The following problems are examples of applications of various mathematical programming models.

- **Pet Store (IP and LP)**

A pet store has determined that each hamster should receive at least 70 units of protein, 100 units of carbohydrates, and 20 units of fat daily. If the store carries the six types of feed shown in the table, what blend of feeds satisfies the requirements at minimum cost to the store? [2] Contents of feeds are listed in Table 4.

Feed	Protein, units/oz	Carbohydrates, units/oz	Fat, units/oz	Cost, cents/oz
A	20	50	4	2
B	30	30	9	3
C	40	20	11	5
D	40	25	10	6
E	45	50	9	8
F	30	20	10	8

Table 4: List of each feed with their contents

Now I will take this word problem and convert it to one of the mathematical programming models introduced previously.

<u>Integer programming formulation:</u>

Minimize: $Z = 2x_1 + 3x_2 + 5x_3 + 6x_4 + 8x_5 + 8x_6$

such that:

1) $20x_1 + 30x_2 + 40x_3 + 40x_4 + 45x_5 + 30x_6 \geq 70$
2) $50x_1 + 30x_2 + 20x_3 + 25x_4 + 50x_5 + 20x_6 \geq 100$
3) $4x_1 + 9x_2 + 11x_3 + 10x_4 + 9x_5 + 10x_6 \geq 20$

and : $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0, x_6 \geq 0$

$x_1, x_2, x_3, x_4, x_5, x_6$ **are integers**

or

Linear programming formulation:

Minimize: $Z = 2x_1 + 3x_2 + 5x_3 + 6x_4 + 8x_5 + 8x_6$

such that:

1) $20x_1 + 30x_2 + 40x_3 + 40x_4 + 45x_5 + 30x_6 \geq 70$
2) $50x_1 + 30x_2 + 20x_3 + 25x_4 + 50x_5 + 20x_6 \geq 100$
3) $4x_1 + 9x_2 + 11x_3 + 10x_4 + 9x_5 + 10x_6 \geq 20$

and : $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0, x_6 \geq 0$

We can apply either LP or IP model for this problem. The reason for this is because the unit of those feeds can be broken into smaller portions, which will result in fractions. If the whole unit was to be used at all times, it will be an integer programming model. They resulted in different optimal solutions just because of the range of the optimal solution each method allowed. Considering both scenarios, a linear programming model came up with a better optimal solution.

It is a minimization problem because we are trying to determine what is the least amount of each feed to compile a blend of feeds that satisfies the given requirements. The objective function represents the portion of each feed that makes up the blend with x_1 is feed A, x_2 is feed B, x_3 is feed C, x_4 is feed D, x_5 is feed E, x_6 is feed F.

For the integer programming model, there is the integer constraint, which is the difference between LP and IP. First constraint is for the amount of protein in each feed which needs to be at least 70 ounces put together. Second constraint is for the amount of carbohydrates in each feed which needs to be at least 100 ounces put together. The last constraint is for the amount of fat in each feed which needs to be at least 20 ounces when combined. For the non negative constraints, all feeds must be more than 0 since they cannot consist of negative amount.

For the linear programming model, we can have a portion of a feed in a blend. First constraint is not necessary since all feeds can be the fractional amount. All three constraints are the same as the ones from the integer programming model for the same reasons given above. Non negative constraints are for all feeds are greater or equal to 0 since they cannot contain negative amounts.

<u>Solution:</u>

- **For LP:**

$$\mathbf{Z = 7.27}$$
$$\boldsymbol{x_1 = 0.91}$$
$$\boldsymbol{x_2 = 1.82}$$

The solution using the linear programming model is 7.27 cents. The blend of feeds is made up with 0.91 ounces of feed A and 1.82 ounces of feed B.

- **For IP:**

$$\mathbf{Z = 8}$$
$$\boldsymbol{x_1 = 1}$$
$$\boldsymbol{x_2 = 2}$$

The solution of this problem using the integer programming model is 8 cents. The blend of feeds is made up with 1 ounce of feed A, and 2 ounces of feed B.

Considering both outcome, the solution using the integer programming model is more appropriate because we cannot have a fraction of a cent.

The solution was verified by TORA.

- **Manufacturing Firm (IP)**

A local manufacturing firm produces four different metal products, each of which must be machined, polished, and assembled. The specific time requirements (in hours) for each product are listed in Table 5.

