Algorithm Design - Homework 2 2015/2016

Travagliati, Lorenzo
`mat. 1495035`

Simic, Djordje
`mat. 1685456`

January 2016

1 Problem 1

1.1 Step 1

Let's call $A = (S_{i_1} \cup S_{i_2} \cup \cdots S_{i_{j-1}})$ the set of elements covered by the algorithm at a certain step. Then we define a function

$$\gamma(S_{i_j}) = \frac{c(S_{i_j})}{|S_{i_j} - A|} = price(e)$$

where $c(S_{i_j})$ is the weight of the set S_{i_j}, and e are the members of $(S_{i_j} - A)$ that will be added to A if the set S_{i_j} will be added to A. We want to show that for $1 \leq k \leq n$:

$$price(e_k) \leq \frac{OPT}{n - (k-1)}$$

Let's say that $C^* = \{T_1, \cdots, T_l\}$ is the optimal coverage for the problem. Then we define

$$J = \{1 \leq h \leq l \mid T_h \ \cap \overline{A} \neq 0\}$$

so J is the set containing all the indexes of the sets T_h that at a given time can be added to the current A.

Now we know that

$$OPT = \sum_{h=1}^{l} c(T_h)$$

Since J selects at most l sets we can write

$$\sum_{h=1}^{l} c(T_h) \geq \sum_{h \in J} c(T_h)$$

Multiplying the right part of the inequality by $\frac{|T_h - A|}{|T_h - A|}$ then we have

$$\sum_{h=1}^{l} c(T_h) \geq \sum_{h \in J} |T_h - A| \cdot \frac{c(T_h)}{|T_h - A|}$$

Since

$$\frac{c(T_h)}{|T_h - A|} = \gamma(T_h)$$

suppose that for the step k we select the minimal $\gamma(S_{i_j}) = \frac{c(S_{i_j})}{|S_{i_j} - A|}$ then we can say that

$$\gamma(T_h) \geq \gamma(S_{i_j})$$

$$\sum_{h=1}^{l} c(T_h) \geq \sum_{i \in J} |T_h - A| \cdot \frac{c(T_h)}{|T_h - A|} \geq \sum_{h \in J} |T_h - A| \cdot \frac{c(S_{i_j})}{|S_{i_j} - A|} \geq$$

$$\geq \sum_{h \in J} |T_h - A| \cdot price(e_k)$$

since $\overline{A} = \bigcup_{h \in J} T_h$ then it's trivial that

$$\sum_{h \in J} |T_h - A| \cdot price(e_k) \geq |\overline{A}| \cdot price(e_k)$$

and since we are considering the worst case possible, which is the case in which we selected just one element for each $k-1$ steps, then we have that

$$\sum_{h=1}^{l} c(T_i) \geq \sum_{i \in J} |T_i - A| \cdot \frac{c(T_h)}{|T_h - A|} \geq \sum_{h \in J} |T_h - A| \cdot \frac{c(S_{i_j})}{|S_{i_j} - A|} \geq$$

$$\geq \sum_{h \in J} |T_h - A| \cdot price(e_k) \geq |\overline{A}| \cdot price(e_k) \geq (n - (k-1)) \cdot price(e_k)$$

Then, we have that

$$OPT \geq (n - k + 1) \cdot price(e_k)$$

where $price(e_k) = \frac{c(S_{i_j})}{|S_{i_j} - A|}$ And that is equal to

$$\frac{1}{c(S_{i_j})} \cdot |S_{i_j} - A| \geq \frac{n - k + 1}{OPT}$$

while $n - k + 1 = |E_j|$, thus

$$\frac{1}{c(S_{i_j})} \cdot |S_{i_j} - (S_{i_1} \cup S_{i_2} \cup \cdots S_{i_{j-1}})| \geq \frac{|E_j|}{OPT}$$

