Topics in Theoretical Computer Science March 9, 2015

Lecture 4 (based on [3] and [6])

Lecturer: Ola Svensson *Scribes: Bartlomiej Dudek*

1 Combinatorial Optimization: Bipartite Matchings and LPs

Last lecture:

- Bipartite Expanders (existence through probabilistic method, deterministic constructions exist)
- Applications: Error Correcting Codes, Saving Random Bits

In this lecture[1] we shall start our journey into combinatorial optimization. We shall see

- Combinatorial Algorithm for Bipartite Matching (König's theorem and Hall's theorem)
- Linear programming (Extreme point structure)
- Minimum weight perfect matching

2 Basics: Matchings

Bipartite matching is a basic combinatorial optimization problem arising in many different applications, like:

- scheduling - consider employees, jobs to do and each employee is capable of doing only certain jobs - what is the biggest number of jobs that can be done simultaneously?
- ad allocation - for example Google AdWords use generalization of online bipartite matching problem to decide what ads to display with each query so as to maximize its revenue [1]
- chemistry - many properties of chemical compounds depend on perfect matching of their carbon skeleton
- economy - stable allocation of goods, workers, spouses...

Definition 1 (Matching) *For a graph $G = (V, E)$ a matching $M \subseteq E$ is a subset of the edges so that every vertex $v \in V$ is incident to at most one edge in M, i.e. $|\{e \in M : v \in e\}| \leq 1$ for all $v \in V$.*

A matching M is perfect *if every vertex is "matched", that is every vertex is incident to exactly one edge in M.*

We will be interested in the following problems on bipartite graphs:

Maximum cardinality matching problem: Find a matching M of maximum size.

Minimum weight/cost perfect matching problem: Given a cost c_{ij} for all $(i, j) \in E$, find a perfect matching of minimum cost where the cost of a matching M is given by $c(M) = \sum_{(i,j)\in M} c_{ij}$.

The same problems are also interesting on general graphs and efficient, but more complicated algorithms exist, like Edmond's *blossom algorithm* [2] that solves maximum cardinality matching problem in general graphs in $O(\sqrt{|V|} \cdot |E|)$ time or a randomized approach [5] running in $O(|V|^\omega)$ time, where ω is the exponent of best known matrix multiplication algorithm (today $\omega \approx 2.373$) [7].

[1] Large parts of this lecture is based on *http://math.mit.edu/ goemans/18433S11/matching-notes.pdf*

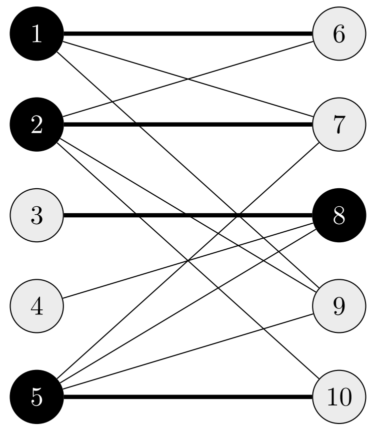


Figure 1: The edges $(1,6),(2,7),(3,8)$ and $(5,10)$ form a matching (of maximum cardinality). Vertices $1,2,5,$ and 8 form a vertex cover (of minimum cardinality).

3 Maximum cardinality matching problem

3.1 Duality

Before giving an algorithm, perhaps an easier question is: how would you give a short proof that a given matching is optimal?

For this purpose, one would like to find upper bounds on the size of the largest matching and hope that the smallest of these upper bounds be equal to the size of the largest matching. This is a *duality* concept that will be very important in this subject. In this case, the dual problem will be a famous combinatorial optimization problem: vertex cover.

Vertex cover: A vertex cover is a set C of vertices so that all edges e of E are incident to at least one edge in C. In other words, there is no edge completely contained in $V \setminus C$.

We clearly have the following:

$$|M| \leq |C| \qquad \text{for any matching } M \text{ and vertex cover } C.$$

This follows from the fact that, given any matching M, a vertex cover C must contain at least one of the end points of each edge in M.

