Mathematical Foundations of Computer Science

CS 499, Shanghai Jiaotong University, Dominik Scheder

7 The Graph Score Theorem

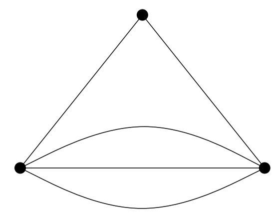
- Homework assignment published on Monday, 2018-04-09.
- Submit questions and first solution by Sunday, 2018-04-15, 12:00 by email to dominik.scheder@gmail.com and the TAs.
- You will receive feedback by Wednesday, 2018-04-18.
- Submit your final solution by Sunday, 2018-04-22 to me and the two TAs.

Exercise 7.1. Describe, in simple sentences with a minimum of mathematical formalism, (1) the score of a graph, (2) what the graph score theorem is, (3) the idea of the graph score algorithm, (4) where the difficult part of its proof is. Imagine you have a friend who does not take this class, and think about how to answer the above questions to them.

7.1 Alternative Graphs

Now we will look at different notions of graphs. As defined in class and in the video lectures, a graph is a pair $G = (V, E)$ where V is a (usually finite) set, called the *vertices*, and $E \subseteq \binom{V}{2}$, called the set of *edges*.

Multigraphs. A *multigraph* is like a graph, but you can have several parallel edges between two vertices. You cannot, however, have self-loops. That is, there cannot be an edge from u to u itself. This is an example of a multigraph:



We can define degree and score for multigraphs, too. For example, this multigraph has score $(4, 4, 2)$. Obviously no graph can have this score.

Exercise 7.2. State a score theorem for multigraphs.

Theorem 7.3 (Multigraph Score Theorem)**.** *Let $A = (a_1, \ldots, a_n) \in \mathbb{N}_0^n$, $a_1 \geq a_2 \geq \cdots \geq a_n$. There is a multigraph with this score if and only if for $A' = (a_1 - 1, a_2 - 1, a_3, \ldots, a_n)$has nonegative integers and is multigraphic.*

Furthermore, if $n = 1$, then exists a graph with score(a_1) if and only if $a_1 = 0$. If $n = 2$, then exists a graph with score(a_1, a_2) if and only if $a_1 = a_2$ and a_1, a_2 are even.

Remark. This is actually simpler than for graphs.

Exercise 7.4. Prove your theorem.

Proof. $A' \Rightarrow A$

Obviously, it is true since if A' is a multigraph, a edge is added between the largest two vertices, so A is also a multigraph.

$A \Rightarrow A'$

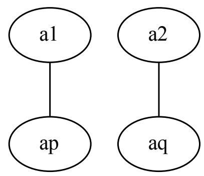
A has corresponding multigraph and we want to prove that A' also has one.

Let's prove that in A, there always exists a multigraph corresponding to A that it has an edge between the vertices of a_1, a_2.

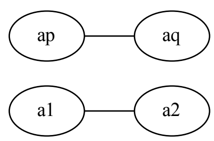
If the multigraph corresponding to A has an edge between the vertices of a_1, a_2, it's okay.

Else if there is no edge between the vertices a_1, a_2. Then there must exist another two vertices connect to them. Suppose the vertice of a_p is connected to a_1 and the vertice of a_q is connected to a_2.

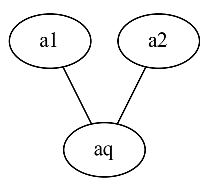
Case 1: a_p and a_q are not the same corresponding vertice.



We can transform edges (a_1, a_p) and (a_2, a_q) to (a_1, a_2) and (a_p, a_q) ((a_i, a_j) means the edge between the vertices of a_i and a_j). It is a new multigraph and it doesn't change the degree of every vertices. So this multigraph is the one we want to find.



Case 2: a_p and a_q are the same corresponding vertice.