1.2 Step 2

Since the remaining elements to be selected are

$$|E_{j+1}| = |E_j| - |S_{i_j} - (S_{i_1} \cup S_{i_2} \cup \cdots S_{i_{j-1}})|$$

$$|S_{i_j} - (S_{i_1} \cup S_{i_2} \cup \cdots S_{i_{j-1}})| = |E_j| - |E_{j+1}|$$

then

$$|E_j| - |E_{j+1}| \geq \frac{|E_j|}{OPT} \cdot w_{i_j}$$

where $w_{i_j} = c(S_{i_j})$

$$|E_{j+1}| \leq |E_j| - \frac{|E_j|}{OPT} \cdot w_{i_j}$$

$$|E_{j+1}| \leq |E_j| \left(1 - \frac{w_{i_j}}{OPT}\right)$$

1.3 Step 3

The above formula can be rewritten in the following way if we consider the whole process from step 0 to step t :

$$|E_t| \leq |E_0| \left(1 - \frac{w_{i_1}}{OPT}\right) \cdot \left(1 - \frac{w_{i_2}}{OPT}\right) \cdots \left(1 - \frac{w_{i_t}}{OPT}\right)$$

$$|E_t| \leq |E_0| \prod_{j=0}^{t} \left(1 - \frac{w_{i_j}}{OPT}\right)$$

Since $1 - x \leq e^{-x}$ then

$$|E_t| \leq |E_0| \prod_{j=0}^{t} \left(e^{-\frac{w_{i_j}}{OPT}}\right)$$

And that can be rewritten in the following form:

$$|E_t| \leq |E_0| \left(e^{-\frac{\sum_{j=0}^{t} w_{i_j}}{OPT}}\right)$$

If we name $W = \sum_{j=0}^{t} w_{i_j}$ the sum of all the set we choose until step j then

$$|E_t| \leq |E_0| e^{-\frac{W}{OPT}}$$

We finally get the full set cover when $|E_t| = 0$ or alternatively when $E_t < 1$ so:

$$|E_0| \cdot e^{-\frac{W}{OPT}} < 1$$

The remaining elements at step 0 are n, thus $E_0 = n$.

$$n \cdot e^{-\frac{W}{OPT}} < 1$$

$$e^{-\frac{W}{OPT}} < \frac{1}{n}$$

$$-\frac{W}{OPT} < \ln \frac{1}{n}$$

$$-\frac{W}{OPT} < -\ln n$$

$$W > \ln n \cdot OPT$$

2 Problem 2

2.1 Task 1

We are asked to prove that the algorithm uses at most $(\ln \frac{1}{1-p} + 1) \cdot OPT_{SC}$ sets, where OPT_{SC} denotes the optimum number of sets needed to cover the whole universe E. Since we are considering a variant of the set cover problem we can perform a similar analysis for the partial set cover problem.

We define E_t as the remaining elements not yet covered after step t and let k be the OPT_{SC}. From what we have seen in the set cover greedy algorithm analysis, for a certain t this inequality must hold:

$$|E_{t+1}| \leq |E_t| - \frac{|E_t|}{k}$$

where $k = OPT_{SC}$ and inductively:

$$|E_t| \leq n \cdot (1 - \frac{1}{k})^t$$

. Now, since we are given a partial set cover problem, we know that the algorithm will stop when $|E_t| \leq n - p \cdot n$, thus:

$$n \cdot (1 - \frac{1}{k})^t \leq n - p \cdot n$$

$$n \cdot (1 - \frac{1}{k})^t \leq n \cdot (1 - p)$$

$$(\frac{k-1}{k})^t \leq (1 - p)$$

$$(\frac{k}{k-1})^t \geq \frac{1}{(1-p)}$$

$$(1 + \frac{1}{k-1})^t \geq \frac{1}{(1-p)}$$

$$\ln (1 + \frac{1}{k-1})^t \geq \ln (\frac{1}{(1-p)})$$

$$t \cdot \ln (1 + \frac{1}{k-1}) \geq \ln (\frac{1}{(1-p)})$$

Since $\ln (1 + \frac{1}{k-1}) \approx \frac{1}{k}$ we can write

$$\frac{t}{k} \geq \ln (\frac{1}{1-p})$$

$$t \geq k \cdot \ln (\frac{1}{1-p})$$

$$t \geq OPT_{SC} \cdot \ln (\frac{1}{1-p})$$

Now we know that after $t \geq OPT_{SC} \cdot \ln(\frac{1}{1-p})$ the remaining elements are lesser than $n - p \cdot n$ so we have the partial cover in at most $t \leq OPT_{SC} \cdot \ln(\frac{1}{1-p})$, i.e. $OPT_{SC} \cdot \ln(\frac{1}{1-p})$ is an upper bound to our problem.

