# Advanced algorithms and data structures

## Assignment 1

Lasse Ahlbech Madsen, Kasper Passov

# Exercise 1

In figure a the requirements for a b-flow could be met, by sending 2 from $v_2$ to $v_4$, 3 from $v_5$ to $v_1$ and 4 from $v_5$ to $v_3$. This would mean that all nodes matched their demands.
In figure b there are no possible b-flows as $v_4$ has a negative demand, but no outgoing edges.

# Exercise 2

## 2.1

We define our z values as seen on tables 1 and 2

| | $a$ | $b$ | $c$ | $d$ | $e$ |
|---|---|---|---|---|---|
| $a$ | $-$ | $(v1, v6)$ | $(v1, v3)$ | $NA$ | $(v3, v5) + (v5, v7) + (v6, v7)$ |
| $b$ | $(v6, v1)$ | $-$ | $(v1, v2)$ | $(v2, v6)$ | $NA$ |
| $c$ | $(v3, v1)$ | $(v2, v1)$ | $-$ | $(v2, v3)$ | $NA$ |
| $d$ | $NA$ | $(v6, v2)$ | $(v3, v2)$ | $-$ | $(v3, v4) + (v4, v6)$ |
| $e$ | $(v7, v6) + (v7, v5) + (v5, v3)$ | $NA$ | $NA$ | $(v6, v4) + (v4, v3)$ | $-$ |

Table 1: The edges between each pair of bounded cycles

| | $a$ | $b$ | $c$ | $d$ | $e$ |
|---|---|---|---|---|---|
| $a$ | 0 | 0 | 0 | 0 | 0 |
| $b$ | 2 | 0 | 1 | 1 | 0 |
| $c$ | 1 | 1 | 0 | 0 | 0 |
| $d$ | 0 | 1 | 0 | 0 | 2 |
| $e$ | 4 | 0 | 0 | 0 | 0 |

Table 2: The actual number of breakpoints between each pair of bounded cycles

$$\sum_g \sum_f z_{gf} = \text{Total breakpoints} \tag{1}$$

We find the total breakpoints in all edges by (1) and get **13**.

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ | $v_6$ | $v_7$ |
|---|---|---|---|---|---|---|---|
| $a$ | 0 | | 1 | | 1 | 1 | 0 |
| $b$ | 1 | 0 | | | | 1 | |
| $c$ | 1 | 1 | 1 | | | | |
| $d$ | | 1 | 1 | $-1$ | | 1 | |
| $e$ | | | 1 | 1 | $-1$ | 1 | 0 |

## 2.2

As a breakpoint in an edge $z_{fg}$ must have a turn of 90 degree or less, its counterturn on $z_{gf}$ must have 180 degrees or more, and be an outer turn. So $z_{gf}$ must be the outer turns of $z_{fg}$. We now define the constraint:

$$b_f = \sum_{g \in E} (z_{fg} - z_{gf}) + \sum_{v \in G} x_{vf} \tag{2}$$

For a this would be:

$$b_a = z_{ab} - z_{ba} + z_{ac} - z_{ca} + z_{ae} - z_{ea} + x_{v_1 a} + x_{v_3 a} + x_{v_5 a} + x_{v_6 a} + x_{v_7 a}$$
$$b_a = 0 - 2 + 0 - 1 + 0 - 4 + 0 + 1 + 1 + 1 + 0 = -4$$

Which is expected of the external face.
For e this would be:

$$b_e = z_{ea} - z_{ae} + z_{ed} - z_{de} + x_{v_3 e} + x_{v_4 e} + x_{v_5 e} + x_{v_6 e} + x_{v_7 e}$$
$$b_e = 4 - 0 + 0 - 2 + 1 + 1 + -1 + 1 + 0 = 4$$

Which is expected of an internal face.

## 2.3

The assumption that a vertex in G can have a degree no higher than 4 is necessary as there are only 4 possible ”entrances” to a vertex when it is drawn on a rectilinear form.

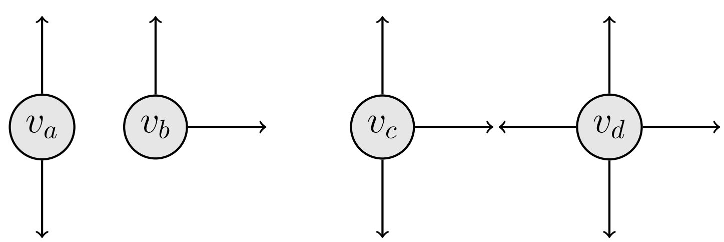


Figure 1: The 4 vertex cases depending on its degree

In figure 1 the possible cases for the vertices can be seen. If v is of degree 2 and placed between boundary cycles $f_1$ and $f_2$ there are two cases. One where there is no turn in the vertex and one where there is a single turn. In the first case both $x_{vf_1}$ and $x_{vf_2}$ will be zero. In the second case the turn will be an outer turn in regards to one of the cycles and an inner turn in regards to the other. The sum is zero in this case as well.

