

# dOpt Compulsory Assignment 2

Christian Zhuang-Qing Nielsen\*

April 2, 2018

## Contents

<b>1 Show that the problem can be formulated as a maximum flow problem</b>	<b>1</b>
<b>2 Why wouldn't the three-point rule work?</b>	<b>2</b>
<b>3 Formulate the problem (with three-point rule) as a integer LP problem</b>	<b>2</b>
<b>1 Show that the problem can be formulated as a maximum flow problem</b>	

The question of whether or not team 1 can win the championship can be formulated as a maximum flow problem. As we can assume without loss of generality that team 1 wins the remaining of their matches, we don't need to include them in our graph. Instead, we introduce a variable  $g_i$  which is the maximum number of points team  $i$  can receive if they win the remainder of their matches.

The network consists of a source node, a sink node, a node for every match and a node for every team. The source node sends two points to every match node, which then distributes them to one or both nodes of the teams linked to that specific match. The source node also sends the current points of each team to the team nodes. The team nodes then send these as well as the points gained from the matches to the sink. If we specify the upper bound of  $< p_1 + g_1$  we ensure that no team will exceed the maximum

---

\*201504624, christian@czn.dk

theoretical point of team one, and since we assume they team one wins all their games we now know that if the network is feasible, team one will win the championship. The network is shown below:

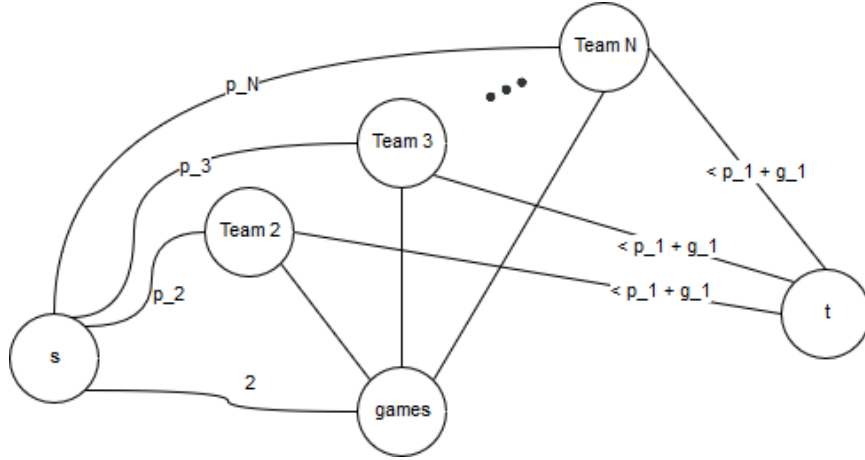


Figure 1: The illustrated network. In reality the games-node consists of a node for every match.o

## 2 Why wouldn't the three-point rule work?

When utilising the two-point rule, there are in reality two cases: In the first case, one team wins and gains two points while the other gains none, and in the other case each team gains a point each (in case of a a draw). In both cases the amount of flow-units is exactly 2. This is not the case when using the three-point rule, as there are 3 flow-units in the winning case and only 2 in the draw-case. This makes the nodes in the network unbalanced, and thus it cannot be defined as a maximum-flow problem. Sending three flow-units to every match and redirecting the additional flow to the sink wouldn't work either, as it could result in the case where a team got two points somehow, which isn't using this rule.

## 3 Formulate the problem (with three-point rule) as a integer LP problem

For every team we define three unique variables, which respectively are the amount of wins, draws and losses every team has. These variables have the

coefficients 3, 1 and 0, which are the amount of points allotted for every result. Furthermore they also have their current point-score  $p_i$  for team  $i$ . Which means every team has an equation on the form:

$$h_i = p_i + x_{1i} + x_{2i} + x_{3i},$$

where the  $i$ -index is the team number.

To constrain this linear equation we define a new constant  $t_1$ , this is the total amount of points that team 1 can end with. We also define  $r_i$  to be the number of rounds left to play for team  $i$ .  $t_1$  is then:

$$t_1 = p_1 + (3 * r_1)$$

As previously mentioned we use this to constrain the example equation earlier. We do this for every team. Using team 2 as an example:

$$h_2 = p_2 + 3x_{12} + 1x_{22} + 0x_{32} < t_1.$$

To make sure that every team does not play more matches than they are allowed, we define  $k_{hi} = |a_i| + |b_i|$ , which is the amount of matches where team  $i$  is a participant. To make sure that all teams don't end up with only losses, we also make a constraint that checks that the points are distributed correctly among the teams. We do this by checking that  $3*x_{1i} + 1*x_{2i} + 0*x_{3i}$  is between the total points to gain if every team draws and every teams wins (i.e. between the amount of remaining matches and three times the amount of remaining matches).

Finally we have the constraint that all variables are non-negative. For our objective function we minimise 0, as we only need to check if the program is feasible (and thus whether or not it is possible for team 1 to win the championship