	Machining, hours	Polishing, hours	Assembling, hours
Product I	3	1	2
Product II	2	1	1
Product III	2	2	2
Product IV	4	3	1

Table 5: List of how many hours each product requires for each process

The firm has available to it on a weekly basis 480 hours of machine time, 400 hours of polishing time, and 400 hours of assembly time. The unit profits on the products are \$6, \$4, \$6, and \$8, respectively. The firm has a contract with a distributor to provide 50 units of product I and 100 units of any combination of products II and III each week. Through other customers, the firm can sell each week as many units of products I, II, and III as it can produce, but only a maximum of 25 units of product IV. How many units of each product should the firm manufacture each week to meet all contractual obligations and maximize its total profit? Assume that any unfinished pieces can be completed the following week.[2]

Now we will take this word problem and convert it to one of the mathematical programming models introduced previously.

<u>Integer programming formulation:</u>

Maximize: $Z = 6x_1 + 4x_2 + 6x_3 + 8x_4$

Such that:

1) $3x_1 + 2x_2 + 2x_3 + 4x_4 \leq 480$
2) $x_1 + x_2 + 2x_3 + 3x_4 \leq 400$
3) $2x_1 + x_2 + 2x_3 + x_4 \leq 400$

4) $x_1 \geq 50$

5) $x_2 + x_3 \geq 100$

6) $x_4 \leq 25$

and: $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0$

x_1, x_2, x_3, x_4 **are integers**

We chose an integer programming model for solving this problem. All variables must be integers because a fraction of a product cannot be produced and the numbers of hours need to be maximized. That is the reason why this formulation must be in integers. So even though a linear programming can be used, an integer programming model is more suitable for solving this problem.

The variable x_1 represents the product I, which has the profit of \$6, x_2 represents the product II, which has the profit of \$4, x_3 represents the product III, which has the profit of \$6, and x_4 represents the product IV, which has the profit of \$8. The objective function is to maximize the total profit.

First constraint is for the hours for machining. 3 hours for product I, 2 hours for product II, 2 hours for product III, and 4 hours for product IV. The constraint is not to exceed 480 hours when combined. Second constraint is for the hours for polishing. 1 hour for product I, 1 hour for product II, 2 hours product III, and 3 hours for product IV. It is not to exceed 400 hours when combined. Third constraint is for the hours for assembling. 2 hours for product I, 1 hour for product II, 2 hours for product III, and 1 hour for product IV. It is not to exceed 400 hours when combined. Fourth constraint is for the quantity of product I. The firm has a contract with a distributor to provide at least 50 units of product I each week. Fifth constraint is for the quantity of combination of product II and III. The firm has a contract with a distributor to provide at least 100 units of any combination of products II and III each week. Sixth constraint is for the quantity of product IV. It can produce, but only a maximum of 25 units of product IV and no more than that each week. All products cannot be negative because it is impossible to produce the negative quantity of any products.

Solution:

$$\mathbf{Z = \$1250}$$

$$x_1 = 50$$

$$x_3 = 145$$

$$x_4 = 10$$

The optimal solution consists of the objective value of \$ 1250 with 50 units of product I, 145 units of product III, and 10 units of product IV.

The solution was verified by TORA.

- **Caterer (LP)**

A caterer must prepare from five fruit drinks in stock 500 gal of a punch containing at least 20 percent orange juice, 10 percent grape juice, 5 percent cranberry juice. If inventory data are as shown below, how much of each fruit drink should the caterer use to obtain the require composition at minimum total cost? [2] Information regarding each drinks are listed in Table 6.

	Orange Juice, %	Grapefruit Juice, %	Cranberry Juice, %	Supply, gal.	Cost, $/gal.
Drink A	40	40	0	200	1.50
Drink B	5	10	20	400	0.75
Drink C	100	0	0	100	2.00
Drink D	0	100	0	50	1.75
Drink E	0	0	0	800	0.25

Table 6: Contents of each drink

Now we will take this word problem and convert it to one of the mathematical programming models introduced previously.

<u>Linear programming formulation:</u>

Minimize: $\mathbf{Z = 1.50x_1 + 0.75x_2 + 2.00x_3 + 1.75x_4 + 0.25x_5}$

Such that:

1) $\mathbf{0.40\, x_1 + 0.05\, x_2 + x_3 \geq 0.20}$
2) $\mathbf{0.40\, x_1 + 0.10\, x_2 + x_4 \geq 0.10}$
3) $\mathbf{0.20\, x_2 \geq 0.05}$
4) $\mathbf{x_1 + x_2 + x_3 + x_4 + x_5 = 500}$
5) $\mathbf{x_1 \leq 200}$
6) $\mathbf{x_2 \leq 400}$
7) $\mathbf{x_3 \leq 100}$
8) $\mathbf{x_4 \leq 50}$
9) $\mathbf{x_5 \leq 800}$

and: $\mathbf{x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0}$

We chose a linear programming model. The reason for this being the optimal solution will be in dollars, which will have decimals. It is also necessary to get an accurate minimum cost broken down to cents. Also note that these variables can be fractions.

