Minimum-Weight Perfect Matching for Points in Plane

Source: Bill,Bill,Bill, Chapter 5.6, page 192+

Let $G = (V, E)$ be a graph. Let $c : E \to \mathbb{R}^+$ is a cost. Find a perfect matching M that is minimizing the sum of costs of the edges in the matching.

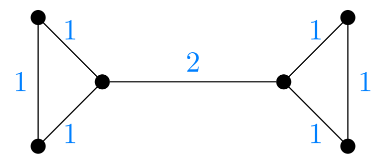
Goal: Find a minimum-weight perfect matching algorithm that works for points in the plane. The plan is to find a relaxation of the integer program and guide us to build an algorithm by cleverly interpreting the dual solution. There is an minimum-weight perfect matching algorithm for any graph but the one for points in plane is somewhat fun to see and there is a cool looking interpretation of the dual.

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 a linear program may result in optimal solution that is not realizable by a perfect matching. You need to cleverly assign weights!



Solution: Optimal solution to the relaxation is $\frac{1}{2}$ on each edge of the triangle and 0 on the remaining edge. This gives weight 3. The minimum-weight perfect matching has weight 4.

2: Write a better program (P) that prevents issue from the previous graph and its dual (D).
(Hint: Trouble is a vertex set of an odd size. Let $\mathcal{C}$ be the set of all cuts in a graph G that have odd number of vertices on each side and are NOT cuts just around one vertex. Assume the cuts are represented as a set of edges and use $\mathcal{C}$ to add more constraints to (IP).)

Solution: We include constraint that for every odd cut, the edges going across sum to at least one. A cut is *odd* if it contains odd number of vertices on each side. Let $\mathcal{C}$ be the set of all odd cuts, where we consider as a cut the set of edges.

$$(P)\begin{cases}\text{minimize} & \sum_{e\in E} c(e)x_e\\ \text{subject to} & \sum_{e\in\delta(v)} x_e = 1 \text{ for all } v \in V\\ & \sum_{e\in D} x_e \geq 1 \text{ for all } D \in \mathcal{C}\\ & x_e \geq 0 \text{ for all } e \in E\end{cases}$$

Notice that the bad solution from the previous relaxation is gone.

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 an optimal solution to (P).

3: Write a dual to the program (P).

$$(P)\begin{cases}\text{minimize} & \sum_{e\in E} c(e)x_e\\ \text{subject to} & \sum_{e\in\delta(v)} x_e = 1 \text{ for all } v\in V\\ & \sum_{e\in D} x_e \geq 1 \text{ for all } D\in\mathcal{C}\\ & x_e\geq 0 \text{ for all } e\in E\end{cases}$$

Solution:

$$(D)\begin{cases}\text{maximize} & \sum_{v\in V} y_v + \sum_{D\in\mathcal{C}} Y_D\\ \text{subject to} & y_u + y_v + \sum_{uv\in D\in\mathcal{C}} Y_D \leq c(uv) \text{ for all } uv\in E\\ & y_v\in\mathbb{R} \text{ for all } v\in V\\ & Y_D\geq 0 \text{ for all } D\in\mathcal{C}\end{cases}$$

4: Write complementary slackness conditions for (P) and (D).

Solution:

$x_e > 0$ implies $y_e + y_e + \sum_{e\in D\in\mathcal{C}} Y_D = c(e)$
$Y_D > 0$ implies $\sum_{e\in D} x_e = 1$ for all D

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

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.

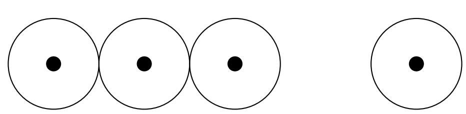
Theorem 5.20 Let G be a complete graph having even number of nodes and $c\geq 0$ satisfy the triangle inequality, then there exists an optimal solution to the dual with $y\geq 0$.

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

Problem: Let V be a set of 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 weights of the edges in the matching. Let E be the set of all pairs of vertices.

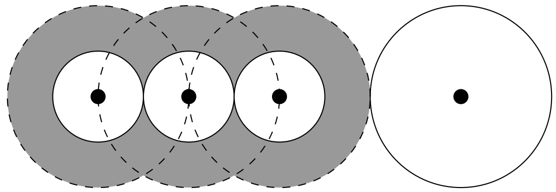
Notice this problem satisfies the triangle inequality.

5: Suppose there are n (even number) points in the plane and there is a disk of radius 1 at each of these points. Assume these disks are disjoint. Can you give a lower bound on the cost of a perfect matching?



