University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
M. Ahmadi, P. Schneider

Algorithms Theory
Sample Solution Exercise Sheet 5

Due: Monday, 7th of January, 2018, 14:15 pm

Exercise 1: Pinning Paper Polygons *(4+3 Points)*

You have two rectangular sheets of paper of equal dimensions. On each sheet an adversary has drawn straight lines that form a subdivision of each sheet into n polygons such that each polygon covers an equal area. The subdivision is different for each sheet. You also receive n pins.

(a) You put *two* of the sheets directly on top of each other. Prove that your pins suffice to pierce *all* polygons on both sheets (no folding or other funny business, polygons need to be pinned through their *interior*).

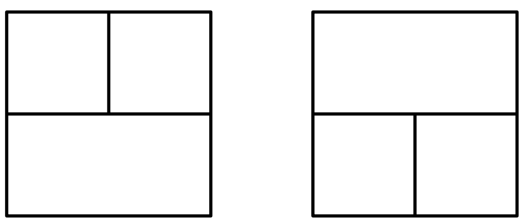
(b) Show that this is not possible if we drop the condition that each polygon covers an equal area on each sheet (however, the area of each polygon is bigger than zero).

Sample Solution

(a) Our goal is to use Hall's Theorem to show the claim. We interpret the polygons on each sheet as sets of nodes U and V respectively. We construct a bipartite graph $(U \cup V, E)$ where we have an edge between $u \in U$ and $v \in V$, iff the interior of two polygons u, v overlap. An edge between u and v means that one pin can be used to pierce both u and v. Since we have exactly n pins the claim is true if and only if $(U \cup V, E)$ has a perfect matching. We already know $|U| = |V| = n$.

W.l.o.g. we assume the area of each sheet is n so that each polygon has area exactly 1. Let $U' \subseteq U$. Then U' covers an area of exactly $|U'|$. The set $N(U') \subseteq V$ are the polygons that have an intersection with some $u \in U'$. The polygons in $N(U')$ cover at least the area that the polygons in U' cover. If we assume $|N(U')| < |U'|$ then this would imply that each $v \in N(U')$ would cover an area of $|U'|/|N(U')| > 1$, a contradiction. Thus we have $|N(U')| \geq |U'|$ and the claim follows from Hall's Theorem.

(b) We give an example where $|N(U')| < |U'|$. Due to Hall's Theorem we can not have a perfect matching for this example. Since a perfect matching is a necessary condition on order to pin all polygons with n pins, the constructed example requires more than n pins to pierce all polygons through their interior (at least 4 in this example).



Exercise 2: Doomsday on Krypton *(6+2 Points)*

Horrible news on Krypton: The planet will be struck by a meteorite within the next m minutes and the planet, all of its n cities and all k Kryptonions inhabiting them will be destroyed. Fortunately the Kryptonian government provided interstellar escape pods for such an emergency.

In city i, where k_i Kryptonians live ($\sum_{i=1}^{n} k_i = k$), there are p_i such pods available. Each pod can carry one person. Kryptonians can either *instantly* use one of the escape pods in the city where they live, or travel to another city and use an escape pod there. It takes d_{ij} minutes to travel from city i to city j. A pod must be reached before the meteorite destroys Krypton.

Due to flight safety concerns the total number of pods that can launch from certain subsets of cities is restricted. I.e., there are subsets $S_i \subseteq [1..n]$ with $i \in [1..\ell], \ell \leq n$ and parameters r_i. Each city j is subject to exactly one flight restriction zone (that is $j \in S_i$). All cities that are part of S_i can not launch more than r_i pods taken together.

(a) You need to determine the maximum number of survivors. Describe how this problem can be reduced to a maximum flow problem.

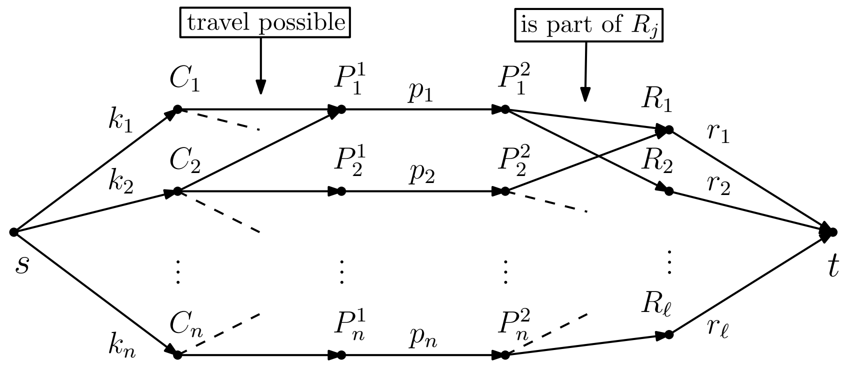
(b) Making no assumptions about k, ℓ, n, m other than the ones given in the exercise, how long does it take in the worst case to solve the maximum flow problem using the Ford-Fulkerson algorithm?