This problem is a minimization problem because we are trying to find the combination of juice that satisfies the given requirement at the minimum cost. The objective function represents the cost of the sum of all juice with x_1 is drink A, x_2 is drink B, x_3 is drink C, x_4 is drink D, and x_5 is drink E.

x_1 represents the number of gallons of drink A in a punch. X_2 represents the number of gallons of drink B in a punch. X_3 represents the number of gallons of drink B in a punch. X_3 represents the number of gallons of drink C in a punch. X_4 represents the number of gallons of drink D in a punch. X_5 represents the number of gallons of drink E in a punch.

First constraint represents the percentage of orange juice contained in each drink A, B, C, D, E, which needs to be more than 20 percent. Second constraint represents the percentage of grapefruit juice contained in each drink A, B, C, D, E, which needs to be more than 10 percent. Third constraint represents the percentage of cranberry juice contained in each drink A, B, C, D, E, which needs to be more than 5 percent. Fourth constraint represents the amount of product made with drink A, B, C, D, E, which needs to be 500 gallons. Since all drinks cannot have a negative quantity, all variables need to be greater or equal to zero. Also, all drinks have the limited amount that can be used. Fifth constraint is for x_1. It must be less than or equal to 200 gallons. Sixth constraint is for x_2. It must be less than or equal to 400 gallons. Seventh constraint is for x_3. It must be less than or equal to 100 gallons. Eighth constraint is for x_5. It must be less than or equal to 50 gallons. Ninth constraint is for x_5. It must be less than or equal to 800 gallons.

<u>Solution:</u>

$$\mathbf{Z = \$125.56}$$

x_1 **= 0.19 gallons**

x_2 **= 0.25 gallons**

x_3 **= 0.11 gallons**

x_5 **= 499.45 gallons**

The optimal solution is \$125.56 for the mix of drinks. The mix of drinks consist of 0.19 gallons of drink A, 0.25 gallons of drink B, 0.11 gallons of drink C, and 499.45 gallons of drink E.

The solution was verified by TORA.

- **Sites Development (BIP)**

A town has budgeted $250,000 for the development of new rubbish disposal areas. Seven sites are available, whose projected capacities and development costs are given below. Which sites should the town develop? [2] Capacities (tons per week) and costs (in thousands) for each cities are in Table 7.

Site	A	B	C	D	E	F	G
Capacity, tons/wk	20	17	15	15	10	8	5
Cost, $1000	145	92	70	70	84	14	47

Table 7: List of the capacity(tons/wk) and the cost ($1000/unit) for each site

Now we will take this word problem and convert it to one of the mathematical programming models introduced previously. Because of the nature of this problem, we will model this problem under different constraints to see the effects of a constraint.

For the cost optimization:

1.

Maximize: $Z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7$

such that:

$$145x_1 + 92x_2 + 70x_3 + 70x_4 + 84x_5 + 14x_6 + 47x_7 \leq 250$$

$$x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6,\ x_7, \text{ all binary}$$

2.

Minimize: $Z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7$

such that:

$$145x_1 + 92x_2 + 70x_3 + 70x_4 + 84x_5 + 14x_6 + 47x_7 \leq 250$$

$$x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6,\ x_7, \text{ all binary}$$

<u>For the capacity optimization – subject to the different maximum capacities:</u>

3.

Maximize: $\mathbf{Z} = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7$

such that:

$$145x_1 + 92x_2 + 70x_3 + 70x_4 + 84x_5 + 14x_6 + 47x_7 \leq 25$$

$x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6,\ x_7$, **all binary**

4.

Maximize: $\mathbf{Z} = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7$

such that:

$$145x_1 + 92x_2 + 70x_3 + 70x_4 + 84x_5 + 14x_6 + 47x_7 \leq 35$$

$x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6,\ x_7$, **all binary**

5.

Minimize: $\mathbf{Z} = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7$

such that:

$$145x_1 + 92x_2 + 70x_3 + 70x_4 + 84x_5 + 14x_6 + 47x_7 \leq 35$$

$x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6,\ x_7$, **all binary**

6.

Maximize: $\mathbf{Z} = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7$

such that:

$$145x_1 + 92x_2 + 70x_3 + 70x_4 + 84x_5 + 14x_6 + 47x_7 \leq 45$$

$x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6,\ x_7$, **all binary**

We chose a binary programming model. The reason for this is because you are trying to decide which area to develop based of the cost/capacity given by the table. It is we do it or not. Also, since both the cost and the capacity has the limit of the sum, that restricts the constraint.

This problem is a maximization problem for scenarios 1, 3, 4, and 6 because we have to maximize the number of sites to develop for those scenarios. For scenarios 2 and 5, the problem is a minimization problem because we are trying to minimize the number of the sites to develop under the condition. The objective function represents the total cost of the sites and the total amount of physical capacity of sites for all scenarios. The variable x_1 is site A, x_2 is site B, x_3 is site C, x_4 is site D, x_5 is site E, x_6 is site F, and x_7 is site G. All variables are binary.