2.2 Task 2

We now need to prove that the greedy algorithm uses at most $(\ln(p \cdot n + 1)) \cdot OPT_{PSC}$ where OPT_{PSC} is the minimum number of sets that cover at least $p \cdot n$ elements of the universe. Consider $k = OPT_{PSC}$ and let's define E_t as the set of elements not yet covered by the partial set cover after step t and let $E_0 = p \cdot n$. We know we are done when $|E_t| < 1$ so that

$$p \cdot n \ (1 - \frac{1}{k})^t < 1$$

$$(1 - \frac{1}{k})^t < \frac{1}{p \cdot n}$$

$$(\frac{k-1}{k})^t < \frac{1}{p \cdot n}$$

$$(\frac{k}{k-1})^t > p \cdot n$$

$$\ln(p \cdot n) < \ln(1 + \frac{1}{k-1})^t$$

$$\ln(p \cdot n) < t \cdot \ln(1 + \frac{1}{k-1}) \approx \frac{t}{k}$$

$$\ln(p \cdot n) < \frac{t}{k}$$

$$t > OPT_{PSC} \cdot \ln(p \cdot n)$$

Basically, now we know that when there are no more elements picked we have $t > OPT_{PSC} \cdot \ln(p \cdot n)$, i.e. $OPT_{PSC} \cdot \ln(p \cdot n)$ is an upper bound for our algorithm, thus

$$t \leq OPT_{PSC} \cdot \ln(p \cdot n)$$

3 Problem 3

3.1 Step 1

We need to demonstrate that $\delta(X_n) = \frac{1}{2} S_n$. First we need to demonstrate that at a given step of the algorithm we have the following relation:

$$X_i = Y_i - R_i$$

where R_i is the set containing the vertices not yet processed.

Base step: $i = 0, X_0 = \{\},\ Y_0 = \{v_1, v_2, v_3, \cdots v_n\},\ R_0 = \{v_1, v_2, v_3, \cdots v_n\}$.
For $i = 0$ we have that:

$$X_0 = Y_0 - R_0$$

$$\{\} = \{v_1, v_2, v_3, \cdots v_n\} - \{v_1, v_2, v_3, \cdots v_n\}$$

$$\{\} = \{\}$$

So, for $i = 0$ the theorem holds.

Now we assume that the formula is true for i steps, so that $X_i = Y_i - R_i$.

Inductive step: Suppose we are at step $i + 1$, we need to demonstrate that:

$$X_{i+1} = Y_{i+1} - R_{i+1} \tag{1}$$

Two cases arise:

1. $X_{i+1} = X_i + \{v_{i+1}\},\ Y_{i+1} = Y_i,\ R_{i+1} = R_i - \{v_{i+1}\}$, so we can rewrite (1) as follows:

$$X_i + \{v_{i+1}\} = Y_i - (R_i - \{v_{i+1}\})$$

$$X_i + \{v_{i+1}\} = Y_i - R_i + \{v_{i+1}\}$$

$$X_i = Y_i - R_i$$

2. $X_{i+1} = X_i,\ Y_{i+1} = Y_i - \{v_{i+1}\},\ R_{i+1} = R_i - \{v_{i+1}\}$, so we can rewrite (1) as follows:

$$X_i = Y_i - \{v_{i+1}\} - (R_i - \{v_{i+1}\})$$

$$X_i = Y_i - \{v_{i+1}\} - R_i + \{v_{i+1}\}$$

$$X_i = Y_i - R_i$$

So the theorem also holds from step $i + 1$ to n. If we consider step n, then we will have that $X_n = Y_n - R_n$ and since $R_n = \{\}$ (because there are no more vertices to be processed) we have that

$$X_n = Y_n$$

3.2 Step 2

Considering sequence $O_i = \delta((OPT \cup X_i) \cap Y_i) + \delta((\overline{OPT} \cup X_i) \cap Y_i)$ we are demonstrating that:

1. $O_0 = 2 \cdot \delta(OPT)$

   Starting with:

   $$O_0 = \delta((OPT \cup X_0) \cap Y_0) + \delta((\overline{OPT} \cup X_0) \cap Y_0)$$

   Where $X_0 = \{\},\ Y_0 = \{v_1, v_2, v_3, \cdots v_n\}$, thus:

   $$O_0 = \delta(OPT \cap Y_0) + \delta(\overline{OPT} \cap Y_0)$$

   $$O_0 = \delta(OPT) + \delta(\overline{OPT})$$

   Since $\delta(OPT)$ is the number of edges that leave OPT and reach the set formed by $\overline{OPT}$, we know for sure that $\delta(\overline{OPT}) = \delta(OPT)$ so we have:

   $$O_0 = 2 \cdot \delta(OPT)$$

2. $O_n = 2 \cdot \delta(X_n)$

   $$O_n = \delta((OPT \cup X_n) \cap Y_n) + \delta((\overline{OPT} \cup X_n) \cap Y_n)$$