Sample Solution

(a) We define a flow network as follows. First we create a set of nodes $C_1, \ldots, C_n$ for the cities and connect the source s with each city, whereas each of these edges has capacity $k_1, \ldots, k_n$, corresponding to the number of inhabitants of the respective city.

Now we create another set of nodes $P_1^1, \ldots, P_n^1, P_1^2, \ldots, P_n^2$ for the escape pods. We establish an edge from P_j^1 to P_j^2 for each $1 \leq j \leq n$, and assign it capacity p_j (representing the number of pods available in city j). Then we create an edge from C_i to P_j^1, iff $d_{i,j} \leq m$ (i.e. if the time suffices to travel from city i to a pod in city j). These edges are not restricted in capacity, so we assign capacity k to each.

Finally we have to take care of the flight restriction zones. We create nodes $R_1, \ldots, R_\ell$ and create an unrestricted edge from P_j^2 to R_i, iff $j \in S_i$. We create an edge from each R_i to the sink and assign these edges the capacity r_i (to restrict the number of pods that can launch from the respective cities).



(b) The flow is restricted by k and the number of edges is $\mathcal{O}(n^2)$ ($\Theta(n^2)$ in case the time suffices to travel from each city to each other city). Thus the Ford-Fulkerson algorithm takes $\mathcal{O}(kn^2)$ time to solve the maximum flow problem.

Exercise 3: Sinkless Orientation *(5+5+3 Points)*

(a) Let $B = (U \cup V, E)$ be a bipartite graph and assume that for every $A \subseteq U$, we have $|N(A)| \geq |A|$. Show that this implies that there exists a matching of size $|U|$ in B (i.e., a matching that matches every node in U).

(b) Let $G = (V, E)$ be a graph with minimum degree at least two (i.e., each node has at least two incident edges). Use the prior statement to show that there is a way to orient the edges of G such that each node of G has at least one out-going edge (this is known as a sinkless orientation).

Hint: Sinkless orientation can be seen as matching nodes and edges.

(c) Let us now assume that the graph G has minimum degree 4. Show that there exists an orientation in which each node has at least one in-coming and at least one out-going edge.

Sample Solution

(a) We have the following equivalence (contra-position):

$$\begin{aligned} &\Big[\forall A \subseteq U : |N(A)| \geq |A| \Rightarrow B \text{ has a matching of size } |U|\Big] \\ \text{if and only if} \quad &\Big[B \text{ has } \textbf{no} \text{ matching of size } |U| \Rightarrow \exists A \subseteq U : |N(A)| < |A|\Big] \end{aligned}$$

Therefore it suffices to prove the second line. Consider a bipartite graph B with *no* matching of size $|U|$. We construct a flow network from B (as we have seen in the lecture), by connecting a source s to all nodes in U and all nodes in V to a sink t, directing all edges from s to t and setting all edge capacities to one.

Since a maximum flow implies a matching of the same size, the maximum flow in this network must also be smaller than $|U|$. Due to the max-flow-min-cut theorem, the minimum cut $C = (A', B')$ (with $A', B' \subseteq U \cup V \cup \{s, t\}$) has size smaller than $|U|$ as well. For brevity, let $|C|$ denote the size of $C = (A', B')$ (number of edges from A' to B'), i.e., $|C| < |U|$.

W.l.o.g. let A' be the set that contains s. We know A' can not be empty nor can it encompass all nodes (by definition of a cut). Further we know that $A' \neq \{s\}$ and $A' \neq U \cup V \cup \{s\}$ (since those cuts would have size $|U|$ and would thus not be minimum). Therefore $A := A' \cap U$ is not empty.