A constraint for the cost represents the sum of the cost of the sites to develop, which must stay under $250,000. Site A costs $145,000 to develop, site B costs $92,000 to develop, site C costs $70,000 to develop, site D costs $70,000 to develop, site E costs $84,000 to develop, site F costs $14,000 to develop, site G costs $47,000 to develop.

A constraint for the capacity represents the sum of the capacity of the sites to develop, which must stay under 35 tons per week in this case. Site A requires 20 tons per week to develop, site B requires 17 tons per week to develop, site C requires 15 tons per week to develop, site D requires 15 tons per week to develop, site E requires 10 tons per week to develop, site F requires 8 tons per week to develop, and site G requires 5 tons per week to develop.

We also worked on both maximization and minimization on all cases to see what kind of difference it will make. Also, as for the capacity model, we have tested with the limit being 35, 25 and 45 to see the change in the result.

Suppose we want to look at scenario 2. It is to minimize the total cost of the sites if developed. The constraint combines the cost of sites developed, which must stay under 250. All variables are binary.

Solutions of all the above scenarios along with explanations are as follows.

Solutions:

1.

Z = 4

$\boldsymbol{x_3 = 1}$

$\boldsymbol{x_4 = 1}$

$\boldsymbol{x_6 = 1}$

$\boldsymbol{x_7 = 1}$

If we maximize the number of the sites to be developed under the budget of $250,000, 4 sites are to be developed, which are site C, site D, site F, and site G.

2.

Z = 0

All variables are 0

If we minimize the number of the sites to be developed under the budget of $250,000, none of the sites will be developed. This is very clear considering we are trying to minimize the cost. If we build no sites, we can keep the cost at 0.

3.

Z = 3

$\boldsymbol{x_5 = 1}$

$\boldsymbol{x_6 = 1}$

$\boldsymbol{x_7 = 1}$

If we maximize the number of the sites to be developed under the capacity of 25 tons per week, 3 sites are to be developed, which are site E, site F, and site G.

4.

Z = 3

$\boldsymbol{x_5 = 1}$

$\boldsymbol{x_6 = 1}$

$\boldsymbol{x_7 = 1}$

All other $\boldsymbol{x_i = 0}$

If we maximize the number of the sites to be developed under the capacity of 35 tons per week, 3 sites are to be developed, which are site E, site F, and site G.

5.

Z = 0

All $\boldsymbol{x_i = 0}$

If we minimize the number of the sites to be developed under the capacity of 35 tons per week, no sites are to be developed. If we develop no sites, we keep the capacity at the minimum value, which is 0.

6.

Z = 4

$\boldsymbol{X_3 = 1}$

$\boldsymbol{x_5 = 1}$

$\boldsymbol{x_6 = 1}$

$\boldsymbol{x_7 = 1}$

If we maximize the number of the sites to be developed under the capacity of 45 tons per week, 3 sites are to be developed, which are site C, site E, site F, and site G.

The solutions were verified by TORA.

- **Supermarket (LP)**

The manager of a supermarket meat department finds she has 200 lb of round steak, 800 lb of chuck steak, and 150 lb of pork in stock in Saturday morning, which she will use to make hamburger meat, picnic patties, and meat loaf. The demand for each of these items always exceeds the supermarket's supply. Hamburger meat must be at least 20 percent ground round and 50 percent ground chuck (by weight); picnic patties must be at least 20 percent ground pork and 50 percent ground chuck; and meat loaf must be at least 10 percent ground round, 30 percent ground pork, and 40 percent ground chuck. The remainder of each product is inexpensive non meat filler which the store has in unlimited supply. How many pounds of each product should be made if the manager desires to minimize the amount of meat that must be stored in the supermarket over Sunday? [2]

Now we will take the information from this word problem and convert it to one of the mathematical programming models introduced previously.

<u>Linear programming formulation</u>:

Maximize: $Z = x_1 + x_2 + x_3$
such that:

1) $0.2\,x_1 + 0.1\,x_3 \leq 200$
2) $0.5\,x_1 + 0.5\,x_2 + 0.4\,x_3 \leq 800$
3) $0.2\,x_2 + 0.3\,x_3 \leq 150$
4) $0.3\,x_1 + 0.3\,x_2 + 0.2\,x_3 \leq 10000000$

and: $x_1,\ x_2,\ x_3 \geq 0$
$x_1,\ x_2,\ x_3$ **are integers**

We chose a linear programming model for solving this problem. All variables must be in pounds, therefore, you must allow fractional values. The total amount of hamburger meat, picnic patties, and meat loaf in pounds is to be maximized, so this problem is a maximization problem.