This is *weak duality*: The maximum size of a matching is at most the minimum size of a vertex cover. We shall in fact prove strong duality (that equality holds) for bipartite graphs:

Theorem 2 (König 1931) *For any bipartite graph, the maximum size of a matching is equal to the minimum size of a vertex cover.*

The proof of this theorem will be algorithmic. It will give an efficient algorithm for both finding a maximum size matching and a minimum size vertex cover of a bipartite graph. Note that whenever one has min max statement as above, the problem lies in $NP \cap coNP$ which may indicate that it has an efficient algorithm.

3.2 Algorithm

Recall that a path is a collection of edges $(v_0,v_1),(v_1,v_2),\ldots,(v_{k-1},v_k)$ where the v_i's are distinct vertices. We can simply represent a path as $v_0 - v_1 - v_2 - \ldots - v_k$.

Definition 3 (Alternating path) *An* alternating path *with respect to M is a path that alternates between edges in M and edges in $E \setminus M$.*

Definition 4 (Augmenting path) *An* augmenting path *with respect to M is an alternating path in which the first and last vertices are unmatched.*

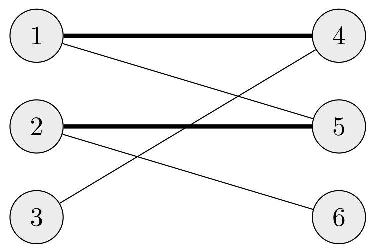


Figure 2: The edges $(3,4),(4,1),(1,5),(5,2),(2,6)$ form an augmenting path

The definition of an augmenting path motivates the following algorithm:

Algorithm:

1. Initialization: let $M = \emptyset$
2. While exists an augmenting path P, update $M = M \Delta P \equiv (M \setminus P) \cup (P \setminus M)$.
3. Return M.

Exercise 1 *Devise an efficient algorithm for finding an augmenting path P (if one exists). What is the total running time of the matching algorithm?*

Solution

Recall, that the graph is bipartite so $V = L \cup R$ and all edges are between L and R.

We can run from all unmatched vertices of L a BFS or DFS that goes from left to right using edges from $E \setminus M$ and from right to left using edges from M. If it reaches an unmatched vertex in R we have an augmenting path. Note, that it's no difference from which "side" of graph we start our search - we'll end in the second.
Clearly, finding an augmenting path takes $O(|V|+|E|)$ time as we need to visit every edge at most once. Furthermore maximum matching can have cardinality $O(|V|)$ so the total running time of the matching algorithm is $O(|V| * (|V| + |E|))$. [2]

We now prove that the algorithm indeed finds a maximum matching. (The statement and proof is taken from [3]).

Theorem 5 *A matching M is maximum if and only if there are no augmenting paths with respect to M.*

Proof (By contradiction)

($\Rightarrow$) Let P be some augmenting path with respect to M. Then $M' = M \Delta P$ is matching of greater cardinality than M. This contradicts the optimality of M.

($\Leftarrow$) If M is not maximum, let M^* be a maximum matching so that $|M^*| > |M|$. Let $Q = M \Delta M^*$. Then

[2] It can be done even better: Hopcroft-Karp algorithm that finds many shortest augmenting paths in each iteration simultaneously, runs in time $O(\sqrt{|V|} \cdot |E|)$. Very recently Aleksander Madry [4] further improved this running time for sparse graphs.

- Q has more edges from M^* than from M (since $|M^*| > |M|$ implies that $|M^* \setminus M| > |M \setminus M^*|$).
- Each vertex is incident to at most one edge in $M \cap Q$ and one edge in $M^* \cap Q$, so each vertex is incident to $0, 1$ or 2 edges from $M \Delta M^*$.
- Thus Q is composed of cycles and paths that alternate between edges from M and M^*.
- Therefore there must be some path with more edges from M^* in it than from M (as in total in Q there are more edges from M^* than in M and cycles consist of the same number of edges from M^* and M). This path is an augmenting path with respect to M.

