Saarland
Informatics Campus

Script of Lecture 11, Approximation Algorithms
Summer term 2017

Tobias Mömke, Hang Zhou
http://www-cc.cs.uni-saarland.de/course/61/

Written by Tobias Mömke

1 Iterated Linear Programming

In the LP rounding approaches that you have learned before in this class, the basic idea was to compute an optimal solution to a linear programming relaxation of a problem and then to use the assignment of values to the variables to construct a feasible integral solution. In iterated linear programming, we proceed similarly but with one important difference. We use the computed LP solution not to solve the entire instance, but only to fix a small part of the solution. Afterwards, we consider the remaining problem instance for which we have no integral solution yet. We then solve the LP for the remaining solution and again fix a part of the solution. We iterate the process until we have obtained an integral solution to the entire instance.

In order to be able to apply this powerful technique, we have to review/introduce some properties of linear programs.

2 Rank Lemma

In every linear program, we want to optimize an objective function subject to a set of constraints. Instead of writing down the LP as sums of linear and constant terms, we can use a standard form. We use column vectors to specify the cost, the variables, and the constant part of the constraints. The linear part of the constraints is a matrix A. If we want to minimize the cost, we can write the LP as

$$\min x^T c \tag{1}$$
$$\text{s.t.} \quad Ax \geq b \tag{2}$$
$$x \geq 0 \tag{3}$$

The notation $x \geq 0$ means that each entry of x is at least zero. Suppose we have two variables x_e, x_f and a cost function c. Then a linear program could state to minimize $x_e \cdot c(e) + x_f \cdot c(f)$ subject to $x_e + x_f \geq 2$ and $x_e, x_f \geq 0$. Then $x = (x_e, x_f)^T$, $c = (c(e), c(f))^T$, $A = (1, 1)$, and $b = (2)$.

The maximization version can be written as

$$\max x^T c \tag{4}$$
$$\text{s.t.} \quad Ax \leq b \tag{5}$$
$$x \geq 0 \tag{6}$$

In the following, we only concentrate on minimization problems, but all results work for maximization problems in a similar manner.

In order to apply iterated rounding, a crucial part is that we want to analyze the number of non-zero variables in the computed solution. In order to be able to do that, we have to consider extreme point solutions (also called vertex solutions or basic solutions).

Definition 1. Let $P = \{x : Ax \geq b, x \geq 0\} \subseteq \mathbb{R}^n$. Then a vector $x \in P$ is called an *extreme point solution* if there does not exist a non-zero vector $y \in \mathbb{R}^n$ such that both $x + y$ and $x - y$ is in P.

Luckily, we know from combinatorial optimization that there is always an optimal solution that is an extreme point solution (if it is finite) and we can find such a solution efficiently.

With this preparation we are ready to state the rank lemma.

Lemma 1 (Rank Lemma)**.** *Let $P = \{x : Ax \geq b, x \geq 0\}$ and let x be an extreme point solution of P such that $x_i > 0$ for each i. Then every maximal number of linearly independent tight constraints of the form $A_i x = b_i$ for some row i of A equals the number of variables.*

Proof. We use the property of linear programs that only the tight constraints (i.e., constraints that hold with equality) contribute to the solution. Formally, each extreme point solution has the property that the sub-matrix formed by the tight constraints has the same rank as A. We will use the result as a black box – a proof can be found in reference [1].

Observe that the number of columns of A equals the number of non-zero variables since we assume that there are no zero variables. We claim that the columns of A are linearly independent, since the possibility to express a column as a linear combination of other columns would contradict the definition of extreme points (the details are left as an exercise). We know from linear algebra that the row rank of a matrix equals the column rank. The row rank in turn equals the number of linearly independent rows of A, and the claim follows. □

The high level idea of one iteration of iterated rounding then is to solve an LP, round up large enough variables, and to obtain a simpler LP that does not contain variables with integer values anymore. The rank lemma allows us to employ counting arguments that ensure to obtain integer values in every iteration.

3 Degree Bounded Spanning Trees

We will use iterated rounding to compute degree bounded spanning trees. Given a graph G, a degree-d-bounded spanning tree is a tree that contains all vertices $V(G)$ and no vertex has a degree larger than d. We aim to find such a tree of minimum cost. For simplicity, we assume that G is complete; non-existing edges can be assigned the value ∞. Observe that we do *not* assume G to be metric.

For degree bounded spanning trees, we will not consider the approximation ratio as before. Instead we will show how to obtain an *additive* approximation. We aim to show the following result.