The variable x_1 represents pounds of hamburger meat produced. x_2 represents pounds of picnic patties produced. x_3 represents pounds of meat loaf produced.

First constraint is for the amount of round steak in pounds. One unit of hamburger meat must contain at least 20 percent and one unit of meat loaf must contain at least 10 percent of round steak. The total amount of round steak is not to exceed 200 pounds. Second constraint is for the amount of chuck steak. One unit of hamburger meat must contain at least 50 percent, one unit of picnic pattie must contain at least 50 percent, and one unit of meat loaf must contain at least 40 percent of chuck steak. The total amount of chuck steak is not to exceed 800 pounds. Third constraint is for the amount of pork. One unit of picnic pattie must contain at least 20 percent of round steak and one unit of meat loaf must contain at least 30 percent of pork. The total amount of round steak is not to exceed 200 pounds. The total amount of pork is not to exceed 150 pounds. Fourth constraint is for non meet filler which the store has in unlimited supply. The constraint cannot be set less than or equal to the infinity, therefore, choose a value that is large enough that can work as infinity value. It is to fill the remainder of each product per unit, which is 30 percent for hamburger meat, 30 percent for picnic pattie, and 20 percent for meat loaf.

Solution:

Z = 1625

$\boldsymbol{x_1 = 937.5}$

$\boldsymbol{x_2 = 562.5}$

$\boldsymbol{x_3 = 125}$

The solution is x_1 equals 937.5, x_2 equals 562.5, x_3 equals 125, and Z equals 1625. This means that 937.5 pounds of hamburger meat will be produced, 563.5 pounds picnic patties will be produced, 125 of meat loaf will be produced, and the total amount of products made is 1625 pounds.

The solution was verified by TORA.

- **Solution Methods**

- **Simplex Method (LP)**

The simplex algorithm, created by the American mathematician George Dantzig in 1947, is a popular algorithm for numerical solution of the linear programming problem.

The simplex method is a matrix procedure for solving linear programs in the standard form

$$\text{optimize: } z = C^T X$$

$$\text{subject to: } AX = B$$

$$\text{with: } X \geq 0$$

where $B \geq 0$ and a basic feasible solution X_0 is known. Starting with X_0, the method locates successively other basic feasible solutions having better values of the objective, until the optimal solution is obtained. For minimization programs in which C_0 designates the objective function coefficient vector associated with the variables in X_0.

Assume that the problem is a minimization problem, the optimality condition calls for selecting the entering variable as the non-basic variable with the most positive objective coefficient in the objective equation. This follows because max z is equivalent to min ($-z$). As for the feasibility condition for selecting the leaving variable, the rule remains unchanged. [8] For a maximization problem, all logics are reversed.

The steps of the simplex method are:

1. Determine a starting basic solution within the feasible region.
2. Select an entering variable using coefficients of the objective function. Stop if there is no entering variable, the last solution is optimal. Else go to step 3.
3. Select a leaving variable using coefficients of the objective function.

4. Determine the new basic variable by using the appropriate Gauss Jordan elimination method. If necessary, go to step 2.

- Feasibility condition:

  If all of the variables in a particular solution hold the values which satisfy all constraints and non-negative constraints, a solution is said to be a feasible solution.

- Optimality condition:

  Within all feasible solutions, if a solution returns the least z value for a minimization problem and the greatest z value for a maximization problem, a feasible solution is said to be the optimal solution.

Also here is a brief summary of Gauss-Jordan Row Operation steps.

- Gauss-Jordan row operations:
  1. Pivot row
     - Replace the leaving variable in the Basic column with the entering variable.
     - New pivot row = Current pivot row / pivot element
  2. All other rows, including z
     - New row = (Current row) – (pivot column coefficient) x (New pivot row)

Once the z-row meets the optimality condition, an optimal solution has been obtained. [5]

- **Branch and Bound method(IP, BIP)**

Branch and Bound is the most common solution method used for integer programming models.

By first, we start from a linear relaxation the solution of the model.

A linear relaxation to the solution of any integer program may be obtained by ignoring the integer requirement. If the optimal solution to the linear program happens to be integral, then this solution is also the optimal solution to the original integer program. Otherwise, one may round the components of the first approximation to the nearest integers.

Then we will move onto branching step. If the first approximation contains a variable that is not integer, say x_j, then $i_1 < x_j < i_2$ where i_1 and i_2 are consecutive nonnegative integers.

Two new integer solutionss are then created by augmenting the original integer program with either the constraint $x_j \leq i_1$ or the constraint $x_j \leq i_2$. This process, called a branching, has the effect of shrinking the feasible region in a way that eliminates the solution obtained by a linear relaxation from further consideration the current non-integral solution for x_j but still preserves all possible integral solutions to the original problem.[3]

Then finally we will obtain the solution by taking the final process of this method, bounding.

