Introduction to Modern Combinatorial Optimization **POSTECH, Winter 2025**
Lecture #8: the matching polytope and separation January 16, 2025
Lecturer: Dabeen Lee

Outline

In this lecture, we introduce the matching polytope that arises as the convex hull of solutions to the integer programming formulation of the maximum weight matching problem. We present the general framework based on the ellipsoid method and separation.

1 Matching polytope

In this section, we provide a linear programming-based approach for computing a maximum weight matching in a general graph. Recall that a maximum weight matching in a graph $G = (V, E)$ with weights $w \in \mathbb{R}^{|E|}$ can be computed by solving the following integer linear program.

$$\begin{aligned} \text{maximize} \quad & \sum_{e\in E} w_e x_e \\ \text{subject to} \quad & \sum_{v\in V: uv\in E} x_{uv} \leq 1 \quad \text{for all } u \in V, \\ & x_e \in \{0,1\} \quad \text{for all } e \in E. \end{aligned} \tag{8.1}$$

Moreover, when G is bipartite, our approach was to take its LP relaxation

$$\begin{aligned} \text{maximize} \quad & \sum_{e\in E} w_e x_e \\ \text{subject to} \quad & \sum_{v\in V: uv\in E} x_{uv} \leq 1 \quad \text{for all } u \in V, \\ & x_e \geq 0 \quad \text{for all } e \in E. \end{aligned} \tag{8.2}$$

Unlike the bipartite case, solving (8.2) when G is not bipartite does not give us a maximum weight matching. Consider the example given by Figure 8.1. In Figure 8.1, we have a triangle with every

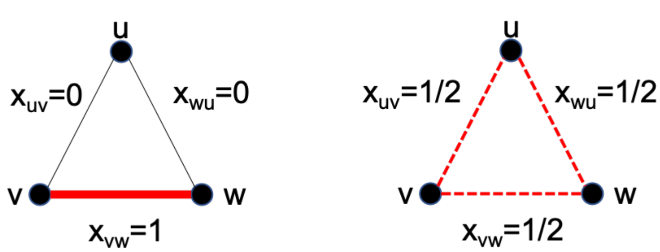

Figure 8.1: factionality of the linear programming relaxation

edge weight 1. Then the maximum weight of a matching is equals the maximum size of a matching, which is 1. However, setting $x_{uv} = x_{vw} = x_{wu} = 1/2$ produces a feasible solution to the LP relaxation, but its value is $3/2$, greater than 1. This example suggests that for a nonbipartite graph G, (8.2) is not equivalent to (8.1).

What we need to characterize is the notion of the **matching polytope**. The matching polytope of a graph G is formally defined as the convex hull of the incidence vectors of matchings in G, which is the set of solutions satisfying the constraints of (8.1). Hence, the matching polytope is given by

$$P_{\text{matching}}(G) = \text{conv}\left\{x \in \{0,1\}^{|E|} : \sum_{v \in V : uv \in E} x_{uv} \le 1 \quad \text{for all } u \in V\right\}.$$

We argued that the formulation (8.1) for the maximum weight bipartite matching problem is equivalent to

$$\max\left\{\sum_{e \in E} w_e x_e : \ x \in P_{\text{matching}}(G)\right\}. \tag{8.3}$$

Proposition 8.1. *Let $G = (V, E)$ be a bipartite graph. Then*

$$P_{\textit{matching}}(G) = \left\{x \in [0,1]^{|E|} : \sum_{v \in V : uv \in E} x_{uv} \le 1 \quad \textit{for all } u \in V\right\}.$$

Proof. Let $P_{\text{matching}}(G)$ denote the set on the right-hand side. Suppose for a contradiction that $P_{\text{matching}}(G) \neq LP_{\text{matching}}(G)$. Since $P_{\text{matching}}(G)$ is a subset of $LP_{\text{matching}}(G)$, it follows that there exists $\bar{x} \in LP_{\text{matching}}(G) \setminus P_{\text{matching}}(G)$. Since $P_{\text{matching}}(G)$ is closed and convex, the **separating hyperplane theorem** implies that there exists $w \in \mathbb{R}^{|E|}$ such that

$$\sum_{e \in E} w_e \bar{x}_e > \max\left\{\sum_{e \in E} w_e x_e : \ x \in P_{\text{matching}}(G)\right\}.$$

