Lecture 1

Introduction to Combinatorial Optimization

Oct 1, 2004
Lecturer: Kamal Jain
Notes: Atri Rudra

1.1 Introduction

Combinatorial Optimization is a broad field where roughly one tries to optimize an objective function subject to certain constraints. For example, one might want to find the Minimum Spanning Tree of a graph, the shortest path between two nodes in a graph or the Maximum matching in a bipartite graph. A related question is checking if a given answer is indeed the optimal solution. For example given a Spanning Tree one would like to check quickly (where quickly means polynomial time) if the tree is indeed the Minimum Spanning Tree (MST). If the given tree is not the MST, then a certificate for this fact can be any cheaper spanning tree. However, if the spanning tree is indeed the MST one can try to enumerate all the spanning trees and show that the given tree is the cheapest. This, however is not efficient. Note that for problems in $\mathsf{NP} \cap \mathsf{coNP}$, the issue of finding a short certificate is easy. We next briefly mention a theorem (which would be covered in a later lecture) which gives a way to check the optimality of a candidate solution for the problem of Maximum flow.

1.1.1 Max Flow Min Cut theorem

In this problem we are given a graph, possibly directed, with each edge having a capacity and the objective is to find the maximum flow in the graph. For example consider the graph in Figure 1.1. A candidate solution is proposed in Figure 1.2 which is shown not to be the optimal by flow in Figure 1.3.

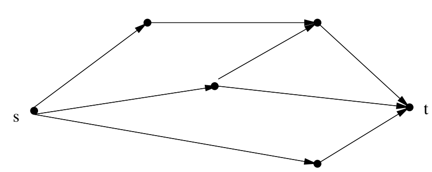

Figure 1.1: The capacities of each edge is one. The source of the flow is denoted by s and the terminal by t

The problem of showing whether a given flow is optimal could be solved if there existed a number say F such for any flow of value f, $f \leq F$. The max-flow min-cut theorem states that F exists and it is the value

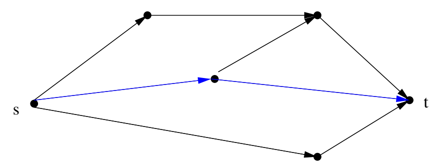

Figure 1.2: A candidate max flow shown by the blue edges. Each blue edge has a flow of one.

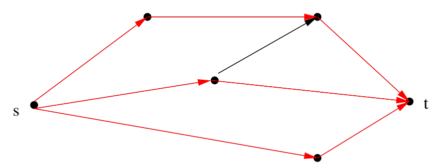

Figure 1.3: A counter example for the candidate max flow of Figure 1.2 shown by the red edges. Each red edge has a flow of one.

of the min-cut of the graph. For example, in the graph of Figure 1.1, $F = 3$ and thus, the flow in Figure 1.3 is optimal.

1.2 Running Time

We now briefly look at how to measure the efficiency of an algorithm. For an input $a_1, a_2, \cdots, a_n$, where each a_i is a rational number (not necessarily minimized), we can define the input length in two ways:

1. Number of bits required to represent the input, $L = \sum_{i=1}^{n} \log_2 a_i$. In this measure we say that an algorithm runs in polynomial time if and only if the number of steps of the algorithm is some polynomial function of L.
2. The number of inputs, n. In this case a polynomial time algorithm has number of steps which is polynomial in n.

In the first case the number of steps is essentially the total work done while in the second measure the number of steps is the number of basic arithmetic operations[1]

1.3 Linear Programming

We will start with a common example of Food Planning. Figure 1.3 list the energy, fat and carb contents of cookies (A), whole milk (B) and juice (C).

We would like to plan our food intake, that is, figure out the values of a, b and c, so that our energy intake is atleast 1500 cal, the fat content is atmost 50% and the Carb content is atmost 100%. In other words

[1]Addition, subtraction and multiplication. Division is not a basic operation. Note that since we allow rational numbers to be not necessarily minimized, one has to take care of the scenario when the number may become too big.

	A	B	C
Energy (in cal)	200	200	200
Fat (% d.v.)	20	20	0
Carb (% d.v.)	10	10	20
	a	b	c

Figure 1.4: Various properties of foods A, B and C. Each of them are consumed in quantities a, b and c respectively.

we need to find a feasible solution to the following LP:

$$200a + 200b + 200c \geq 1500 \tag{1.1}$$

$$-20a - 20b \geq -50 \tag{1.2}$$

$$-10a - 10b - 20c \geq -100 \tag{1.3}$$

It turns out that the LP given by (1.1)-(1.3) is infeasible. To see why this so, multiply (1.2) by 10 to get

$$-200a - 200b \geq -500 \tag{1.4}$$

From (1.1) and (1.4), we get $c \geq 5$. This along with (1.3) implies $a = b = 0, c = 5$. This however does not satisfy (1.1).