Hence, there must exist an augmenting path P with respect to M, which is a contradiction. ■

3.3 Proof of König's Theorem

Consider a maximum matching M^* of $G = (A \cup B, E)$. Let L the vertices reachable from unmatched vertices in A by alternating paths with respect to M^*. The following proves König's theorem.

(The proof is taken from [3].)

Lemma 6 *We have that* $C^* = (A \setminus L) \cup (B \cap L)$ *is a vertex cover. Moreover,* $|C^*| = |M^*|$.

Proof We first show that C^* is a vertex cover. Suppose toward contradiction that it is not. Then there is an edge $e = (a, b) \in E$ with $a \in A \cap L$ and $b \in B \setminus L$. The edge cannot belong to the matching. If it did then b should be in L because otherwise a would not be in L, but $b \notin L$. Hence $e \in E \setminus M$. This however, implies that there is an alternating path from an unmatched vertex to b (namely go to a and take the edge (a, b)) contradicting the fact that $b \notin L$.

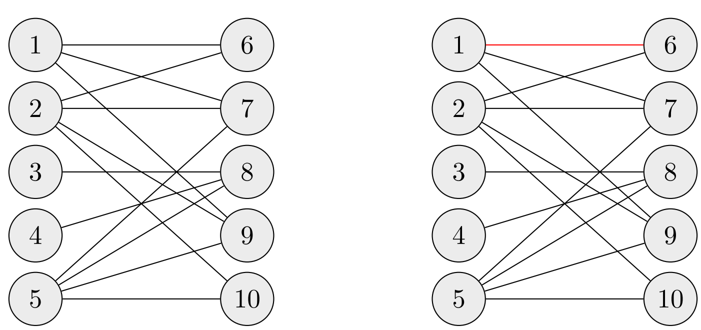
To show the second part of the proof, we show that $|C^*| \leq |M^*|$, since the reverse inequality is true for any matching and any vertex cover. The proof follows from the following observations:

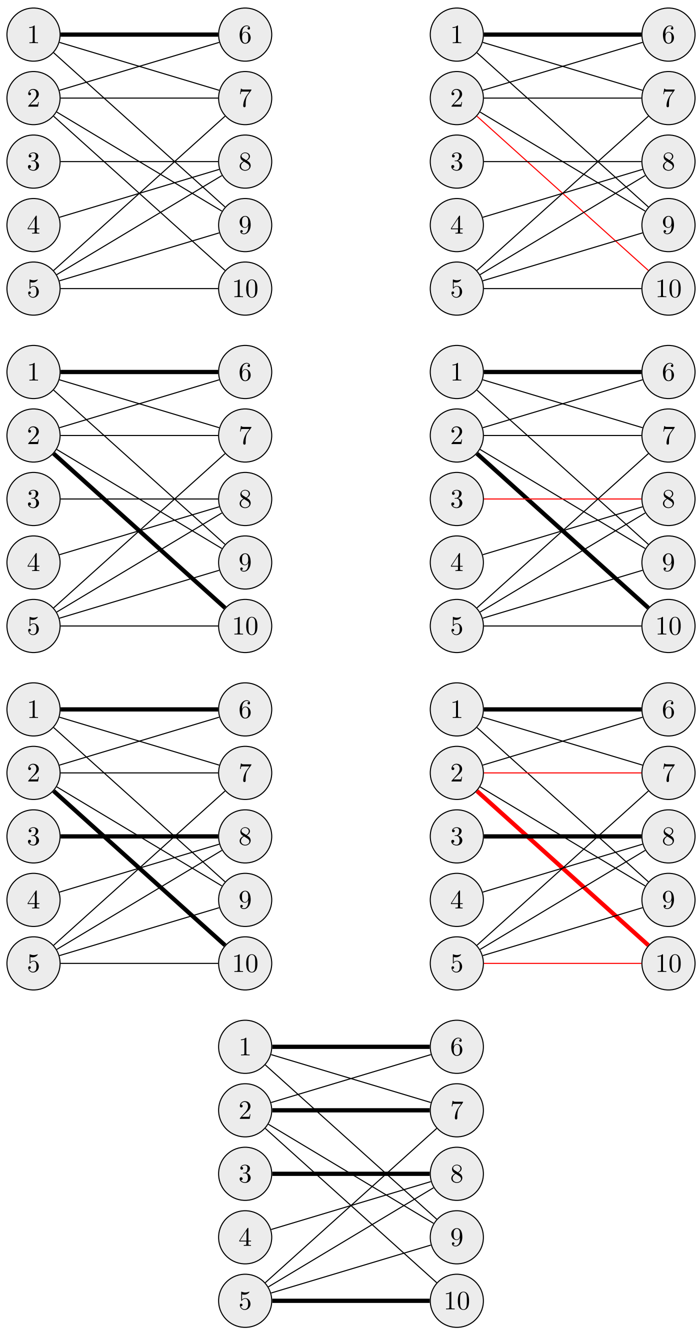
1. No vertex in $A \setminus L$ is unmatched by the definition of L;
2. No vertex in $B \cap L$ is unmatched since this would imply the existence of an augmenting path (which contradicts that M^* is optimal).
3. There is no edge of the matching between a vertex $a \in A \setminus L$ and a vertex $b \in B \cap L$. Otherwise a would be in L, because either the alternating path to b goes through a or we can extend it to a from b.