On the other hand, we have proved that solving the LP relaxation (8.2) finds a maximum weight matching when G is bipartite. This means that for any $w \in \mathbb{R}^{|E|}$, we have

$$\max\left\{\sum_{e \in E} w_e x_e : \ x \in P_{\text{matching}}(G)\right\} = \max\left\{\sum_{e \in E} w_e x_e : \ x \in LP_{\text{matching}}(G)\right\}.$$

This leads to a contradiction. Therefore, $P_{\text{matching}}(G) = LP_{\text{matching}}(G)$, as required. □

For a nonbipartite graph, the example in Figure 8.1 implies that the degree constraints are not enough to characterize the matching polytope. We next explain additional inequalities that are necessary to describe the matching polytope. Let $U \subseteq V$ be a subset of the vertex set with an odd number of vertices, as in Figure 8.2. Then look at the set of edges that are fully contained in U. Then the following inequality is satisfied by any solution to the integer program (8.1):

$$\sum_{e \in E(U)} x_e \le \frac{|U| - 1}{2}$$

where $E(U)$ is the set of edges fully contained in U. We call this inequality an **odd-set inequality**. Why is the odd-set inequality valid? Note that the left-hand side $\sum_{e \in E(U)} x_e$ counts the maximum number of edges from $E(U)$ a maching can take. Here, if a matching takes an edge in $E(U)$, then it covers two vertices in U. Note that $|U|$ is odd, and by parity, at least one vertex always remains unmatched. Equivalently, at most $|U| - 1$ vertices in U can be matched by a matching. Hence, $E(U)$ contains at most $(|U| - 1)/2$ edges in a matching.

Let us get back to the example in Figure 8.1. Note that the $U = \{u, v, w\}$ is an odd cardinality subset, and the corresponding odd-set inequality is $x_{uv} + x_{vw} + x_{wu} \le 1$. Hence, imposing the odd-set inequality, we may exclude the fractional solution $(x_{uv}, x_{vw}, x_{wu}) = (1/2, 1/2, 1/2)$.

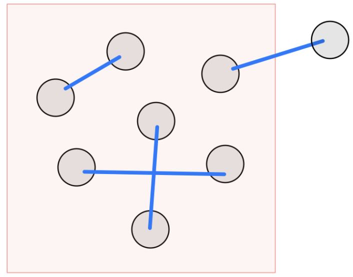

Figure 8.2: odd cardinality subset

Theorem 8.2 (Edmonds [Edm65]). *Let $G = (V, E)$ be a graph, not necessarily bipartite. Then*

$$P_{matching}(G) = \left\{ x \in [0,1]^{|E|} : \begin{array}{ll} \displaystyle\sum_{v \in V: uv \in E} x_{uv} \leq 1 & \text{for all } u \in V, \\ \displaystyle\sum_{e \in E(U)} x_e \leq \frac{|U|-1}{2} & \text{for all } U \subseteq V \text{ with } |U| \geq 3 \text{ odd} \end{array} \right\}.$$

2 Ellipsoid algorithm and its consequences in combinatorial optimization

In this section, we introduce the ellipsoid algorithm. The problem that we consider is as follows.

> Given a polyhedron $P = \{x \in \mathbb{R}^d : Ax \leq b\}$, (1) conclude that the interior of P is empty, or (2) find a point $\bar{x}$ contained in the interior of P.

This is a variant of the **feasibility problem**. The basic outline of the ellipsoid algorithm for the feasibility problem is as follows.

Algorithm 1 Ellipsoid algorithm

```
Initialize a polyhedron P = {x ∈ R^d : Ax ≤ b} and a sufficiently large ellipsoid E_1.
for t = 1, ..., T do
    if the center x^t of ellipsoid E_t is in the interior of P then
        Stop and conclude that P contains x^t.
    else
        There exists some inequality α^⊤x ≤ β in the system Ax ≤ b such that α^⊤x^t ≥ β.
        Let E_{t+1} be the smallest ellipsoid containing E_t ∩ {x ∈ R^d : α^⊤x ≤ β}.
        t → t + 1.
    end if
    Conclude that the interior of P is empty.
end for
```

Theorem 8.3 (Kachyan). *The ellipsoid algorithm (Algorithm 1) terminates with a correct answer if E_1 and T are properly chosen.*

In fact, Kachyan showed that one can choose E_1 and T so that their encoding sizes are polynomially bounded, in which case Algorithm 1 runs in polynomial time.