Now assume that B instead of being whole milk was skimmed milk, that is, the fat content was 0%. Now we need to solve the following LP:

$$200a + 200b + 200c \geq 1500 \tag{1.5}$$

$$-20a \geq -50 \tag{1.6}$$

$$-10a - 10b - 20c \geq -100 \tag{1.7}$$

One can easily verify that $a = c = 0, b = 10$ is a valid solution for the LP of (1.5)-(1.7).

Thus, checking for the feasibility of an LP is easy: just check the constraints as we did for the LP of (1.5)-(1.7). However, the method we used to check for the infeasibility of the LP of (1.1)-(1.3) was ad-hoc and we would like to develop a theory for verifying the feasibility of an LP.

Going back to the LP of (1.1)-(1.3) if we multiplied the three inequalities by λ_1, λ_2 and λ_3 respectively and got an inequality where all the coefficients on the left hand side are atmost zero while the right hand side is strictly positive, then we can prove that the LP is infeasible. In the whole milk example, $\lambda_1 = 1, \lambda_2 = 9, \lambda_3 = 10$ works.

Thus, in general for an infeasible LP with n inequalities we need to show that n multipliers $\lambda_1, \cdots, \lambda_n$ exist (while for feasible LPs none exist). In the next section we discuss this theory in more detail.

1.4 Fourier Motzkin Elimination Method

To check the feasibility of a system of linear equations, one can use Gaussian elimination and check if any determinant of the resulting matrix is zero. However, Gaussian elimination does not work if instead of equalities one is given inequalities of the following form:

$$\forall i = 1, \cdots, n \ \sum_{j=1}^{m} a_{ij} x_j \geq b_i \tag{1.8}$$

To check the feasibility of the above systems of inequalities we look at the Fourier Motzkin elimination method which is a generalization of the Gaussian elimination method and is based on the idea that one can eliminate a variable say x_j if it had coefficients of different signs in two inequalities. We now describe the method in more detail. In the discussion that follows we assume w.l.o.g. that we want to eliminate x_1.

Reorder the inequalities to obtain numbers k_1 and k_2 where $0 \leq k_1 \leq k_2 \leq n$ such that the first k_1 inequalities have positive coefficient for x_1, the next $k_2 - k_1$ have negative coefficients for x_1 and the rest do not have x_1 in them. We call these classes of inequalities as Category one, two and three inequalities respectively. Thus, for any $i_1 \in \{1, \cdots, k_1\}$:

$$x_1 \geq \frac{b_{i_1}}{a_{i_1 1}} - \sum_{j=2}^{m} \frac{a_{i_1 j}}{a_{i_1 1}} x_j \ ; \ \ a_{i_1 1} > 0 \tag{1.9}$$

and for any $i_2 \in \{k_1 + 1, \cdots, k_2\}$:

$$x_1 \leq \frac{b_{i_2}}{a_{i_2 1}} - \sum_{j=2}^{m} \frac{a_{i_2 j}}{a_{i_2 1}} x_j \ ; \ \ a_{i_2 1} < 0 \tag{1.10}$$

Thus, eliminating x_1 using Category one and two inequalities we get the following LP:

$$\begin{gathered} \sum_{j=2}^{m} \left(\frac{a_{i_1 j}}{a_{i_1 1}} - \frac{a_{i_2 j}}{a_{i_2 1}}\right) x_j \geq \frac{b_{i_1}}{a_{i_1 1}} - \frac{b_{i_2}}{a_{i_2 1}} \quad ; \ i_1 = 1, \cdots, k_1; i_2 = k_1 + 1, \cdots, k_2 \\ \sum_{j=2}^{m} a_{ij} x_j \geq b_i \quad ; \ i = k_2 + 1 \cdots, n \end{gathered} \tag{1.11}$$

We now have the following claim:

Claim 1. The LP of (1.8) is feasible if and only if the LP of (1.11) is feasible.

Proof. ($\Rightarrow$:) This direction is obvious.

($\Leftarrow$:) Suppose we have a solution $(x_2, \cdots, x_m)$ for the LP of (1.11). It is easy to see that a feasible solution to the LP of (1.8) is $(x_1, x_2, \cdots, x_m)$ where x_1 satisfies equations (1.9) and (1.10). We first look at the boundary cases when there are no category one or two inequalities. If there are no category one (two) inequalities, set x_1 to $-\infty$ (∞) and we are done. If both k_1 and $k_2 - k_1$ are non-zero, set

$$x_1 \in \left[\min_{i_2 = k_1+1, \cdots, k_2} \left(\frac{b_{i_2}}{a_{i_2 1}} - \sum_{j=2}^{m} \frac{a_{i_2 j}}{a_{i_2 1}} x_j \right), \max_{i_1 = 1, \cdots, k_2} \left(\frac{b_{i_1}}{a_{i_1 1}} - \sum_{j=2}^{m} \frac{a_{i_1 j}}{a_{i_1 1}} x_j \right) \right]$$

This completes the proof. $\square$

Now we have the following procedure to decide whether the original LP of (1.8) is feasible or not: apply the Fourier Motzkin elimination method m times to eliminate all the variables to get an LP with inequalities not involving any variables. By Claim 1, if one of the final inequalities is infeasible (which would be of the form a negative number is greater than a positive number) then the original system of inequalities is infeasible otherwise it is not.