Assume that the objective function is to be maximized. Branching continues until an integral solution is obtained. The value of the objective variable for this first integral solution becomes a lower bound for the problem, and all solutions yield values of the objective variable smaller than the lower bound are discarded.

Branching continues from those solutions having non-integral linear relaxation solutions that give values of the objective variable greater than the lower bound. If a new integral solution is found containing a value of the objective variable less than the current lower bound, then this value of the objective function becomes the new lower bound. The branching process continues until there are no more non-integral linear relaxation solutions remaining under consideration. At this point, the current lower-bound solution is the optimal solution to the original integer programming model. If the objective variable is to be maximized, the procedure remains the same, except that upper bounds are used. Thus, the value of the first integral solution becomes an upper bound for the problem, and solutions are eliminated when their z-values are greater than the current upper bound. Figure 1 below illustrates the general flow.

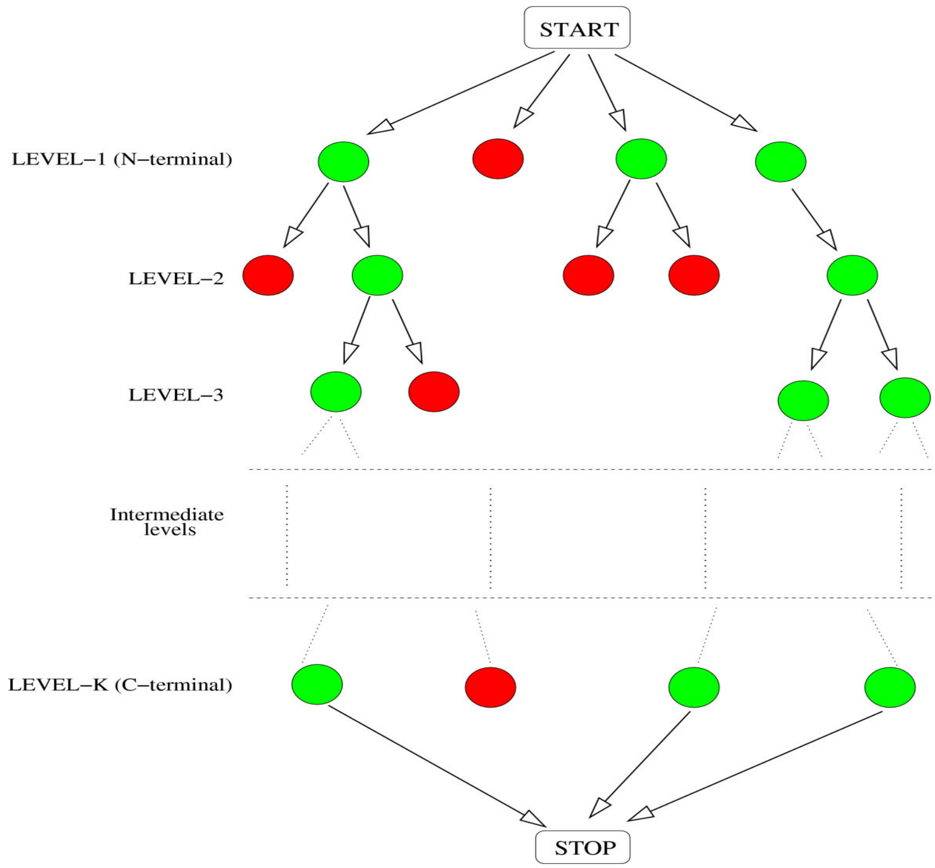

Figure 1: Branch and Bound diagram

Green nodes are the solutions that contain the integer values and red nodes are the bounded solution, the solutions that contain non-integer values.

To illustrate this description of the branch and bound method, we will look back to one of the applications that I have solved using the integer programming model.

Branch and Bound starts with a linear relaxation, solving the model disregarding the requirement that all variables are to be integers. Figure 2 shows the initial solution obtained by a linear relaxation step.

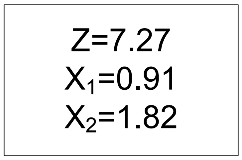

Figure 2: Initial solution

Then choose a variable that is not an integer and start the first step of branching. In this example, both x_1 and x_2 are not integers. So we choose x_1.

Then we pick two integers i_1 and i_2 where $i_1 < x_1$, and $i_2 > x_1$. In this case, $i_1 = 0$, and $i_2 = 1$.

Now, we use those integers to replace x_1, and solve the problem using those i's.

As a result of a first branching step, we have two solutions as shown in Figure 3.