The important part is that the ellipsoid algorithm can be turned into a polynomial algorithm for the problem of optimizing a linear function over P. The idea is based on binary search. Basically, if we want to minimize a linear function $c^\top x$, then we consider

$$\left\{x \in \mathbb{R}^d : \ Ax \le b, \ c^\top x \le v\right\}$$

for varying v.

Next we formally state the **equivalence between optimization and separation**. Let $P \subseteq \mathbb{R}^d$ be a rational polytope such that

$$P = \operatorname{conv}\{v^1, \dots, v^n\}.$$

Then we say that $P \subseteq \mathbb{R}^d$ belongs to a **well-described family of rational polyhedra** if the length L of input needed to describe P satisfies $d \le L$ and $\log D$ is bounded by a polynomial function of L, where D is the largest numerator or denominator of the rational vectors v^k for $k \in [n]$ and $h \in [\ell]$. Here, we care about the number D to bound the complexity of the ellipsoid method.

1. **Separation Problem**
   Given a well-defined polyhedron $P \subseteq \mathbb{R}^d$ and $\bar{x} \in \mathbb{Q}^d$, either show that $\bar{x} \in P$ or find an inequality $\alpha^\top x \le \beta$ satisfied by all $x \in P$ such that $\alpha^\top \bar{x} > \beta$.

2. **Optimization Problem**
   Given a well-defined polyhedron $P \subseteq \mathbb{R}^d$ and $c \in \mathbb{Q}^d$, find x^* such that $c^\top x^* = \max\{c^\top x : x \in P\}$ or show that $P = \emptyset$.

Theorem 8.4 (Grötschel, Lovász, and Schrijver [GLS81])**.** *For a well-defined polyhedron P, the separation problem can be solved in polynomial time if and only if the optimization problem can be solved in polynomial time.*

By Theorem 8.2, we know that (8.1) is equivalent to

$$\max\left\{\sum_{e\in E} w_e x_e : \ x \in P_{\text{matching}}(G)\right\},$$

which is given by

$$\begin{aligned}
\text{maximize} \quad & \sum_{e\in E} w_e x_e \\
\text{subject to} \quad & \sum_{v\in V: uv\in E} x_{uv} \le 1 \quad \text{for all } u \in V, \\
& \sum_{e\in E(U)} x_e \le \frac{|U|-1}{2} \quad \text{for all } U \subseteq V \text{ with } |U| \ge 3 \text{ odd}, \\
& x_e \ge 0 \quad \text{for all } e \in E.
\end{aligned} \tag{8.4}$$

Although (8.4) is a linear program, one issue is that the number of odd cardinality subsets of V can be exponential in $|V|$. In that case, writing down all odd-set inequalities for (8.4) cannot be done in polynomial time. Nevertheless, the optimization problem (8.4) is shown to be solvable in polynomial time, based on the equivalence between separation and optimization.

To show that (8.4) can be solved in polynomial time, by Theorem 8.4, it suffices to show that the separation problem over the matching polytope $P_{\text{matching}}(G)$ can be solved in polynomial time. Given $\bar{x} \in \mathbb{Q}^{|E|}$, we want to decide whether $\bar{x} \in P_{\text{matching}}(G)$ or find an inequality $\alpha^\top x \leq \beta$ that separates $\bar{x}$ from $P_{\text{matching}}(G)$. For the matching polytope, we can check whether $\bar{x}$ satisfies the degree constraints and the nonnegativity constraints in $O(|V| + |E|)$ time. Hence, the question is as to whether we can decide that $\bar{x}$ satisfies the odd-set inequalities in polynomial time. In fact, the separation problem can be solved in polynomial time with its connection to the so-called **minimum odd cut problem**.

References

[Edm65] J. Edmonds. Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research National Bureau of Standards Section B*, 69:125–130, 1965. 8.2

[GLS81] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981. 8.4