   Knowing that $X_n = Y_n$ (as we have demonstrated in Step 1) we can rewrite this as:

   $$O_n = \delta((OPT \cup X_n) \cap X_n) + \delta((\overline{OPT} \cup X_n) \cap X_n)$$

   Since $(OPT \cup X_n) \cap X_n = X_n$:

   $$O_n = \delta(X_n) + \delta(X_n)$$

   $$O_n = 2 \cdot \delta(X_n)$$

3.3 Step 3

To show that the function $S \to \delta(S)$ is submodular we introduce a new function $f(U_1, U_2) \to \mathbb{N}$ where U_1, U_2 are two sets of vertices and the function returns the number of edges going from U_1 to U_2, so we can rewrite:

$$\delta(T + v) - \delta(T) \leq \delta(S + v) - \delta(S)$$

$$\delta(v) - f(v, T) \leq \delta(v) - f(v, S)$$
$$-f(v, T) \leq -f(v, S)$$
$$f(v, T) \geq f(v, S)$$

Last inequality is true since $S \subseteq T$.

3.4 Step 4

In this step we are asked to prove the following inequality:

$$S_{i+1} - S_i \geq O_i - O_{i+1} \tag{2}$$

By doing that, 4 cases needs to be demonstrated:

1. *Case 1*: $v_{i+1} \in OPT,\ Y_{i+1} = Y_i - v_{i+1}, X_i = X_{i+1}$

   First, note that with this condition $S_i = \delta(Y_i) + \delta(X_i)$ and $S_{i+1} = \delta(Y_{i+1}) + \delta(X_i)$ so, the left part of the inequality can be rewritten as $S_{i+1} - S_i = \delta(X_i) + \delta(Y_{i+1}) - \delta(Y_i) - \delta(X_i) = \delta(Y_{i+1}) - \delta(Y_i)$.

   After that, we can also discuss the fact that since $v_{i+1} \notin \overline{OPT}$ then

   $$\delta((\overline{OPT} \cup X_i) \cap Y_i) = \delta((\overline{OPT} \cup X_i) \cap Y_{i+1})$$

   thus, we can say that

   $$O_i - O_{i+1} = \delta((OPT \cup X_i) \cap Y_i) - \delta((OPT \cup X_i) \cap Y_{i+1}).$$

   So we can rewrite (2) as follows:

   $$\delta(Y_{i+1}) - \delta(Y_i) \geq \delta((OPT \cup X_i) \cap Y_i) - \delta((OPT \cup X_i) \cap Y_{i+1}).$$

   From the algorithm we know that $\delta(Y_{i+1}) - \delta(Y_i) > \delta(X_{i+1}) - \delta(X_i)$ so if we can prove that $X_i \subseteq ((OPT \cup X_i) \cap Y_i)$ then, thanks to the property of submodularity of the δ-function, we can prove (2) for $v_{i+1} \in OPT,\ Y_{i+1} = Y_i - v_{i+1}$, but this is trivial since

   $$X_i \cap ((OPT \cup X_i) \cap Y_i) = X_i$$

   and thanks to the fact that $X_i \subseteq Y_i$ we can conclude that $X_i \subseteq ((OPT \cup X_i) \cap Y_i)$. It is legit then to write

   $$\delta(X_{i+1}) - \delta(X_i) \geq \delta((OPT \cup X_i) \cap Y_i) - \delta((OPT \cup X_i) \cap Y_{i+1})$$

   and this is true due to the fact that the δ-function is a *submodular* function. So we have the following:

   (a) $\delta(Y_{i+1}) - \delta(Y_i) > \delta(X_{i+1}) - \delta X_i)$

   (b) $\delta(X_{i+1}) - \delta X_i) \geq \delta((OPT \cup X_i) \cap Y_i) - \delta((OPT \cup X_i) \cap Y_{i+1})$

   Thus,

   $$\delta(Y_{i+1}) - \delta(Y_i) \geq \delta((OPT \cup X_i) \cap Y_i) - \delta((OPT \cup X_i) \cap Y_{i+1})$$