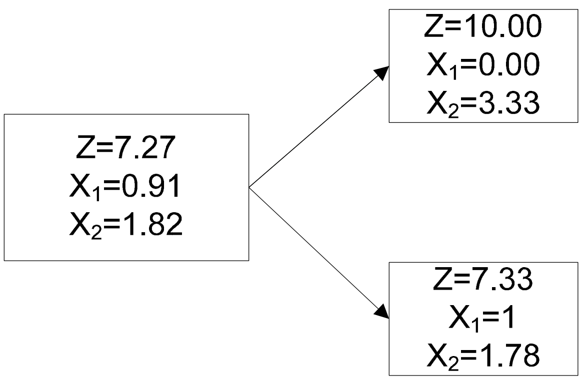

Figure 3: First branching step

So now, we have set x_1 to an integer, we start branching on x_2 since x_2 is the only variable left with non integer value. The process of this branching is identical to the process for x_1.

Then we choose two integers i_1 and i_2, where $i_1 < x_2$, and $i_2 > x_2$. In this case, $i_1 = 3$, and $i_2 = 4$ for the node above, and $i_1 = 1$, and $i_2 = 2$ for the node in Figure 2.

In Figure 4 and Figure 5, we have illustrations of the second branching steps from each of the first round solutions.

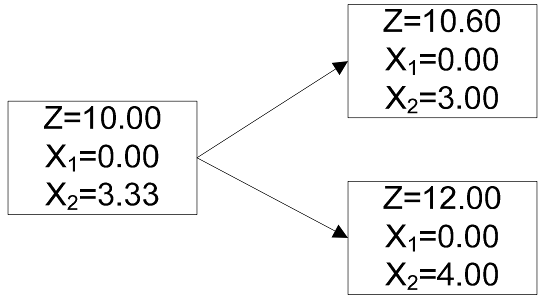

Figure 4: Second branching step of a first round solution 1 with a bounded solution

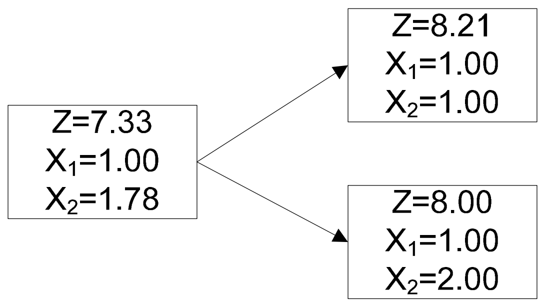

Figure 5: Second branching step of a first round solution 2 with a bounded solution

Now that all x's are in integers, we can start bounding. For those solutions that have non-integer Z value with both x_1 and x_2 being integers, they are to be ignored, which means they are the bound.

Then we move onto solutions with Z, x_1, and x_2 that are integers. Since this was a minimization problem, we choose the solution with the least Z value.

So the solution for this problem is:

$$Z=8.00$$
$$x_1=1.00$$
$$x_2=2.00$$

The solution was obtained by branch and bound method.

- **Traveling Salesman Problem**

Another famous application type of a mathematical programming model is called the Traveling Salesman Problem (TSP).

TSP is not very difficult to understand its concept: given a finite numbers of nodes, or "cities" along with the cost of travel on each arc, find the cheapest way of visiting all of the cities and returning to your starting point. The travel costs are symmetric in the sense that traveling from city X to city Y costs just as much as traveling from Y to X. Once the cycle is completed, satisfying all requirements, it is also known as Hamiltonian cycle. Hamiltonian cycle is a graph cycle that visits each node exactly once. [1]

Because of its nature, even though TSP is not hard to understand it concept, it is very difficult to solve. It contains *(n-1)!/2* numbers of solutions. As the number of nodes, *n*, increases, then the number of alternatives for the optimal solution increases dramatically.

For some solution methods, this can cause a problem.

Solution methods for TSP include:

- Brute-force search
  - Look at every possible combinations of paths and determine which combination returns the least total cost for the travel
- Branch and Bound algorithm (for 40 to 60 cities)
- Progressive improvement algorithm (works well up to 200 cities)
- Implementation of branch and bound (this approach holds the current record of 85900 cities)
- Dynamic programming solution (only in *2^n* space)

Following the theoretical studies of J.B. Robinson and H.W. Kuhn in the late 1940s and the early 1950s, G.B. Dantzig, R. Fulkerson, and S.M. Johnson demonstrated in 1954 that large instances of the TSP could be solved by linear programming. Their approach remains the only known tool for solving TSP instances with more than several hundred cities; over the years, it has evolved further through the work of M. Grőtschel, S. Hong,

M. Junger, P. Miliotis, D. Naddef, M. Padberg, W.R. Pulleyblank, G. Reinelt, G. Rinaldi and others. [1]

In Figure 6, we have an illustration of what TSP may look like in some situation.