Solution: The lower bound is 4. In general, the lower bound is n as for every edge in the matching, it must go trough a radius of two circles.

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

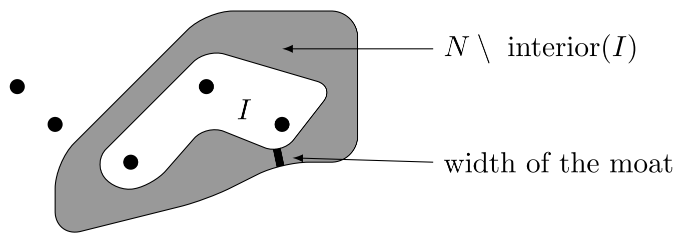


Solution: Lower bound would be 6. This comes from $1 + 1 + 1 + 2$ for the radius of the disks. Then additional $+1$ comes from the fact that they gray area encloses an odd number of points. So at least one edge of the perfect matching must pass through the gray area.

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

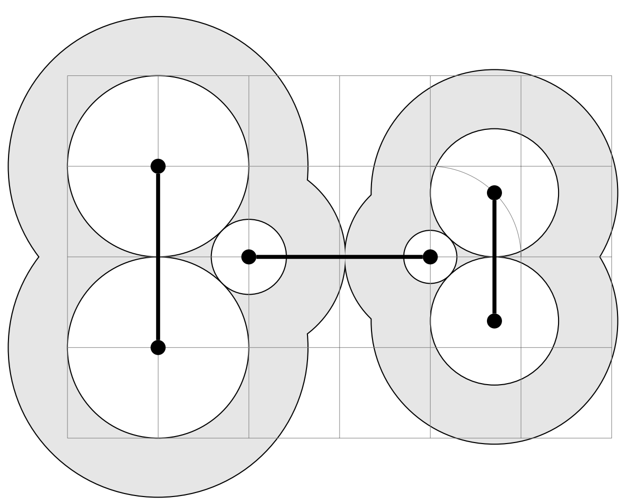
$$I \subset N, |I \cap V| \text{ is odd, and } N \setminus \text{ interior}(I) \text{ contains no points in } V.$$

Example of a moat showing I that contains 3 vertices and N is I and the gray area. The exercise above also contains a moat, where three white discs are I and adding the gray area makes N.



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.

7: In the following example, find a minimum-weight perfect matching, where weights are given by the Euclidean distances, and find the corresponding control zones and moats.



Solution: The radiuses of control zones are 1, $\sqrt{2} - 1$, $1 - \sqrt{0.5}$, and $\sqrt{0.5}$ from left to right. Both moths have width $(2 - \sqrt{2} + \sqrt{1/2})/2$. Sum of all is $4 + 2\sqrt{0.5}$.

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

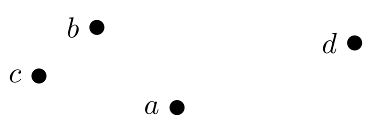
$$parity(C) = \begin{cases} 0 & \text{if } |C| \text{ even} \\ 1 & \text{if } |C| \text{ odd.} \end{cases}$$

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

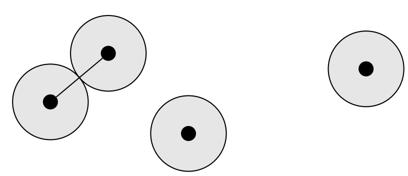
Goemans-Williamson algorithm (sketch)
Goal: Find an even forest F and a feasible solution to dual program (D).

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 $C_i \cup C_j$.

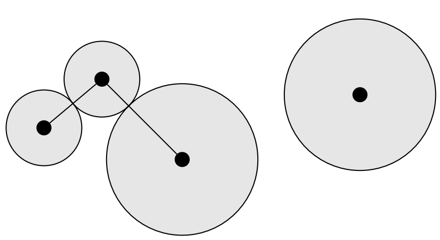
Example:



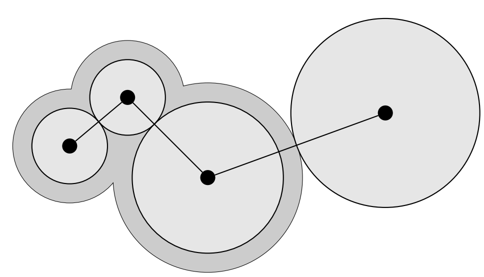
$\mathcal{C} = \{\{a\}, \{b\}, \{c\}, \{d\}\}$