Example 1.1. Consider the following set of inequalities: $x_1 \geq 2$, $-x_1 \geq -1$. The Fourier Motzkin elimination method gives $1 \geq 2$ which is an infeasible inequality.

Note that at each elimination step in the worst case the number of inequalities increases quadratically, that is, in the worst case, the above "algorithm" takes time $O(n^{2^m})$.

1.5 Farkas Lemma

In this section, we look at getting shorter certificates for proving infeasibility of a system of inequalities. To this end, we present two lemmas.

Lemma 1.1. *A system of inequalities $\boldsymbol{a_1}^T\boldsymbol{x} \geq b_1, \boldsymbol{a_2}^T\boldsymbol{x} \geq b_2, \cdots, \boldsymbol{a_m}^T\boldsymbol{x} \geq b_m$ has no solutions if and only if there exist positive $\lambda_1, \cdots, \lambda_m$ such that $\sum_{i=1}^m \lambda_i \boldsymbol{a_i} = \boldsymbol{0}$ and $\sum_{i=1}^m \lambda_i b_i = 1$.*

Proof. ($\Leftarrow$:) Multiply the ith inequality with the given λ_i and sum them up to get $(\sum_{i=1}^m \lambda_i \boldsymbol{a_i})\boldsymbol{x} \geq \sum_{i=1}^m \lambda_i b_i$, which by the properties of λ_is is same as infeasible inequality $0 \geq 1$.

($\Rightarrow$:) The Fourier Motzkin elimination process gives the required $\lambda_1, \cdots, \lambda_m$. As the given set of inequalities is infeasible, repeated application of the Fourier Motzkin elimination process would finally give an infeasible inequality. Note that this infeasible inequality is a just a linear combination of some subset of the original inequalities. In particular we get λ_is such that $\sum_{i=1}^m \lambda_i \boldsymbol{a_i} = \boldsymbol{0}$ and $\sum_{i=1}^m \lambda_i b_i > 0$. Scaling the λ_is appropriately gives $\sum_{i=1}^m \lambda_i b_i = 1$. Finally note that from equations (1.11), these λ_is are positive which completes the proof. $\square$

We now state the homogeneous version of Farkas Lemma:

Lemma 1.2. $\boldsymbol{a_1}^T\boldsymbol{x} \geq 0, \cdots, \boldsymbol{a_m}^T\boldsymbol{x} \geq 0 \;\Rightarrow\; \boldsymbol{c}^T\boldsymbol{x} \geq 0$ *if and only if there exist positive $\lambda_1, \cdots, \lambda_m$ such that $\sum_{i=1}^m \lambda_i \boldsymbol{a_i} = \boldsymbol{c}$.*

Proof. ($\Leftarrow$:) In this case we have positive λ_is such that $\sum_{i=1}^m \lambda_i \boldsymbol{a_i}^T = \boldsymbol{c}^T$. Applying this fact to $\sum_{i=1}^m \lambda_i \boldsymbol{a_i}^T\boldsymbol{x} \geq \sum_{i=1}^m \lambda_i \cdot 0$, we have $\boldsymbol{c}^T\boldsymbol{x} \geq 0$ as desired.

($\Rightarrow$:) In this case we have that the following infeasible system of inequalities $\boldsymbol{a_1}^T\boldsymbol{x} \geq 0, \cdots, \boldsymbol{a_m}^T\boldsymbol{x} \geq 0, \boldsymbol{c}^T\boldsymbol{x} < 0$. By suitably scaling $\boldsymbol{x}$ one can obtain an equivalent set of inequalities $\boldsymbol{a_1}^T\boldsymbol{x} \geq 0, \cdots, \boldsymbol{a_m}^T\boldsymbol{x} \geq 0, -\boldsymbol{c}^T\boldsymbol{x} \geq 1$. As this set is also infeasible, applying Lemma 1.1, we get positive $\lambda_1, \cdots, \lambda_{m+1}$ such that $\sum_{i=1}^m \lambda_i \cdot 0 + \lambda_{m+1} \cdot 1 = 1$ and $\sum_{i=1}^m \lambda_i \boldsymbol{a_i} = \lambda_{m+1}\boldsymbol{c}$. The first relation gives $\lambda_{m+1} = 1$ which simplifies the second relation to $\sum_{i=1}^m \lambda_i \boldsymbol{a_i} = \boldsymbol{c}$ as desired. $\square$

In the next lecture we will look at convexity and derive an alternate proof of Lemma 1.2 as well as the non-homogeneous version of Farkas Lemma.