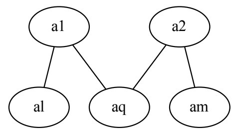
Suppose there are total u edges between the vertices of a_1, a_p and v edges between the vertices of a_2, a_p. $u, v \geq 1$

Then $a_1 \geq a_2 \geq a_p \geq u + v$ since the vertice of a_p is connected to at least two vertices.

So the vertices of a_1, a_2 have another v, u vertices to connect.

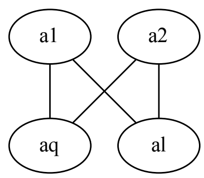
Suppose the vertice of a_1 is connected to the vertices of a_l and the vertice of a_2 is connected to the vertices of a_m.

Case 2(a): a_l and a_m are not the same corresponding vertices.



In fact, it is the same as **Case 1**. a_l, a_m are the new a_p, a_q.

Case 2(b): a_1 and a_m are the same corresponding vertice.

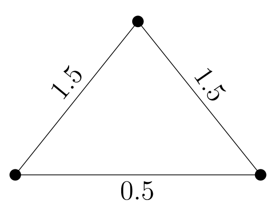


In fact, it is the same as **Case 1**. a_p, a_l are the new a_p, a_q.

So, in A, there always exists a multigraph corresponding to A that it has an edge between the vertices of a_1, a_2. And in A', we just reduce one edge between the vertices of a_1, a_2. Therefore, A' is also multigraphic.

□

Weighted graphs. A weighted graph is a graph in which every edge e has a non-negative weight w_e. In such a graph the *weighted degree* of a vertex u is $\mathrm{wdeg}(u) = \sum_{\{u,v\}\in E} w_{\{u,v\}}$.



This is an example of a weighted graph, which has score $(3, 2, 2)$. Obviously no graph and no multigraph can have this score.

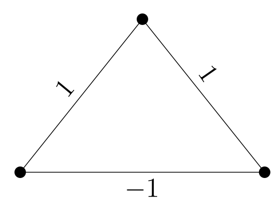
Exercise 7.5. State a score theorem for weighted graphs. That is, state something like

Theorem 7.6 (Weighted Graph Score Theorem)**.** *Let* $(a_1, \ldots, a_n) \in \mathbb{R}_0^n$. *There is a weighted graph with this score if and only if* `<fill in some simple criterion here>`.

Remark. This is actually even simpler.

Exercise 7.7. Prove your theorem.

Allowing negative edge weights. Suppose now we allow negative edge weights, like here:



This "graph with real edge weights" has score $(2, 0, 0)$. This score is impossible for graphs, multigraphs, and weighted graphs with non-negative edge weights.

Exercise 7.8. State a score theorem for weighted graphs when we allow negative edge weights. That is, state a theorem like

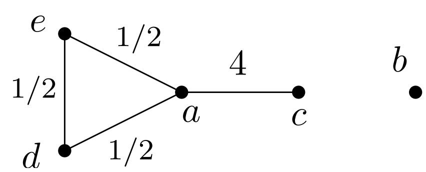
Theorem 7.9 (Score Theorem for Graphs with Real Edge Weights)**.** *Let* $(a_1, \ldots, a_n) \in \mathbb{R}^n$. *There is a graph with real edge weights with this score if and only if* `<fill in some simple criterion here>`.

Exercise 7.10. Prove your theorem.

Exercise 7.11. For each student ID $(a_1, \ldots, a_n)$ in your group, check whether this is (1) a graph score, (2) a multigraph score, (3) a weighted graph score, or (4) the score of a graph with real edge weights.

Whenever the answer is *yes*, show the graph, when it is *no*, give a short argument why.

Example Solution. My work ID is 50411. This is a weighted graph score, as shown by this picture:



This settles (3). It is not a multigraph score, because BLABLABLA. I won't give more details, as it might give too many hints about Exercise 7.2. Alright, this settles (2). Note that I do *not* need to answer (4), as this is already answered by (3). Neither do I need to answer (1), as a "no" for (2) implies a "no" for (1).