2. *Case 2*: $v_{i+1} \notin OPT,\ Y_{i+1} = Y_i - v_{i+1}, X_i = X_{i+1}$

   We proceed similarly to what we have done in the previous case, except that this time $v_{i+1} \in \overline{OPT}$ so we can write

   $$O_i - O_{i+1} = \delta((\overline{OPT} \cup X_i) \cap Y_i) - \delta((\overline{OPT} \cup X_i) \cap Y_{i+1})$$

so we can rewrite (2) as follows:

$$\delta(Y_{i+1}) - \delta(Y_i) \geq \delta((\overline{OPT} \cup X_i) \cap Y_i) - \delta((\overline{OPT} \cup X_i) \cap Y_{i+1}) \quad (3)$$

Then we can proceed as seen above, since $X_i \subseteq ((\overline{OPT} \cup X_i) \cap Y_i)$ so that

$$\delta(X_{i+1}) - \delta(X_i) \geq \delta((\overline{OPT} \cup X_i) \cap Y_i) - \delta((\overline{OPT} \cup X_i) \cap Y_{i+1})$$

holds thanks to the submodularity of the δ-function, and since also the inequality

$$\delta(Y_{i+1}) - \delta(Y_i) > \delta(X_{i+1}) - \delta(X_i)$$

holds because of the construction of the problem, the inequality (3) holds.

3. *Case 3*: $v_{i+1} \in OPT,\ X_{i+1} = X_i + \{v_{i+1}\},\ Y_{i+1} = Y_i$
Considering this case we have that $S_i = \delta(X_i) + \delta(Y_i)$ and $S_i = \delta(X_{i+1}) + \delta(Y_i)$, from this we can write that $S_{i+1} - S_i = \delta(X_i + 1) - \delta(X_i)$. Similarly to what we've done before, we can rewrite $O_i - O_{i+1} = \delta((\overline{OPT} \cup X_i) \cap Y_i) - \delta((\overline{OPT} \cup X_{i+1}) \cap Y_i)$ since $\delta((OPT \cup X_i) \cap Y_i) = \delta((OPT \cup X_{i+1}) \cap Y_i)$ because $OPT \cup X_i = OPT \cup X_{i+1}$ for the reason that v_{i+1} either belongs to OPT and X_{i+1}.
From the algorithm we also know that

$$\delta(X_{i+1}) - \delta(X_i) \geq \delta(Y_{i+1}) - \delta(Y_i)$$

so we want to prove that

$$\delta(Y_{i+1}) - \delta(Y_i) \geq \delta((\overline{OPT} \cup X_i) \cap Y_i) - \delta((\overline{OPT} \cup X_{i+1}) \cap Y_i)$$

We can rewrite $\delta(Y_{i+1}) - \delta(Y_i) = \delta(Y_{i+1}) - \delta(Y_{i+1} + v_{i+1})$ and if we consider the set

$$G = (\overline{OPT} \cup X_i) \cap Y_i$$

then

$$G + v_{i+1} = (\overline{OPT} \cup X_{i+1}) \cap Y_i$$

since the two sets differ just for the element v_{i+1}. So, basically we have

$$\delta(Y_{i+1}) - \delta(Y_{i+1} + \{v_{i+1}\}) \geq \delta(G) - \delta(G + \{v_{i+1}\})$$

. By multiplying with -1 to both sides of the inequalities we obtain

$$\delta(Y_{i+1} + \{v_{i+1}\}) - \delta(Y_{i+1}) \leq \delta(G + \{v_{i+1}\}) - \delta(G)$$

and this is true due to the sub-modularity of the δ-function if and only if $G \subseteq Y_{i+1}$ so we need to prove that, but that's trivial since $((\overline{OPT} \cup X_i) \cap Y_i)$ returns a subset of Y_i, we also know that Y_{i+1} and Y_i differs for the element v_{i+1} and we know for sure that v_{i+1} is not included in $\overline{OPT} \cup X_i$ because for hypothesis it doesn't belong either to $\overline{OPT}$ and X_i . So the inequality

$$\delta(Y_{i+1}) - \delta(Y_i) \geq \delta((\overline{OPT} \cup X_i) \cap Y_i) - \delta((\overline{OPT} \cup X_{i+1}) \cap Y_i)$$

is true, and since

$$\delta(X_{i+1}) - \delta(X_i) \geq \delta(Y_{i+1}) - \delta(Y_i)$$

it is also true that

$$\delta(X_{i+1}) - \delta(X_i) \geq \delta((\overline{OPT} \cup X_i) \cap Y_i) - \delta((\overline{OPT} \cup X_{i+1}) \cap Y_i)$$