These remarks imply that every vertex in C^* is matched and moreover the corresponding edges of the matching are distinct. Hence, $|C^*| \leq |M^*|$. ■

3.4 Example of Algorithm

Here we can see augmenting paths (in red color) that are found in each iteration of the algorithm:





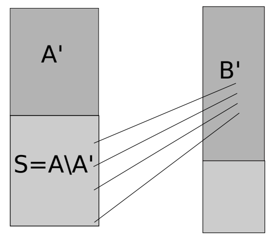
3.5 Exercises

(Exercises are taken from [3].)

Exercise 2 *Deduce Hall's theorem from König's theorem:*

Theorem 7 (Hall'35) *Given a bipartite graph $G = (A \cup B, E)$ G has a matching of size $|A|$ if and only if for every $S \subseteq A$ we have $|N(S)| \geq |S|$, where $N(S) = \{b \in B | \exists a \in S \text{ with } (a, b) \in E\}$.*

Proof Clearly right-hand-side is a necessary condition (otherwise we would have small proof that there's no matching of size $|A|$). To prove sufficiency let's suppose that (i)$|M^*| < |A|$ and we'll show that exists $S : |N(S)| < |S|$. From König's theorem exists vertex cover C^* s.t. $|C^*| = |M^*|$. Let $A' = C^* \cap A, B' = C^* \cap B$, hence (ii)$|A'| + |B'| = |M^*|$.



Since $A' \cup B'$ is vertex cover, all neighbours of vertices in $S = A \setminus A'$ lie in B', so (iii) $N(S) \subseteq B'$ and we have:

$$|S| = |A \setminus A'| = |A| - |A'| > |M^*| - |A'| = |A'| + |B'| - |A'| = |B'| \geq N(S)$$

So finally we get: $|S| > N(S)$ what proves that RHS from Hall's theorem is both necessary and sufficient for existence of matching of size $|A|$. ■

Exercise 3 *Show that in any graph (not necessarily bipartite) the size of minimum vertex cover is at most twice the size of maximum matching.*

Solution Clearly taking all ends of edges from maximum matching gives us a vertex cover. Otherwise suppose there is an edge e that has both ends unmatched, then our matching wasn't maximum, because we can extend it adding the edge e.

