CPCS 500
November 4, 2013
Scribe: Fatemeh Dorri

1 Overview

In this lecture we start to introduce a set of problems called linear programming problems. Simplex algorithm is also described for solving linear programming problems. Then we continue to focus on a particular class of linear programming algorithms called Network Flows which have been more investigated in computer science and specialized algorithm have been developed for solving them.

2 Linear Programming

For the first time, you might think that linear programming is a language for programming like C, Java or Python, nut it is not. It is a general framework for finding the best outcome based on a list of requirements represented as linear relationships. To be more accurate, it a tool for solving/optimizing a linear objective function subject to a bunch of linear equality/inequality constraints on the variables.

The linear programming has a solution unless

1- It is infeasible which means there is no point satisfying all the constraints. For example:

   Max x_1
   Subject to:
      $x_1 \geq 2$
      $x_1 \leq 1$

2- The feasible set is unbounded. For example

   Max (x_1+x_2)
   Subject to:
      $x_1, x_2 \geq 0$

Here, we bring two examples to clarify linear programming problems.

2.1 Linear Programming Example - Chocolate Maker

A chocolate maker has two products. The first product is boxes of regular chocolates which their demand are less than 200 boxes per day and the profit for each regular chocolate box is one dollar. The second product is a fancy chocolate box with less than 300 boxes demand per day and the profit of making such boxes is six dollars each. However, the company is not a big one and can produce up to 400 boxes per day. The question is how many boxes of each the company should make to have the maximum profit?

Let's define the following parameters:

x_1: # of boxes of type 1 produced per day
x_2: # of boxes of type 2 produced per day

Objective function: max (x_1+6x_2)
Subject to: $x_1 \leq 200$
$x_2 \leq 300$
$x_1+x_2 \leq 400$
$x_1, x_2 \geq 0$

As you can see all the constraints and the objective function are linear.

A linear equation in the two-dimensional (2D) space (2 variables) is a line in the plane, and a linear inequality specifies the region on one side of the line. Thus the set of five constraints in this linear program make the feasible set, the intersection of five half-spaces. It is a convex polygon, shown in Figure 1- (a).

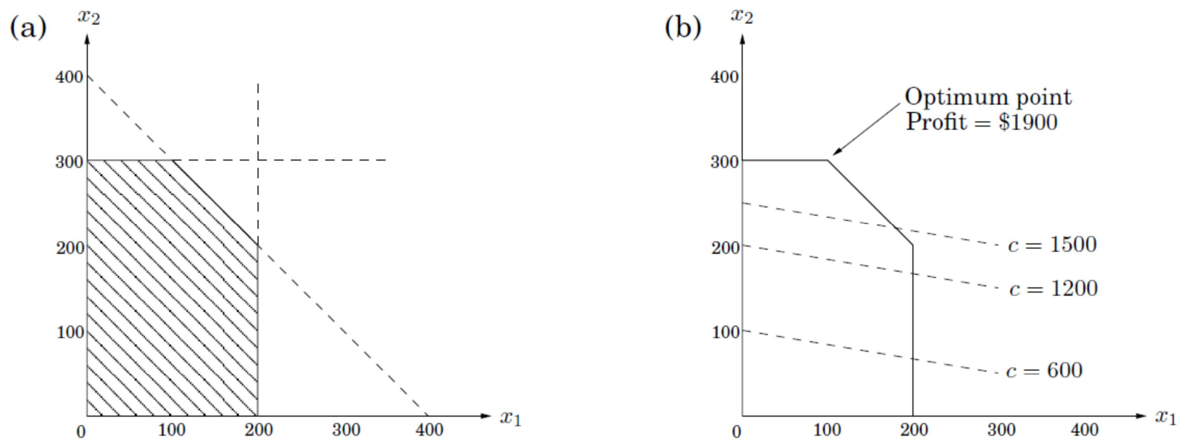

Figure 1: The feasible region of chocolate maker linear program and the optimal point

A linear equation in the two-dimensional (2D) space (2 variables) is a line in the plane, and a linear inequality specifies the region on one side of the line. Thus the set of five constraints in this linear program make the feasible set, the intersection of five half-spaces. It is a convex polygon, shown in Figure 1-a above.

We want to find the point in this region at which the profit (the objective function) is maximized. The points with a profit of dollars lie on the line, which has a slope of -1/6 and is shown in Figure 1-b for selected values of c . As increases, this "profit line" moves parallel to itself, up and to the right. Since the goal is to maximize, we must move c the line as far up as possible, while still in the feasible region. The optimum solution will be the very last feasible point that the profit line sees and must therefore be a vertex of the polygon, as shown in the Figure 1-b.

If the slope of the objective function line were different, then could touch one of the polygon entire edge rather than a single vertex. Therefore, the optimum solution would not be only one, but there would certainly be at least one optimum vertex

2.2 Linear Programming Example – Bandwidth Allocation

In a network, shown Figure 2, we need to connect every two users with at least 2 units of bandwidth. The numbers shown on the edges of the graph is the capacity. Three dollars will be paid for each bandwidth units assigned between A and B, and two and four dollars for connection between B & C and A & C respectively. The question is how to route these connections to maximize route revenue?