4. *Case 4*: $v_{i+1} \in \overline{OPT},\ X_{i+1} = X_i + v_{i+1},\ Y_i = Y_{i+1}$
   The demonstration is identical to the third case we analyzed before, except that this time:

$$O_i - O_{i+1} = \delta((OPT \cup X_i) \cap Y_i) + \delta((\overline{OPT} \cup X_i) \cap Y_i) -$$
$$-(\delta((OPT \cup X_{i+1}) \cap Y_i) + \delta((\overline{OPT} \cup X_{i+1}) \cap Y_i))$$
$$= \delta((OPT \cup X_i) \cap Y_i) + \delta((OPT \cup X_{i+1}) \cap Y_i)$$

   If we name $G = (OPT \cup X_{i+1}) \cap Y_i)$, for the same reasoning we can conclude that $G \subseteq Y_{i+1}$ so, as before, the inequality

$$\delta(Y_{i+1}) - \delta(Y_i) \geq \delta((OPT \cup X_i) \cap Y_i) - \delta(OPT \cup X_{i+1}) \cap Y_i)$$

   holds, and by knowing that, we can conclude that

$$\delta(X_{i+1}) - \delta(X_i) \geq \delta((\overline{OPT} \cup X_i) \cap Y_i) - \delta((\overline{OPT} \cup X_{i+1}) \cap Y_i)$$

   holds as well.

3.5 Step 5

From Step 4 we have that:

$$\sum_{i=0}^{n-1}(S_{i+1} - S_i) \geq \sum_{i=0}^{n-1}(O_i - O_{i+1})$$

Since this is a telescopic series, after $n-1$ steps we will have

$$S_n - S_0 \geq -O_n + O_0$$

From Step 1 we know that $S_n = 2 \cdot \delta(X_n)$ and $S_0 = \delta(X_0) + \delta(Y_0) = 0$ since $X_0 = \{\}$ and $Y_0 = \{v_0, v_1 \cdots v_n\}$. Combining this with Step 2 for right side of inequality we get

$$2 \cdot \delta(X_n) \geq -2 \cdot \delta(X_n) + 2 \cdot \delta(OPT)$$

$$4 \cdot \delta(X_n) \geq 2 \cdot \delta(OPT)$$
$$2 \cdot \delta(X_n) \geq \delta(OPT)$$

4 Problem 4

Having the primal formulation of the problem, we can construct the dual problem by introducing a new variable y'_i for each node in the given graph. With the methods provided by *Operational Research* we get to the dual formulation of our original problem:

$$\begin{array}{ll} \textbf{max} & y'_s - y'_t \\ \textbf{subject to} & y'_i - y'_j \le c_{ij}, \quad \forall ij \in E \end{array}$$

Now, the constraints on the primal formulation of the problem come all with the equality symbol, so the variables for the dual formulation are free of sign, hence we can consider the following assignation to be true:

$$y_i = -y'_i$$

Then the problem will looks like this:

$$\begin{array}{ll} \textbf{max} & y_t - y_s \\ \textbf{subject to} & y_j - y_i \le c_{ij}, \quad \forall ij \in E \end{array}$$

where each y_i variables in the dual problem can be seen as the cost to reach the node i starting from the source s.

The Dijkstra's algorithm for Shortest Path is based on the use of *labels*: the *Distance* label, that for each node keeps the cost for reaching node i and the *Previous* label that indicates from which node we reached the node i.

4.1 Feasible solutions for Primal and Dual problem formulations

Let's now concentrate on the feasible solutions for both the dual and the primal problem formulations.

Suppose that, at the end of the Dijkstra's algorithm, P_i^* is the shortest path from node s to node i for each $i \in V$. Then we can construct a feasible dual solution by having, for each node $i \in V$, $\overline{y}_i = Distance(P_i^*)$ and by showing that $\overline{y}_i$ it's a feasible solution for the dual formulation of the problem, i.e.

$$\overline{y}_j - \overline{y}_i \le c_{ij}$$

for each $ij \in E$. Suppose then that the last inequality is false, yet exists an edge $ij \in E$ such that

$$\overline{y}_j - \overline{y}_i > c_{ij}$$

and this implies that

$$Distance(P_j^*) - Distance(P_i^*) > c_{ij}$$

$$Distance(P_j^*) > c_{ij} + Distance(P_i^*)$$

So the node j doesn't belong to P_i^*; then we can build a path from s to j by chaining the path P_i^* to the edge ij. This path has cost