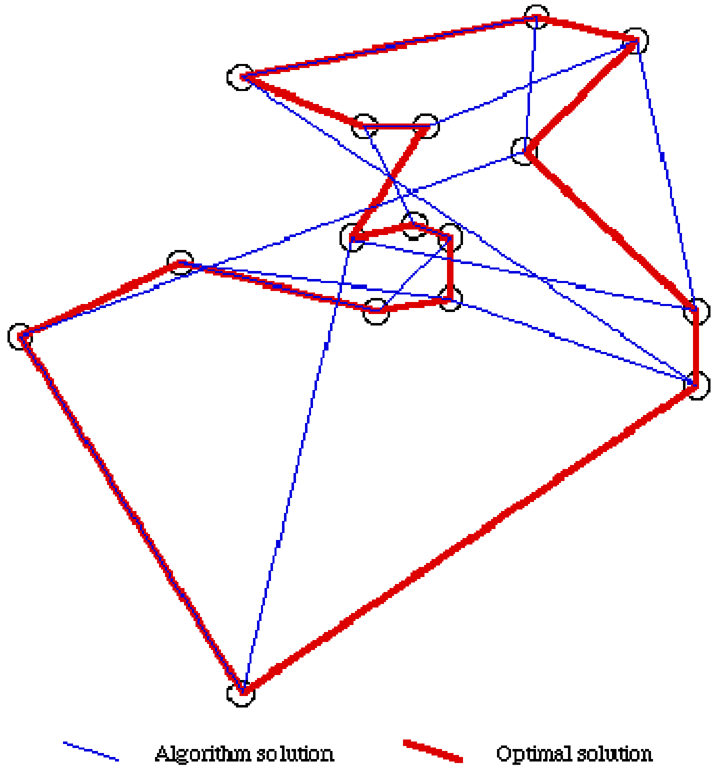

Figure 6: Traveling Salesman Problem example diagram

Here, we have 16 nodes (cities) and the blue paths represent all paths between every node. Using those paths, we must find a way to visit every node and return to the initial node with the least total cost.

The red path is the optimal solution, the path with the least cost going through all nodes and visit each only once, and returning to the initial node to create a cycle.

- **Conclusion / Future Work**

Future work for my project will include the following:

- Analysis of other solution methods for IP and their applications
- Analysis of solution methods for TSP
- Analysis of various applications of TSP in the real world
- Analysis of Dynamic Programming Model and its applications

Through this project, I have learned the structure of Operations Research more in depth and gained a deeper understanding of each Linear, Integer and Binary Programming models and their characteristics. Also, by conducting various analyses, using real examples, clarified the similarities and differences in a way that I understand them a lot better. I can confidently say that this project has provided me with a great opportunity to learn about this particular topic to prepare myself a lot better to start my career in this field.

Even I have learned so much academically, this also was a life lesson for me. Upon completion of this project, I felt something that I have never felt before, a sense of an accomplishment. Having to go through so many hardships to come this far, it seems as if those obstacles were there to teach me what it really means to try, and put in a real effort to achieve something that I can be proud of.

This project has grown into something that I feel like it is my own child in course of past few months. I am very proud of what I have done here, and I would not have it in any other way.

- **References**

1. Applegate, David and Robert Bixby and Vasek Chvatal and William Cook. On The Solution of Traveling Salesman Problem. Documenta Mathematica, Extra Volume. ICM 1998 645-656

2. Bronson, Richard, Ph.D. and Govindasami Naadimuthu, Ph.D., P.E.. Schaum's outline of Theory and Problems of Operations Research. McGraw-Hill, 1997

3. Bradley, Stephen P, and Arnold C. Hax, and Thomas L. Magnanti. Applied Mathematical Programming. Addison-Wesley. Boston, MA. 1997.

4. Clausen, Jens. Branch and Bound Algorithms – Principles andExamples. Department of Computer Science, University of Copenhagan, Universitetsparken1, DK-2100 Copenhagen, Denmark. 1999

5. Ferguson, S, Thomas. Linear Programming. University of California. 2009

6. Poldner, Michael and Herber Kuchen. Algorithmic Skeletons for Branch & Bound. Springer Berlin Heidelberg, 2008

7. Redl, Tim. Class Notes, Spring 2009 STAT4311 Operations Research. University of Houston – Downtown

8. Taha, Hamdy. Operations Research: An Introduction 8th edition. Prentice Hall, 2007

9. "Linear Programming." Wikipedia. <http://en.wikipedia.org/wiki/Linear_programming>

10. "Traveling Salesman Problem." Wikipedia. <http://en.wikipedia.org/wiki/Traveling_salesman_problem>