Theorem 1. *Given a graph G and a number d, there is a degree-$(d+1)$-bounded spanning tree of at most the cost of an optimal degree-d-bounded spanning tree.*

Thus the parameter that we approximate is not the cost of the solution but the degree bound. And we do not increase the degree by a factor, but instead we add a constant.

We will approach this result step by step. To this end, we first show how to use iterated linear programming in the absence of degree constraints, i.e., we compute a minimum cost spanning tree.

4 Minimum Spanning Tree

The minimum spanning tree problem (MST) can be formulated with the following LP (T)[1].

[1]The LP is known to describe the spanning tree polytope, i.e., all extreme point solutions are integral. The proof presented here implies the classical result.

$$\text{Minimize} \quad \sum_{e \in E} x_e c_e \tag{7}$$

$$\text{s.t.} \quad \sum_{e \in E[S]} x_e \leq |S| - 1 \quad \text{for all } \emptyset \neq S \subset V \tag{8}$$

$$\sum_{e \in E[V]} x_e = |V| - 1 \tag{9}$$

$$x_e \geq 0 \qquad \text{for all } e \in E \tag{10}$$

In the formulation, $E[S]$ denotes the set of edges in the sub-graph induced by S. Observe that (T) has exponentially many constraints. Using a standard minimum cut algorithm, one can design a separation oracle.

We will show how to iteratively determine leaves of an MST using iterated rounding. The final result will be an optimal solution.

5 Characterization of Extreme Point Solutions

In order to use the rank lemma, we have to identify a set of tight linearly independent constraints. This set should have some structure that is useful for us. To this end, for each edge set $F \subseteq E(G)$ we consider the characteristic vector $\chi(F)$. The characteristic vector has an entry for each $e \in E$. The entry of e is 1 if $e \in F$ and 0 otherwise.

5.1 Supermodularity

We show that χ has a property called supermodularity. Let $E[X, Y]$ be the edges with one end in X and the other end in Y.

Lemma 2. *For all $X, Y \subset V$,*

$$\chi(E[X]) + \chi(E[Y]) \leq \chi(E[X \cup Y]) + \chi(E[X \cap Y])\,.$$

Equality holds if and only if $E[X \setminus Y, Y \setminus X] = \emptyset$.

Proof. We have

$$\chi(E[X]) + \chi(E[Y]) = \chi(E[X \cup Y]) + \chi(E[X \cap Y]) - E[X \setminus Y, Y \setminus X]\,,$$

by considering all edges within or between the sets used in the equality. □

5.2 Tight Constraints

We now consider the set of tight constraints $\mathcal{F} = \{S \subseteq V : x(E[S]) = |S| - 1\}$. (To be precise, this is a family of vertex sets, but each set corresponds to a constraint of (T).) This family is closed under intersection and union.

Lemma 3. *If $S, T \in \mathcal{F}$ and $S \cap T \neq \emptyset$, then both $S \cap T$ and $S \cup T$ are in $\mathcal{F}$. Furthermore, $E[S \setminus T, T \setminus S] = \emptyset$.*

Proof. We obtain

$$|S| - 1 + |T| - 1 = x(E[S]) + x(E[T]) \tag{11}$$

$$\leq x(E[S \cap T]) + x(E[S \cup T]) \tag{12}$$

$$\leq |S \cap T| - 1 + |S \cup T| - 1 \tag{13}$$

$$\leq |S| - 1 + |T| - 1 \tag{14}$$

using the following arguments.

- The constraints for S and T are tight.
- We apply Lemma 2.
- We use the constraints of (T).
- $|S| + |T| = |S \cap T| + |S \cup T|$ for all sets S, T.

Hence all five terms are equal. We conclude that $x(E[S \cap T]) = |S \cap T| - 1$ and $x(E[S \cup T]) = |S \cup T| - 1$, i.e., also these constraints are tight. □

5.3 Laminar Families

A set of subsets $\mathcal{L}$ is called a *laminar family* if for all $X, Y \in \mathcal{L}$, either $X \subset Y$ or $Y \subset X$ or $X \cap Y = \emptyset$.

Laminar families are directly related to the set of tight feasible solutions $\mathcal{F}$.

Lemma 4. *There is a laminar family $\mathcal{L} \subseteq \mathcal{F}$ such that both $\mathcal{L}$ and $\mathcal{F}$ have the same maximal number of linearly independent constraints.*

We leave the proof of the lemma as an exercise.

References

[1] Lap-Chi Lau, R. Ravi, and Mohit Singh. *Iterative Methods in Combinatorial Optimization.* Cambridge University Press, New York, NY, USA, 1st edition, 2011.