$$Distance(P_i^*) + c_{ij} < Distance(P_j^*)$$

and this is a contradiction because assumed that P_j^* was the shortest path from s to j. So the vector $\overline{y} = (P_s^*, P_1^*, \cdots, P_t^*)$ is a feasible solution for the dual problem.
What we need now is a way to find a feasible solution for the primal problem. After we ran the Dijkstra's algorithm over our graph G, we can iteratively pick the selected edges starting from the sink and going back to the source thanks to the label *Previous* previously defined. In any case, the selected edges must satisfy the flow conservation conditions provided by the primal formulation of the problem, and this is trivial since for every minimum *st*-path, every node belonging to P_t^* is being crossed only once; therefore we have that the equality

$$\sum_{ij \in \delta(i)_{out}} \overline{x}_{ij} - \sum_{ji \in \delta(i)_{in}} \overline{x}_{ji} = 0$$

is always true, since either the node $i \in P_t^*$, thus it will have only one incoming edge and one outgoing edge, or the node $i \notin P_t^*$ so it won't have any outgoing nor incoming edges. For the same reason we have that the following two equalities holds

$$\sum_{tj \in \delta(i)_{out}} \overline{x}_{tj} - \sum_{jt \in \delta(i)_{in}} \overline{x}_{jt} = -1$$

$$\sum_{sj \in \delta(i)_{out}} \overline{x}_{sj} - \sum_{js \in \delta(i)_{in}} \overline{x}_{js} = 1$$

since there are no outgoing edges from t and no incoming edges towards s, and they only have one incoming edge and one outgoing edge respectively because of the nature of the minimum *st*-path.

Algorithm 1 Feasible solution for Primal Problem

procedure FEASIBLESOL
    $j \leftarrow t$
    while $j \neq s$ **do**
        $i \leftarrow Previous(j)$
        $\overline{x}_{ij} \leftarrow 1$
        $j \leftarrow i$
    end while
end procedure

The selected edges are indeed a feasible solution since they all belong to the minimum *st*-path and they satisfy the primal slackness conditions by construction, in such a way that:

$$\forall\ x_{ij} \neq 0 \implies \overline{y}_j - \overline{y}_i = c_{ij}$$

4.2 Optimality and Integrality

The dual slackness condition is satisfied for the same reason mentioned above regarding the satisfiabilty of the constraints of the primal, i.e. for each $\overline{y}_i \in \overline{y}$ we have that

$$\sum_{ij\in\delta(i)_{out}} \overline{x}_{ij} - \sum_{ji\in\delta(i)_{in}} \overline{x}_{ji} = 0$$

$$\sum_{tj\in\delta(i)_{out}} \overline{x}_{tj} - \sum_{jt\in\delta(i)_{in}} \overline{x}_{jt} = -1$$

$$\sum_{sj\in\delta(i)_{out}} \overline{x}_{sj} - \sum_{js\in\delta(i)_{in}} \overline{x}_{js} = 1$$

either because the $\overline{x}_{ij}$ edge belongs to the P_t^* (and therefore its value is $\overline{x}_{ij} = 1$, so the reasons mentioned above are valid) or because $\overline{x}_{ij} = 0$ (it doesn't belong to the minimum st-path). Since the $\overline{x}_{ij}$ has being constructed in a way so that the primal slackness condition are satisfied, we have, from the *Complementary Slackness Condition Theorem* that since $\overline{x}$ and $\overline{y}$ are both feasible respectively for the primal and the dual problem and since they all satisfy the following equality:

$$\overline{x}_{ij}(c_{ij} - \overline{y}_j + \overline{y}_i) = 0$$

then we can conclude that both $\overline{x}$ and $\overline{y}$ are optimal and therefore we have the following equality (from *Strong Duality Theorem*):

$$\sum_{ij\in E} c_{ij}\overline{x}_{ij} = \overline{y}_t - \overline{y}_s$$

Since every variable we select in the primal problem is either 1 or 0 and by assuming that every costs presented in the problem are in $\mathbb{N}$ we can state that the problem is indeed integral.