Between every two nodes, there exist two paths, a long path and a short one. Let's define x_{AB} the short path bandwidth allocation between A and B, and x'_{AB} the bandwidth allocation for the long path between the same nodes. In a similar way, we define x_{AC}, x'_{AC}, x_{BC} and x'_{BC} for paths between A & C and B & C. We can write it down as

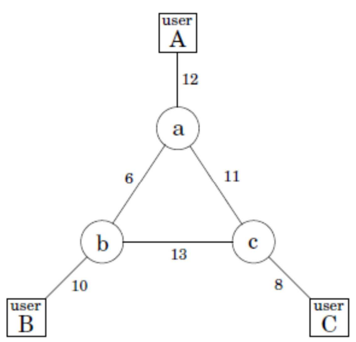

Figure 2: communication network between three users

$$\text{Max } 3x_{AB} + 3x'_{AB} + 2x_{AC} + 2x'_{AC} + 4x_{BC} + 4x'_{BC}$$
$$x_{AB} + x'_{AB} + x_{BC} + x'_{BC} \le 10$$
$$x_{AB} + x'_{AB} + x_{AC} + x'_{AC} \le 12$$
$$x_{AC} + x'_{AC} + x_{BC} + x'_{BC} \le 8$$
$$x_{AB} + x'_{AC} + x'_{BC} \le 6$$
$$x'_{AB} + x'_{AC} + x_{BC} \le 13$$
$$x'_{AB} + x_{AC} + x'_{BC} \le 11$$
$$x_{AB} + x'_{AB} \ge 2$$
$$x_{AC} + x'_{AC} \ge 2$$
$$x_{BC} + x'_{BC} \ge 2$$
$$x_{AB}, x'_{AB}, x_{AC}, x'_{AC}, x_{BC}, x'_{BC} \ge 0$$

The solution is not obvious and cannot be computed based on try and errors. Therefore, even for small problems of this type, we need to have an algorithm to compute the optimal solution.

2.2 Simplex Algorithm

There are different algorithms for solving linear programming problems. The most famous of these, the simplex method, was proposed by George Dantzig in 1947. The idea is pretty simple and performs well in practice.

The idea is to move from the current point in the feasible set to one of the neighbors that show improvement in terms of the objective function. Therefore, it can be described as below:

Start from a vertex v of feasible set.
While there is a neighbor v' of v with better objective value, then
    Set $v = v$' and repeat again

For example, in order to solve the chocolate maker problem, we only need to follow the path in Figure 3 to reach the optimal point.

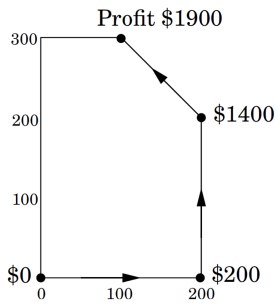

Figure 3: Simplex algorithm path for chocolate maker problem

Upon reaching a vertex that has no better neighbor, simplex declares it to be optimal and stops.

For the bandwidth allocation problem the optimal solution based on simplex algorithm will be $x_{AB} = 0$, $x'_{AB} = 7$, $x_{AC} = 0.5$, $x'_{AC} = 4.5$, $x_{BC} = 1.5$, $x'_{BC} = 1.5$, which was not easy to compute on one's try.

3 Network Flows

Flow Networks is a directed graph $G = (V, E)$ in which each edge $(u, v) \in E$ has a positive capacity $c(u, v)$. There is a source vertex $s \in V$ and a sink vertex $t \in V$.

A Flow is an assignment function f of real numbers to edge G

1- $f(u, v) \le c(u, v)$

2- $\sum_{\substack{(u,v)\in E \\ u\in V}} f(u,v) - \sum_{\substack{(v,w)\in E \\ w\in V}} f(v,w)$

$$Size(f) = \sum_{(s,v)\in E} f(s,v) - \sum_{(v,s)\in E} f(v,s)$$

The goal is to find flow with maximum size. As you can see, this problem is a linear programming question. The variables are $f(u,v)$ for $(u,v) \in E$. The objective function is max $(size(f))$. And the constraints are numbered above. As the objective function and constraints are linear combination of variables, the problem is a linear programming problem.

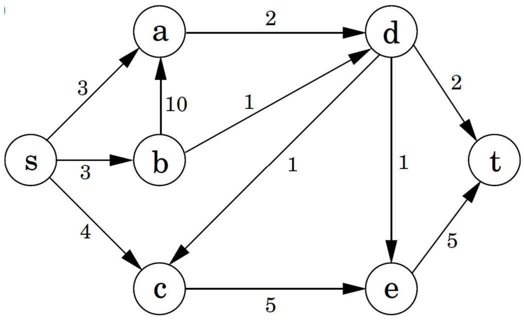

Figure 4: A Network Flow graph

Example – This is an example of network flow problems that can be solved via path augmentation suggested by Ford, Fulkerson in 1962. The idea is to start from a zero flow and choose "augmenting path" from s to t in residual network by increasing the flow on every edge of this path up to the capacity of the path. The residual network will be defined as $G^f = (V, E^f)$, where

$$E^f = \{(u,v): (u,v) \in E, f(u,v) < c(u,v) with\ c^f(u,v) = c(u,v) - f(u,v)$$
$$(u,v): (v,u) \in E, f(u,v) > 0\ with\ c^f(u,v) = f(u,v)\ \}$$

Though the algorithm gives us a solution, but the proof for showing that it is the maximum flow and the algorithm will terminates after a finite steps was not provided here and will be covered in next lectures.

Conclusion

Many problems in the world can be modeled as linear programming problems. A class of linear programming problems called Network Flows has been the focus of computer scientist so that different algorithm has been specialized for solving them in particular.

References

S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, Algorithms, McGraw-Hill 2006

http://en.wikipedia.org/wiki/Linear_programming