Exercise 4 *An edge cover of a graph $G = (V, E)$ is a subset R of E such that every vertex of V is incident to at least one edge in R. Let G be a bipartite graph with no isolated veretx. Show that the cardinality of the minimum edge cover R^* of G is equal to $|V|$ minus the cardinality of the maximum matching M^* of G. Give an efficient algorithm for finding the minimum edge cover of G. Is it also true for non-bipartite graphs?*

Solution Let's consider a general graph, as the statement is true for all graphs.
Let R^* - minimum edge cover, M^* - maximum matching, V - set of vertices in G. We'll bound the size of edge cover from two sides:

Firstly, (i) $|R^*| \leq |M^*| + (|V| - 2|M^*|) = |V| - |M^*|$ as we can take all edges from M^* and one edge from each unmatched vertex.

Secondly, let's consider minimum edge cover R^* and let M - maximum matching on G using edges only from R^*.

There are $|V| - 2|M|$ unmatched vertices and there is no edge in R^* between any pair of them - otherwise M won't be maximal. As R^* is an edge cover, for each unmatched vertex v there will be exactly one edge in R^* incident to v and one matched vertex.

Thus we have $|R^*| = |M| + |V| - 2|M| = |V| - |M| \geq |V| - |M^*|$ as $|M| \leq |M^*|$, beacuse M^* is maximum matching using all edges E of G, not only those from R^*.
So we have that $|V| - |M^*| \leq |R^*| \leq |V| - |M^*|$ so $|R^*| = |V| - |M^*|$. ■

In part (i) we showed how to construct an edge cover of that size: simply take all edges from M^* and edges from each of unmatched vertex. We showed in Exercise 1 how to find maximum matching it bipartite graphs in $O(|V| \cdot (|V| + |E|))$ time and also mentioned how to do it faster. In [2] there's described an algorithm finding maximum matching in general graphs, but it's more complicated.

4 Linear Programming

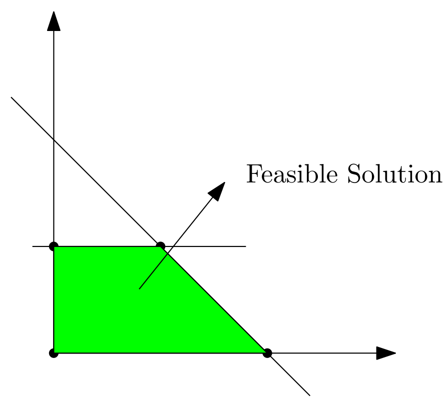
(Large parts of the following is taken from [6].)

A linear programing problem is the problem of finding values for variables that optimize a given linear objective function, subject to linear constraints.

Let us state an LP problem.

$$\begin{aligned} \text{Maximize} \quad & x + y \\ \text{Subject to} \quad & x + y \leq 2 \\ & y \leq 1 \\ & x, y \geq 0 \end{aligned}$$

The following figure shows the feasible area.



Definition 8 *A linear programming problem is the problem of finding values for n variables $x_1, x_2, \ldots, x_n \in \mathbb{R}$ that minimize (or equivalently, maximize) a given linear objective function, subject to m linear constraints*

$$\begin{aligned} \textbf{minimize:} \quad & \sum_{i=1}^{n} c_i x_i & \\ \textbf{Subject to:} \quad & \sum_i e_{i,j} x_i = b_j & \text{for } j = 1, \ldots, m_1 \\ & \sum_i d_{i,k} x_i \geq g_k & \text{for } k = 1, \ldots, m_2 \\ & \sum_i f_{i,p} x_i \leq l_p & \text{for } p = 1, \ldots, m_3 \end{aligned}$$

where $m_1 + m_2 + m_3 = m$.