Let x be the number of edges from s to B'. Since by construction we have one edge from s to each node in U we know that $x = |U \setminus A| = |U| - |A|$ $(*)$. Let z be the number of edges from A to B'. Let y be the number of edges from $A' \setminus (A \cup s) = A' \cap V$ to B'. Then $y = |A' \cap V|$ since by construction each node in V has exactly one edge to the sink t.

Taken together x, y, z represent the size of the cut C, i.e., $x + y + z = |C| < |U|$ $(**)$. Further we know that $|N(A)| \leq y + z$ $(***)$ since $N(A)$ can not contain more nodes then the edges going from A to B' (which is z), plus the nodes in $A' \cap V$ (which is y). We obtain

$$|N(A)| \overset{(***)}{\leq} y + z \overset{(**)}{<} |U| - x \overset{(*)}{=} |A|$$

(b) One can set up the problem of finding such an orientation as a bipartite matching problem as follows. The bipartite graph has a node v for each node of G (let's call this L) and it has a node e for each edge of G (let's call this R). Nodes v and e are connected in the bipartite graph $(L \cup R, E')$, if v is a node of the edge e in G. A matching in this bipartite graph is a pairing of nodes and edges in G such that each edge of G is assigned to at most one of its two nodes.

If the matching contains (v, e), node v in G orients the edge e as an out-going edge for itself. In order to guarantee the existence of a sinkless orientation, we thus need to show that the constructed bipartite graph has a matching that matches each node v. Note that each node in L has degree at least 2 (because G has minimum degree at least 2) and each node in R has degree exactly 2 (because an edge of G has exactly two incident nodes). Therefore, for a given subset $A \subseteq L$, we have at least $2|A|$ outgoing edges from A to R. And since each node in R has degree two, the $2|A|$ outgoing edges of A go to at least $|A|$ different nodes in R. Hence $|N(A)| \geq |A|$ for every $A \subseteq L$.

(c) After constructing the bipartite graph as before, we split each node $v \in L$ into two nodes v_1 and v_2, which each get at least two of the edges of v (we partition the edges evenly among v_1 and v_2). Minimum degree 4 implies that each node in L in this new bipartite graph has minimum degree 2 and we can thus compute a bipartite matching which matches each $v \in L$. Each node v now has two edges, which it can orient freely.

Exercise 4: Triangles in Random Graphs *(2+2+3+5 Points)*

Given a fixed vertex set $V = \{v_1, v_2, \ldots, v_n\}$ with n being an even number. Then the following (randomized) process defines the (undirected) random graph $G_p = (V, E_p)$.

For each vertex pair $\{v_i, v_j\}, i \neq j$ we independently decide with probability p whether the edge defined by this pair is part of the graph, i.e., whether $\{v_i, v_j\}$ is an element of the edge set E_p.

Furthermore we say that a subset $T = \{v_i, v_j, v_k\}$ of V of size 3 is a triangle of a graph, if all three edges $\{v_i, v_j\}$, $\{v_i, v_k\}$, $\{v_j, v_k\}$ are in the edge set of the graph.

(a) Let Z be the random variable that counts the number of edges in G_p. What is the distribution of Z? What is the probability that Z has value k, for some k?

(b) Calculate m_T, the number of all triangles that could *possibly* occur in G_p.

(c) Let X denote the number of triangles in G_p. Calculate $\mathbb{E}[X]$.

The generation of the random graphs is now changed as follows. Before edges are determined each vertex is colored either red or green; we let K be the random variable that counts the number of red vertices. Between two red vertices there is an edge with probability p_{rr}, between two green vertices with probability p_{gg} and between vertices of different color with probability p_{rg} (edges are still picked independently).

(d) Assume first that with probability $\frac{1}{7}$ all vertices are red, with probability $\frac{2}{7}$ all vertices are green and with probability $\frac{4}{7}$ each vertex independently gets color red or green with probability $1/2$ each. Also $p_{rr} = 1$, $p_{rg} = \frac{1}{\sqrt{3}}$ and $p_{gg} = 0$. Calculate $\mathbb{E}[X]$ under these conditions!

Sample Solution

(a) To construct the n-node random graph, there are $\binom{n}{2}$ trials corresponding to $\binom{n}{2}$ pairs of vertices. In each trial the edge between one pair of nodes is picked independently with probability p. Hence, Z follows the binomial distribution with parameters $\binom{n}{2}$ and p; that is $Z \sim Bin\left(\binom{n}{2}, p\right)$.