5 Problem 5

Let's have $0 \leq x \leq 1$ and let's assume that x is the amount of traffic routed on the upper edge of the network, hence $1 - x$ amount of traffic is being used for the lower edge. Thus, total cost function will be:

$$C(x) = 1 \cdot x + (1 - x)(1 - x)^d = x + (1 - x)^{d+1}$$

Since we want to find the minimum cost, first we derive the function $C(x)$.

$$C'(x) = 1 - (d + 1)(1 - x)^d$$

$$C'(x) = 0 \implies 1 - (d + 1)(1 - x)^d = 0$$

$$(d + 1)(1 - x)^d = 1$$

$$
\begin{aligned}
(1-x)^d &= \frac{1}{d+1} \\
1-x &= \sqrt[d]{\frac{1}{d+1}} \\
x &= 1-\sqrt[d]{\frac{1}{d+1}} \\
x &= 1-\frac{1}{\sqrt[d]{d+1}}
\end{aligned}
$$

In case d is an even number we have that

$$x = 1-\left(\pm\frac{1}{\sqrt[d]{d+1}}\right)$$

but since $0 \leq x \leq 1$ and $1+\frac{1}{\sqrt[d]{d+1}} > 1$ then the only admissible solution for our problem is

$$x = 1-\frac{1}{\sqrt[d]{d+1}}$$

Now we must be sure that the x that minimizes our function is actually a minimum point of the total cost function so we proceed calculating the second derivative:

$$C''(x) = d(d+1)(1-x)^{d-1}$$

And since $x \in [0,1]$ we can conclude that

$$C''(x) = d(d+1)(1-x)^{d-1} \geq 0$$

for every $x \in [0,1]$.

Now

$$
\begin{gathered}
C\left(1-\frac{1}{\sqrt[d]{d+1}}\right) = 1-\frac{1}{\sqrt[d]{d+1}} + \left(1-1+\frac{1}{\sqrt[d]{d+1}}\right)^{d+1} = \\
1-\frac{1}{\sqrt[d]{d+1}} + \left(\frac{1}{\sqrt[d]{d+1}}\right)^{d}\left(\frac{1}{\sqrt[d]{d+1}}\right) = \\
1-\frac{1}{\sqrt[d]{d+1}} + \frac{1}{(d+1)\cdot\sqrt[d]{d+1}} = \\
\frac{(d+1)\cdot\sqrt[d]{d+1}-(d+1)+1}{(d+1)\cdot\sqrt[d]{d+1}} = \\
\frac{(d+1)\cdot\sqrt[d]{d+1}-d}{(d+1)\cdot\sqrt[d]{d+1}} = \\
1-\frac{d}{(d+1)\cdot\sqrt[d]{d+1}}
\end{gathered}
$$

Since the *price of anarchy* is a ratio between the worst case and the optimal case, and since the worst case is either when everyone uses the upper edge of the network or the one in which, in a non-centralized scenario, every player is going to perform its dominant strategy that in this particular case is to route the traffic on the lower edge, we finally get

$$PoA = \frac{WorstCase}{OPT} = \frac{1}{1 - \frac{d}{(d+1)\cdot \sqrt[d]{d+1}}} =$$

$$\frac{1}{\frac{(d+1)\sqrt[d]{d+1}-1}{(d+1)\sqrt[d]{d+1}}} =$$

$$\frac{(d+1)\sqrt[d]{d+1}}{(d+1)\sqrt[d]{d+1}-1}$$

6 Problem 6

Let $G < V, E >$ be a graph. A 2-edge colorable graph is a graph in which we can color the edges with 2 colors, in a way no edges of the same color share a vertex. Due to the nature of the problem, every 2-edge colorable subgraph S of G is a vertex disjoint paths or cycles of even length.

Let S be a maximum size 2-edge colorable subgraph $S \subseteq G$ and let m be the number of the edges of S, s.t. $OPT = |E(S)|$. After we run the maximum size matching algorithm we obtain the set of edges M_1, that consists in one of the set of edges we are going to consider in our solution, i.e. the set of edges we are going to color. It is legit then to write that

$$m_1 = |M_1| \geq \frac{|E(S)|}{2} = \frac{OPT}{2}$$

since we performed a maximum size matching out of the graph G.

Then we proceed by running another maximum size matching over the subgraph $G - M_1$. Since a subgraph of a 2-edge colorable graph is 2-edge colorable, in the set $G - M_1$ we can find a matching of size

$$m_2 = |M_2| \geq \frac{|E(S) - M_1|}{2}$$

and the inequality holds for the same reason above, since M_2 is going to be a color in the final 2-edge colorable graph. Moreover

$$|M_1 \cup M_2| \geq M_1 + \frac{|E(S) - M_1|}{2} \geq m_1 + \frac{OPT - m_1}{2} =$$

$$= \frac{(m_1 + OPT)}{2} \geq \frac{\frac{OPT}{2} + OPT}{2} = \frac{3 \cdot OPT}{4}$$

and then

$$|M_1 \cup M_2| \geq \frac{3}{4} \cdot OPT$$