4.1 Motivation

We see that Linear Programming is a very powerful tool - it can be used in many applications, industrial and theoretical as well, like:

- obtaining an optimal production plan to maximize a profit of a factory

- modeling network flow problems (what is maximum possible flow that doesn't exceed capacity of each connection)
- microeconomics
- theory - many theoretical problems can be introduced as Integer Program (LP in which $x_i \in \mathbb{Z}$) that can be relaxed to LP obtaining p.e. a good approximation of optimal result- we'll see such applications later in the course

4.2 Some history

It was first formalized and applied to problems in economics in the 1930s by Leonid Kantorovich. Kantorivich's work was hidden behind the Iron Curtain (where it was largely ignored) and therefore unknown in the West. Linear programming was rediscovered and applied to shipping problems in the early 1940s by Tjalling Koopmans.
Simplex Method was published by George Dantzig in 1947: it is the first complete algorithm to solve linear programming problems.
The principle is the following: we start from an extreme point and then we look at its neighbors. If one of these is better we move to it and continue in the same way, else we stop. Once we stop, we can be sure that we have an optimal solution, since we're in a convex polytope. Even if it's usually extremely fast, we know some bad examples where this method visits an exponential number of extreme points before to reach a solution, so this method does not always run in polynomial time.
Ellipsoid Method was studied by Leonid Khachiyan in the seventies.
This method is guaranteed to run in poly time (exactly $poly(n, m, \log(u))$ where u is the largest constant) but it is slow in practice.
We do a binary search for the optimal value of the objective function, so we can add the objective function as a constraint, and just need to decide if there exists a feasible point. We start by taking an ellipsoid surrounding the feasible area. We then check the center of the ellipsoid, if it's inside the area then we have our solution, else we identify a violated constraint, and cut our ellipsoid in two parts using this constraint, and construct a new ellipsoid around the part of the old ellipsoid in which the constraint is satisfied. We repeat this process until we find a feasible point or can be sure that the feasible area is empty.
Interior point Method developed by Narendra Karmarkar in 1984.
In this Method we move in the region to find an OPT solution.

4.3 Extreme Points

Let us first define an extreme point:

Definition 9 *A feasible solution is an extreme point if it cannot be written as a convex combination of other feasible solutions.*

Just to recall:

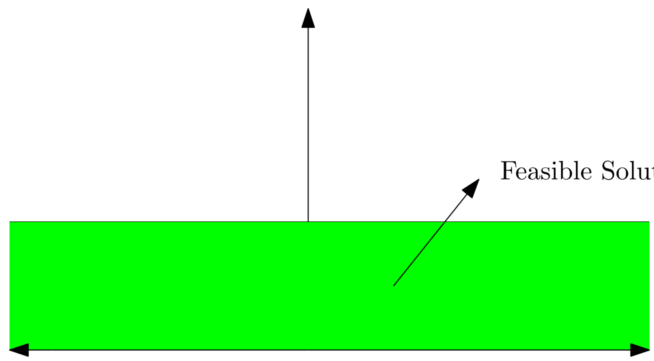
Definition 10 *A convex combination of points $x_1, x_2, ..., x_n$ is a point of the form $\sum_{i=1}^{n} \lambda_i x_i$ where the real numbers λ_i satisfy $\sum_{i=1}^{n} \lambda_i = 1$ and $\forall_i \lambda_i \in [0, 1]$*

Now, we state a theorem about extreme points. Extreme points are important because sometimes they have useful structural properties, which we can exploit to round LP solutions.

Theorem 11 *If the feasible region is bounded, there always exists an optimum which is an extreme point.*

If the feasible region is not a bounded, then we might not have any extreme points. For example, the following LP does not contain any extreme points.

$$\begin{aligned} \text{Maximize} \quad & y \\ \text{Subject to} \quad & y \leq 1 \\ & y \geq 0 \end{aligned}$$



Exercise 5 *(LP-duality) Consider the problem of finding* $x_1, x_2, x_3 \in \mathbb{R}$ *so as to*

$$\begin{aligned} \textit{minimize} \quad & 7x_1 + x_2 + 5x_3 \\ \textit{subject to} \quad & x_1 - x_2 + 3x_3 \geq 10 \\ & 5x_1 + 2x_2 - x_3 \geq 6 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

How can you give a short certificate that there is a solution of value 26*?*
How can you give a short certificate that there is no solution of value less than 26*?*

Solution Consider such a solution: $x_1 = 1.75, x_2 = 0, x_3 = 2.75$. Clearly it meets all constraints and it's value is 26.

