Fall 2016, MATH-566

Minimum-Weight Perfect Matching for Point in Plane

Source: Bill,Bill,Bill

Goal: Find a minimum-weight perfect matching algorithm that works for points in the plane. Plan is to find a relaxation of the integer program and guide us to build an algorithm by cleverly interpreting the dual solution.

First we write a good linear program for general matching problem.

Recall that Minimum-weight perfect matching problem can be formulated as an integer programming problem in the following way

$$(IP)\begin{cases}\text{minimize} & \sum_{e\in E} c(e)x_e\\ \text{subject to} & \sum_{e\in\delta(v)} x_e = 1 \text{ for all } v\in V\\ & \mathbf{x}\in\{0,1\}^{|E|},\end{cases}$$

1: Show that a relaxation of (IP) to linear program may result in optimal solution that is not realizable by a perfect matching. You need to cleverly assign weights!

2: Write a better program (P) that prevents issue from the previous figure and its dual (D). (Hint: Let $\mathcal{C}$ be the set of all odd cuts as set of edges and use $\mathcal{C}$.

Theorem Edmonds: G has a perfect matching iff (P) has a feasible solution. Moreover, the minimum weight of the perfect matching is equal to the value of optimal solution to (P).

3: Write complementary slackness conditions for (P).

A family of sets $\mathcal{A}$ is *nested* if for any $A, B \in \mathcal{A}$ exactly one of $A \cap B = \emptyset$, $A \subseteq B$, and $B \subseteq A$ holds.

A solution to (D) is *nested* if the family of Ds corresponding to $Y_D > 0$ is nested.

Theorem 5.17 If an optimal solution to (D) exists, then there exists a nested one.

If c satisfies triangle-inequality, then the dual has a nice solution.

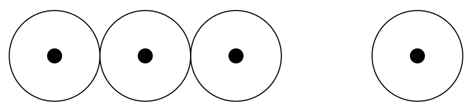
Theorem 5.20 Let G be a complete graph having even number of nodes and $c \geq 0$ satisfy the triangle inequality, then there exists an optimal solution to the dual with $y \geq 0$.

Finally, we are ready for some geometric ideas behind the algorithm that is on the cover of the textbook!

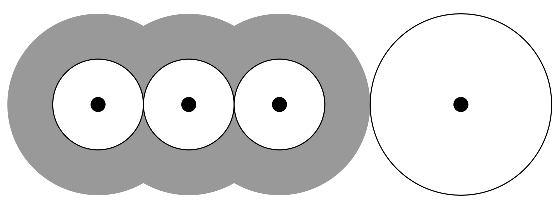
Let V be points in the plane and let $c : uv \to \mathbb{R}^+$ the Euclidean distance of u and v. Find a perfect matching M that is minimizing the sum of costs of the edges in the matching. Let E be the set of all pairs of vertices.

Notice this problem satisfies the triangle inequality.

4: Suppose $|V| = n$ and every point has a disk of radius 1. Assume these disks are disjoint. Can you give a lower bound on the cost of a perfect matching?



5: Consider the following extension, where big white disk has radius 2 and the gray area is actually union of three disk of radius 2 without white disk of radius one. Can you provide a lower bound on the cost of a perfect matching?

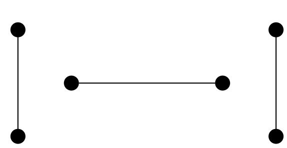


We call the white disks around vertices *control zones*. A pair of compact sets (N, I) is a *moat* if

$$I \subset N, |I \cap V| \text{ is odd, and } N \setminus \text{ interior}(I) \text{ contains no points in } V.$$

Example of a moat is the pair of gray set as N and the three white disks inside as I.

Notice we could interpret y_v as a radius of the control zone and Y_D as a width of a moat. This is a possible interpretation of a dual solution to D - try example, where you find optimal matching and corresponding control zones and moates.



Goal: Algorithm that finds a perfect matching of cost at most twice of the optimum solution.

Algorithm outline:
1) Obtain a forest F, where every component has even number of vertices. So called *even forest*.
2) Transform the forest F into a perfect matching M such that the cost of used edges does not increase.

In order to provide a bound on the cost of the resulting matching M, we build a forest $F^\star$ whose cost would be at most twice the lower bound obtained from control zones and moats.