$\mathcal{C} = \{\{a\}, \{b, c\}, \{d\}\}$
$y_a = 1, y_b = 1, y_c = 1, y_d = 1$

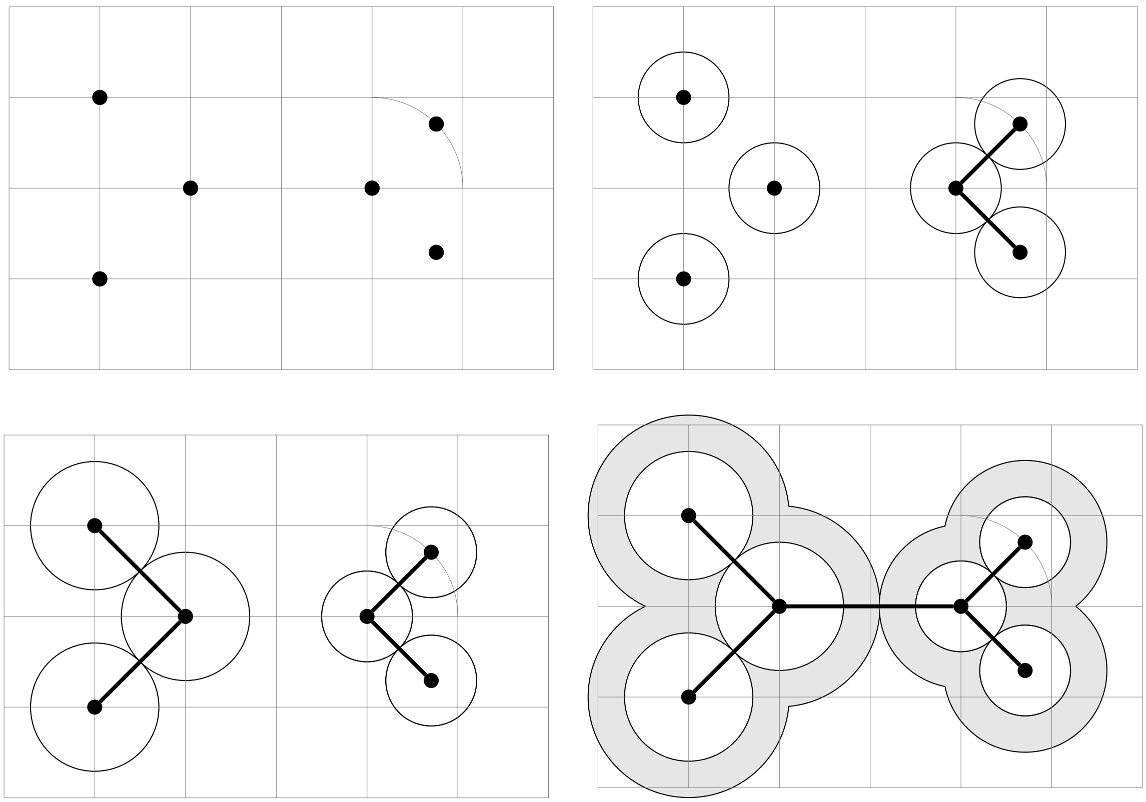


$\mathcal{C} = \{\{a, b, c\}, \{d\}\}$
$y_a = 2, y_b = 1, y_c = 1, y_d = 2$



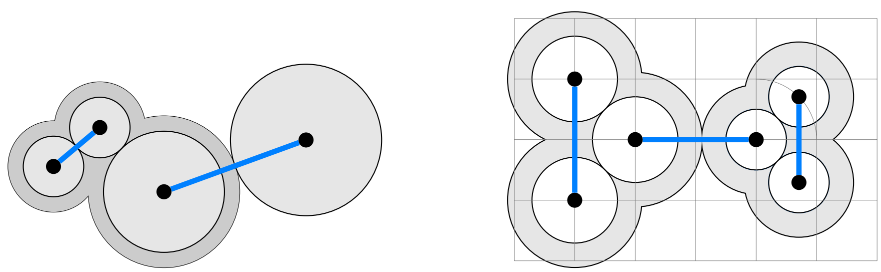
$\mathcal{C} = \{\{a, b, c, d\}\}$
$y_a = 2, y_b = 1, y_c = 1, y_d = 2.5, Y_{abc} = 0.5$

8: Try the algorithm on the following set of six points.



9: For the previous two examples, construct matchings M such that $\sum_{e \in M} c(e) \leq \sum_{e \in F} c(e)$. Hint: No need to use edges only from M.