Now let's multiply the first inequality by 2 and add it to the second and fourth obtaining: $7x_1 + x_2 + 5x_3 \geq 26$ so that's a short certificate that there is no solution of value less than 26.

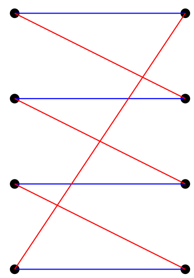
5 Minimum Cost Perfect Matching

In this section we are going to concentrate on minimum cost bipartite perfect matching. The LP for this problem is (we may assume the graph is complete bipartite, why?)

$$\begin{aligned} \text{Minimize} \quad & \sum_{e \in E} x_e c_e \\ \text{Subject to} \quad & \sum_{e=(a,b) \in E} x_e = 1 \quad \forall a \in A \\ & \sum_{e=(a,b) \in E} x_e = 1 \quad \forall b \in B \\ & x_e \geq 0 \quad \forall e \in E \end{aligned}$$

Claim 12 *Any extreme point solution to Matching LP is integral.*

Proof Let x^* be an extreme point for the graph $G = (V_1, V_2, E)$ and let $E_f = \{0 < x^*_e < 1\}$. Suppose towards contradiction that $E_f \neq \emptyset$. Note that E_f must then contain a cycle: indeed any vertex incident to an edge in E_f is incident to at least two edges in E_f. All these edges are fractional and we want to



define y and z so that they are feasible solutions and $x^* = \frac{1}{2}(y + z)$. Let $e_1, e_2, ..., e_{2k}$ be the edges of the cycle. Let y, z be

$$y_e = \begin{cases} x^*_e + \epsilon & \text{if } e \in \{e_1, e_3, e_5, ..., e_{2k-1}\} \\ x^*_e - \epsilon & \text{if } e \in \{e_2, e_4, e_6, ..., e_{2k}\} \\ x^*_e & \text{otherwise} \end{cases}$$

$$z_e = \begin{cases} x^*_e - \epsilon & \text{if } e \in \{e_1, e_3, e_5, ..., e_{2k-1}\} \\ x^*_e + \epsilon & \text{if } e \in \{e_2, e_4, e_6, ..., e_{2k}\} \\ x^*_e & \text{otherwise} \end{cases}$$

Now one need to choose such small ϵ that both y and z are feasible, so all the numbers y_e and z_e are in $[0, 1]$. For example $\epsilon = min\{min\{x^*_e, 1 - x^*_e\} : e \in E_f\}$ gives y and z feasible. Now one can easily see that $x^* = \frac{1}{2}(y + z)$ what contradicts with assumption that x^* is an extreme point. ■

Because of the above theorem, the polytope

$$P = \{x : \sum_{b:(a,b)\in E} x_{(a,b)} = 1 \qquad a \in A$$
$$\sum_{a:(a,b)\in E} x_{(a,b)} = 1 \qquad b \in B$$
$$x_{(a,b)} \geq 0 \qquad (a,b) \in E\}$$

is called the *bipartite perfect matching polytope.*

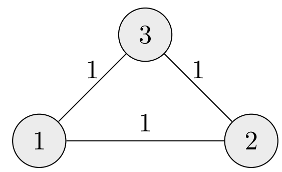
Exercise 6 *Show that the LP relaxation for matching is not an exact relaxation for non-bipartite graphs. Then show that any extreme point is half-integral.*

(Hint: use the fact that we can map every feasible point into a feasible point for the bipartite case and vice-versa)

Solution Let's write an LP for the maximum matching problem:

$$\text{Maximize} \quad \sum_{e\in E} x_e$$
$$\text{Subject to} \quad \sum_{e=(a,b)\in E} x_e = 1 \quad \forall a \in V$$
$$x_e \geq 0 \quad \forall e \in E$$

Here is an example showing that it is not an exact relaxation:



Because we take $\forall_{e \in E} e = 1/2$ that gives optimal value, but is not integral.