Therefore, the probability of getting exactly k successes in $\binom{n}{2}$ trials is:

$$\Pr(Z = k) = \binom{\binom{n}{2}}{k} p^k (1-p)^{\binom{n}{2}-k}$$

(b) Any set of 3 distinct nodes from the vertex set has non-zero probability to form a triangle in the graph. In other words, any set of 3 distinct nodes can possibly form a triangle. The number of these subsets is $\binom{n}{3}$. Hence, $m_T = \binom{n}{3}$.

(c) Any triple of (distinct) vertices in the graph can form a triangle. Consider all the $\binom{n}{3}$ triples of vertices in any order. Consider the random variable X_i for the i^{th} triple as follows:

$$X_i = \begin{cases} 1, & \text{if the } i^{th} \text{ triple forms a triangle} \\ 0, & \text{otherwise.} \end{cases}$$

Since the probability of any triple to form a triangle is p^3 (recall that probability of one specific edge being in E_p is p),

$$\mathbb{E}[X_i] = 1 \cdot \Pr(X_i = 1) + 0 \cdot \Pr(X_i = 0) = p^3$$

We can define the random variable X which represents the number of triangles in G_p, that is,

$$X = \sum_{i=1}^{\binom{n}{3}} X_i \, .$$

Therefore, using the linearity of expectation we have

$$\mathbb{E}[X] = \mathbb{E}\left[\sum_{i=1}^{\binom{n}{3}} X_i\right] = \sum_{i=1}^{\binom{n}{3}} \mathbb{E}[X_i] = \binom{n}{3} p^3 \, .$$

(d) Let us define the following events:

$\mathcal{A}$: is the event when all the vertices are red.
$\mathcal{B}$: is the event when all the vertices are green.
$\mathcal{C}$: is the event when each vertex gets color red or green with probability 1/2.

$$\Pr(\mathcal{A}) = \frac{1}{7}, \;\; \Pr(\mathcal{B}) = \frac{2}{7}, \;\; \Pr(\mathcal{C}) = \frac{4}{7}$$

Then,

$$\begin{aligned}
\mathbb{E}[X] &= \mathbb{E}\left[\sum_{i=1}^{\binom{n}{3}} X_i\right] = \sum_{i=1}^{\binom{n}{3}} \mathbb{E}[X_i] \qquad \text{[from the linearity of expectation]} \\
&= \sum_{i=1}^{\binom{n}{3}} \Pr(X_i = 1) \\
&= \sum_{i=1}^{\binom{n}{3}} \Big(\Pr(X_i = 1|\mathcal{A}) \Pr(\mathcal{A}) + \Pr(X_i = 1|\mathcal{B}) \Pr(\mathcal{B}) + \Pr(X_i = 1|\mathcal{C}) \Pr(\mathcal{C})\Big) \quad \text{[law of total probability]} \\
&= \sum_{i=1}^{\binom{n}{3}} \left(p_{rr}^3 \Pr(\mathcal{A}) + p_{gg}^3 \Pr(\mathcal{B}) + \Pr(X_i = 1|\mathcal{C}) \Pr(\mathcal{C})\right) \\
&\overset{**}{=} \sum_{i=1}^{\binom{n}{3}} \left(1 \cdot \left(\frac{1}{7}\right) + 0 \cdot \left(\frac{2}{7}\right) + \left(\frac{1}{8} \cdot 1^3 + \frac{1}{8} \cdot 0 + \frac{3}{8} \cdot 1 \cdot \frac{1}{\sqrt{3}} \cdot \frac{1}{\sqrt{3}} + \frac{3}{8} \cdot 0 \cdot \frac{1}{\sqrt{3}} \cdot \frac{1}{\sqrt{3}}\right) \cdot \frac{4}{7}\right) \\
&= \binom{n}{3} \cdot \left(\frac{1}{7} + \frac{2}{8} \cdot \frac{4}{7}\right) = \frac{2}{7} \cdot \binom{n}{3}
\end{aligned}$$

** For calculating $\Pr(X_i = 1|\mathcal{C})$ we apply the law of total probability on the color of the three nodes in the i^{th} triple of vertices. Note that the probability of having the three vertices all red or all green is 1/8 each, having two red vertices and one green vertex is 3/8, and one red vertex and two green vertices is 3/8.