Solution: We remove edge on the left and use triangle inequality on the right.



10: Assuming c satisfies the triangle inequality, show that for any even forest F, there exists a matching M such that

$$\sum_{e \in M} c(e) \leq \sum_{e \in F} c(e).$$

What operations have you used above?

Solution: We provide an algorithmic way and use two steps to find a matching.
(a) If there are is an edge e in F such that $F - e$ is still an even forest, remove e.
(b) If v has two neighbors x and y then remove edges vx, vy and add the edge xy. By triangle inequality, this does not increase the cost and still leaves the graph as an even forest. (removing any edge leaves odd number of vertices on each side).

Let F be an even forest. And edge in F is *even* if $F - e$ is also an even forest.

11: Let F be an even forest. Consider only the following two operations.
(a) Remove all even edge, and denote the result by $F^\star$.
(b) Let v have two neighbors x and y that are leaves. Remove edges vx, vy and add the edge xy.
Show that if none of these two operations can be applied, the resulting graph is a perfect matching.
Hint: Vertices of degree 2?

Solution: (a) If there are is an edge e in F such that $F - e$ is still an even forest, remove e.
Now observe that there are no vertices of degree 2. If v is a vertex of degree 2 in F, then one of its adjacent edges is an even edge and can be removed. Now operations:
(b) Otherwise consider F', which is F without leaves. Let v be a leaf in F'. Since the edge of F' incident with v is not removed in (a), v must have at least two pendant leaves x and y in F that were removed. Replace edges xv and yv by edge xy.
Notice none of these operations increase the cost of F and if none is applicable, F is a

We have shown how to construct a matching from an even forest. Now we need to show that the even forest from Goemans-Williamson algorithm has cost at most twice the weigh of a minimum-weight perfect matching.

Define $cost(F) = \sum_{e \in F} c(e)$.

Notice that Goemans-Williamson calculates a feasible solution to the following program

$$(D) \begin{cases} \text{maximize} & \sum_{v \in V} y_v + \sum_{D \in \mathcal{C}} Y_D \\ \text{subject to} & y_u + y_v + \sum_{uv \in D \in \mathcal{C}} Y_D \le c(uv) \text{ for all } uv \in E \\ & y_v \in \mathbb{R} \text{ for all } v \in V \\ & Y_D \ge 0 \text{ for all } D \in \mathcal{C} \end{cases}$$

which is the dual of (P) as we seen in one of the early exercise. Hence it provides a lower bound to the minimum cost of a perfect matching.

Theorem: Let LB be a lower bound obtain by the algorithm. Then perfect matching M obtained from $F^\star$ satisfies

$$LB \le cost(M) \le cost(F^\star) \le 2LB.$$

The argument is on the next page but it is too technical to be done in the class.

12: 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 is a lower bound on the cost of minimum-weight perfect matching M, i.e. $LB \le cost(M)$.

Solution: This exactly corresponds to how much each control zone and moat grows in each iterations. Notice that we are growing only control zones and moats that are *odd* so the added distance(s) to radiuses must be all crossed by matching edges. By the trivial observation that odd number of vertices do not have a perfect matching.

13: Let the forest $F^\star$ obtained from F by *dropping even edges*. Show that

$$cost(F^\star) \le 2\left(\sum_v y_v + \sum_D Y_D\right).$$

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

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

The goal is to show for all k

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

which gives

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

Solution: Look at each of the k steps separately. Our goal is to show that in each of them, we are not adding too many to weight to edges in $F^\star$.

Contract all sets of $\mathcal{C}^k$ into single vertices and use edges of $F^\star$, no loops. This creates an auxiliary graph H, which is also a forest. Partition $V(H)$ into V_{ODD} and V_{EVEN} based on the parity of sets in $\mathcal{C}^k$.

Goal is to show

$$\sum_{v \in V_{ODD}} deg(v) \le 2|V_{ODD}|.$$

Since $F^\star$ is an even forest, vertices in V_{EVEN} are NOT leaves in H, hence they have degree at least 2 in H.

$$\sum_{v \in V(H)} deg(v) \le 2|V(H)| = 2|V_{EVEN}| + 2|V_{ODD}|$$

When combined this gives

$$\sum_{v \in V_{ODD}} deg(v) = \sum_{v \in V(H)} deg(v) - \sum_{v \in V_{EVEN}} deg(v) \le (2|V_{EVEN}| + 2|V_{ODD}|) - 2|V_{EVEN}| = 2|V_{ODD}|.$$