Definition Let C be a set. Define

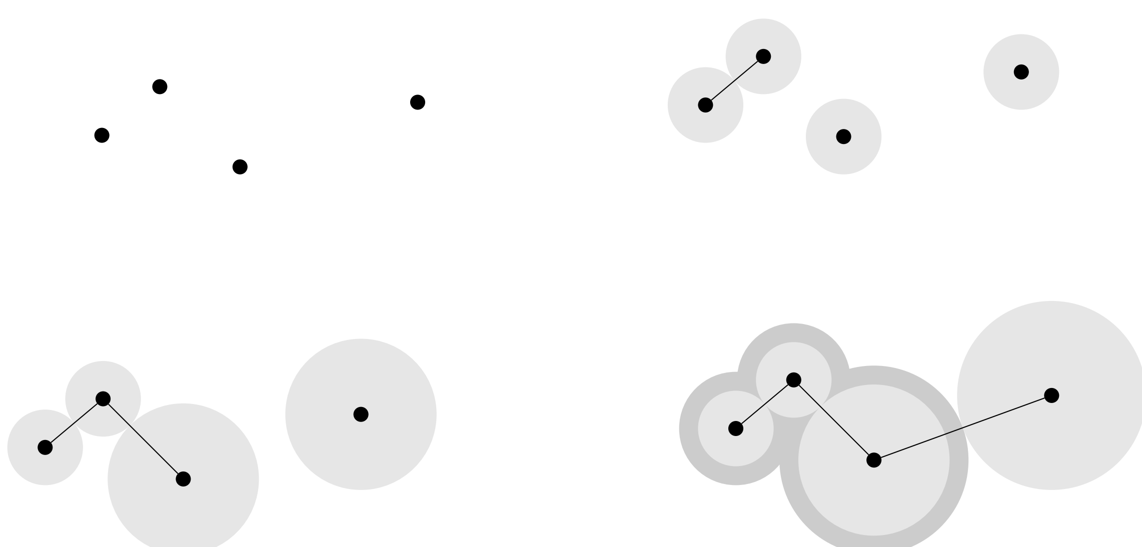
$$parity(C) = \begin{cases} 0 & \text{if } |C| \text{ even} \\ 1 & \text{if } |C| \text{ odd.} \end{cases}$$

Let $\overline{c}_e = c(uv) - y_u + y_v + \sum_{uv \in D \in \mathcal{C}} Y_D$. (Slack in the constraints in D.)

Goemans-Williamson algorithm (sketch)
Goal: Find an even forest F and a feasible solution to dual program (D).

1. $\mathcal{C} = \{\{v\} : v \in V\}$; $F = \emptyset$; $y = 0$; $Y = 0$
2. while exists $C \in \mathcal{C}$ with $|C|$ odd
3. Find an edge $e = uv$, with $u \in C_i$ and $v \in C_j$, $C_i \neq C_j$, that minimizes $\varepsilon = \overline{c}_e/(parity(C_i) + parity(C_j))$. Notice at least one of $|C_i|$ and $|C_j|$ is odd.
4. For all $C \in \mathcal{C}$ where $C = \{x\}$, add ε to y_x.
5. For all $C \in \mathcal{C}$ where $|C| > 1$ and $|C|$ is odd, add ε to Y_C.
6. add e to F and replace C_i and C_j by $C_i \cup C_j$.

Example:



6: Try the algorithm on the following example:

7: Show that there exists a matching M such that $\sum_{e \in M} c(e) \leq \sum_{e \in F} c(e)$.

More precise way of computing the cost of the resulting lower bound LB on the matching:

8: Show that

$$LB = \sum_k \varepsilon^k |\{C \in \mathcal{C}^k : |C| \text{ odd }\}|$$

where $\mathcal{C}^k$ and ε^k are from the kth iterations of the algorithm.

9: Show that the forest $F^\star$ obtained from F by *dropping even edges* has cost at most $2\left(\sum_v y_v + \sum_D Y_D\right)$.
Cost of $F^\star$ is $\sum_{e \in F^\star} c(e)$.
More precisely, consider every edge $e \in F^\star$ of the forest and decompose its cost into small pieces.

$$c_e^k = \begin{cases} 0 & \text{if } C_i = C_j \\ \varepsilon^k(parity(C_i) + parity(C_j)) & \text{otherwise} \end{cases}$$

Now

$$c(e) = \sum_k c_e^k$$

Goal is to show for all k

$$\sum_e c_e^k \leq 2\varepsilon^k |\{c \in \mathcal{C}^k : |C| \text{ odd }\}|,$$

which gives

$$cost(M) \leq cost(F^\star) = \sum_k \sum_e c_e^k \leq 2 \sum_k \varepsilon^k |\{C \in \mathcal{C}^k : |C| \text{ odd }\}| = 2LB.$$