If v is of degree 3 and placed between boundary cycles $f_1$, $f_2$ and $f_3$ there is one case. It will be a T-cross where the vertex is flat in regards to one of the cycles and an inner turn for the two others. The sum is then 2. If v is of degree 4 and placed between boundary cycles $f_1$, $f_2$, $f_3$ and $f_4$ there is one case. In this case it will be a cross in the middle of the 4 cycles and it will form 4 inner circles. Thus the sum is 4. There are no more cases and the form held for all of them.

## 2.4

We have the set of all boundary cycles in the rectilinear layout G and V being all edges in G.
We then define the objective function to minimize as follows:

$$\sum_{f \in G} \sum_{g \in G} (z_{fg} + z_{gf}) + \sum_{f \in G} \sum_{v \in V} x_{vf} \tag{3}$$

And the linear program as follows:

$$\begin{array}{llllll}
\text{minimize} & \sum\limits_{f \in G} \sum\limits_{g \in B} (z_{fg} + z_{gf}) & + & \sum\limits_{f \in B} \sum\limits_{v \in V} x_{vf} & & \\
\text{subject to} & \sum\limits_{g \in E} (z_{fg} - z_{gf}) & + & \sum\limits_{v \in G} x_{vf} & = 4 & \text{for f being an internal face} \\
 & \sum\limits_{g \in E} (z_{fg} - z_{gf}) & + & \sum\limits_{v \in G} x_{vf} & = -4 & \text{for f being the external face} \\
 & \sum\limits_{f \in G} x_{vf} & & & = 0 & \text{if } v \text{ has degree 2} \\
 & \sum\limits_{f \in G} x_{vf} & & & = 2 & \text{if } v \text{ has degree 3} \\
 & \sum\limits_{f \in G} x_{vf} & & & = 4 & \text{if } v \text{ has degree 4} \\
 & & z_{fg}, x_{vf} & & \geq & 0
\end{array}$$

Table 3: Linear program of the breakpoint problem.

We can use equality in our constraints as the given theorem tells us the optimal solution contains integer values.

## 2.5

**Transformation** To transform a graph to the wanted form, we use all of our vertices and faces as vertices in the new graph. The vertix representing the external face gets a demand of -4 and all the internal faces get a demand of 4. The vertices have weight equal to the negation of the results from 2.3. That is: If a vertex has two edges, it has a demand of 0, if it has three -2 and finally if it has 4 edges it has a demand of -4.

We connect the face-vertices to all the vertex-vertices that appear in their boundary cycle.

The MCFP instance coresponding to Figure 2(b) can be seen on figure 2.

**MCFP reduction** We wish to minimize the sum of all incoming and outgoing edges from all vertices as this minimization must be the least amount of flow through the network. For