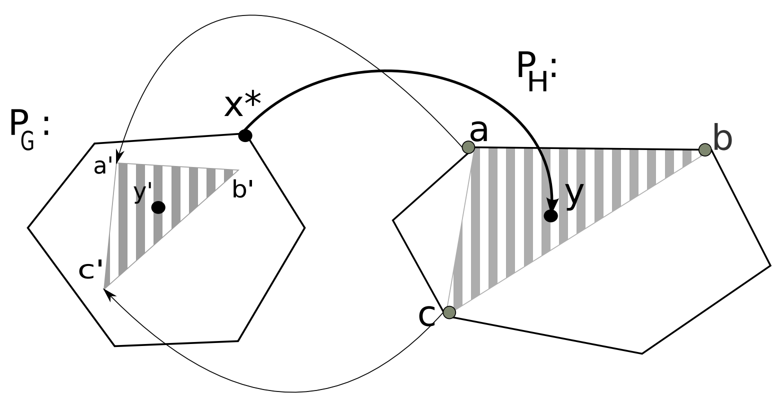
Half-integrality Now, considering LP for maximum matching in graph G we'll think of it's solutions as points in a polytope P_G formed by the given LP.
We just learned that all extreme points in P_G are integral if G is bipartite. To see that every extreme point in polytope of general graph is half-integral, we need to consider such bipartite graph: $H = (L \cup R, F)$, where $L = \{u_L : u \in V\}, R = \{u_R : u \in V\}$ and $F = \{(u_L, v_R) : (u,v) \in E\} \cup \{(v_L, u_R) : (u,v) \in E\}$.

Let's see how we can map feasible points from P_G (let's denote them as x) to feasible points (y) in P_H and vice-versa. The mapping will only "look" at correspoing edges:

$$G \to H : \forall (u,v) \in E : y_{(u_L, v_R)} := y_{(v_L, u_R)} := x_{(u,v)}$$

$$H \to G : \forall (u,v) \in E : x_{(u,v)} := \frac{y_{(u_L, v_R)} + y_{(v_L, u_R)}}{2}$$

Observe that mapped points lie inside corresponding polytopes and mapping from G to H doubles value of the objective function whilst mapping from H to G divides by 2.



Suppose toward contradiction that there is an extreme point x^* of polytope P_G that is not half-integral. It can be mapped to a point $y \in P_H$ that is not an extreme point (as it's not integral), so it's a convex combination of $2|F| + 1$ extreme integral points of P_H (in the picture: a, b, c) [3]. It's easy to see that integral points of P_H are mapped to half-integral points of P_G so y' (mapping of y into P_G) is a convex combination of half-integral points of P_G (here: a', b', c'). That leads to a contradiction with assumption that x^* is an extreme point, so all extreme points of P_G are half-integral. ■

References

[1] A. Mehta, A. Saberi, U. Vazirani, V.Vazirani: *AdWords and Generalized On-line Matching*, 2007.

[2] J. Edmonds: *Paths, trees, and flowers*, 1965.

[3] Michel X. Goemans: *Lecture notes on bipartite matching*, 2009, *http://www-math.mit.edu/ goemans/18433S09/matching-notes.pdf*

[3] Carathéodory's theorem states that in d-dimensional space a convex hull enclosing the points consits of $d+1$ points

[4] Aleksander Madry: *Navigating Central Path with Electrical Flows: From Flows to Matchings, and Back*, 2013.

[5] M. Mucha, P. Sankowski *Maximum Matchings via Gaussian Elimination.*

[6] Ashkan Norouzi-Fard, Christos Kalaitzis: *Scribes of Lecture 9 in Topics in TCS 2014.* *http://theory.epfl.ch/courses/topicstcs/Lecture9.pdf*

[7] V. Vassilevska Williams *Multiplying matrices in* $O(n^{2.373})$ *time*, 2012.