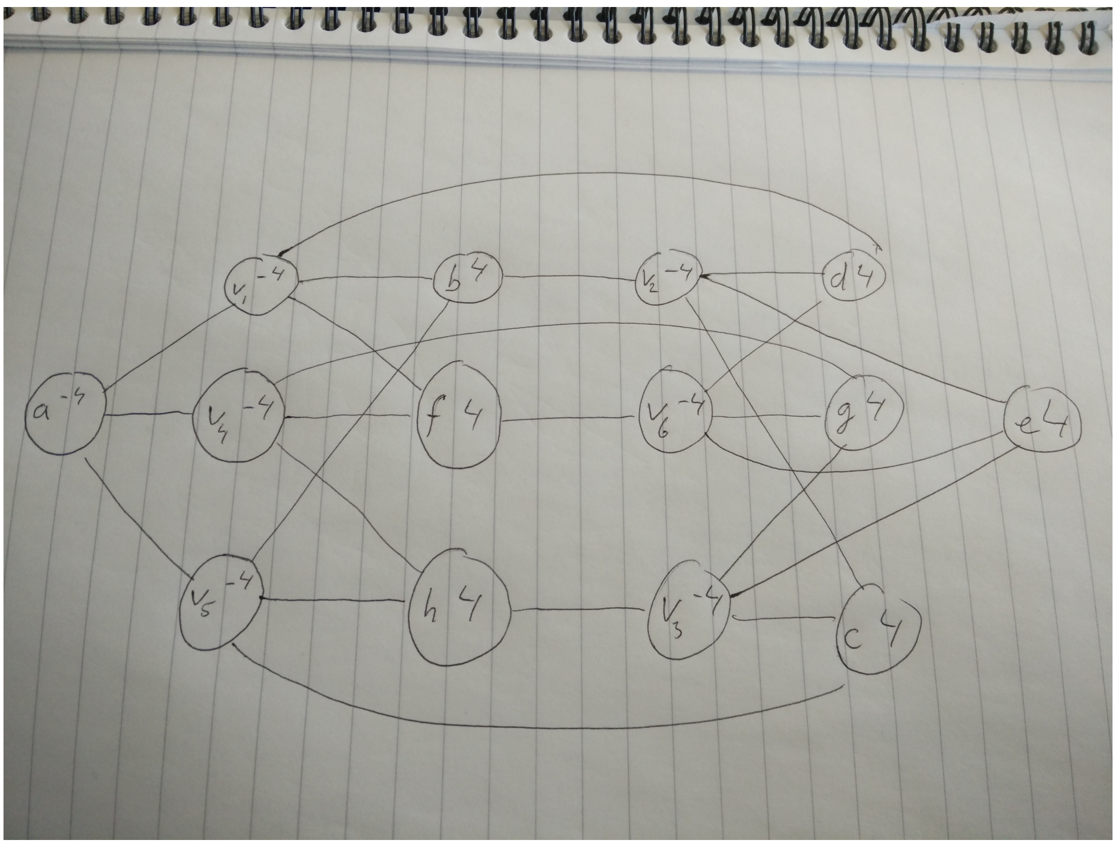


Figure 2: MCFP instance

this we define $\delta(v) = \delta^-(v) \cap \delta^+(v)$.

$$\sum_{v \in V} \sum_{e \in \delta(v)} x_e \tag{4}$$

$$
\begin{array}{llllll}
\text{minimize:} & \sum_{v \in V} \sum_{e \in \delta(v)} x_e & & & & \\
\text{subject to:} & \sum_{v \in V} \sum_{ein \in \delta^-(v)} x_{ein} \cdot c_{ein} & - & \sum_{v \in V} \sum_{eout \in \delta^+(v)} x_{eout} \cdot c_{eout} & \geq & b_v \\
 & 0 \leq \; x_v \; \leq \; u_e & & & &
\end{array}
$$

Table 4: MCFP solution reduced from the breakpoint problem.

**Linear program**

# Exercise3

## 3.1 - 3.3

Per definition, we assume that a negative capacity allows flow to be directed backwards through an edge. This means the edge (v,w) with upper capacity u and lower capacity l, can be treated as an anti-par of edges. With both edges (v,w),(w,v) having the lower capacity 0 and upper capacity u and l respectively. We will now show the transforming $I_0$

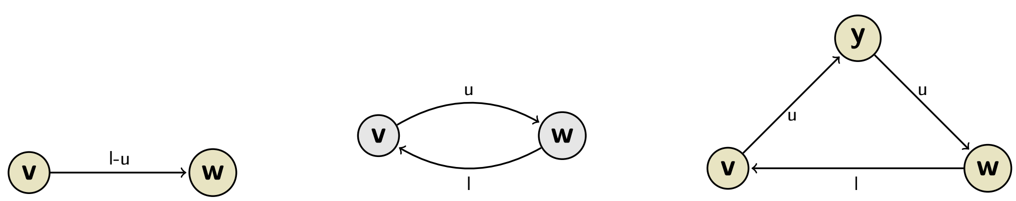


Figure 3: the transformation to be done on each edge (v,w)

to $I_1$ as shown on Figure 3 leaves a flow with $l_e = 0$ for each edge.

We assume $I_0$ is a flow satisfying flow conservation and the modified capacity constraint (5).

$$\text{For all } u, v \in V, \text{ we require } l(u,v) \leq f(u,v) \leq c(u,v) \tag{5}$$

The modified network still obeys flow conservation. As the flow from v to w is changed from negative x to 0 while the incoming flow from w through y is x. Leaving the note at negative x flow.

The change also obeys the capacity constraint. Before the modification the vertex obeys the modified capacity constraint. Removing the negative value edge from v increases the lower bound of the node capacity to 0, but since it does not modify positive edges of the upper bound, all other outgoing edges should still uphold the constraint.
This solves the first 3 subexercises of exercise 3.

### 3.4

Since we for each edge add an additional vertex and two edges, $I_3$ will be larger than the initial graph. The size of the graf is thus

$$V + E + V + 2E = 2V + 3E$$

asymptotically this is still $O(V + E)$.

### 3.5

The transformation proposed in 3.1-3.4 leaves the optimal solution $I_1$ equivalent to the optimal solution to $I_0$ as shown.