Algorithmics WS2013 Programming exercise

Lukas Petermann (0725146) Nick Mayerhofer (0726179)

January 21, 2014

Contents

1 Task Description

Given an

- undirected graph: $G = (V, E, w)$ with
- nonnegative weighting function $w(e) \in \mathbb{R}_0^+, \forall e \in E$
- and an integer $k \leq |V|$

Goal: Find a minimum weight tree, spanning exactly k nodes using the cplex framework.

For this we use three different approaches:

- Singe Commodity Flow
- Mutli Commodity Flow
- Miller Tucker Zemlin

2 Single commodity flows (SCF)

2.1 Our Approach

The Single commodity flow uses an outgoing flow from an artificial node. This flow is set to the amount of nodes we want our spanning tree to have. Each time the flow passes a node the sum of the outgoing flows will be reduced by one. Any node receiving any flow larger than zero and every edge having a flow larger then zero are part of the spanning tree.

For this we are introducing a decision variable $x_{ij} \forall x_{ij} \in \{0,1\}$ which determines if the edge (i,j) is part of the k-MST solution.

Our objective function is the spanning tree with the lowest costs.

$$min \sum_{(i,j)\in E,\ i,j\neq 0} w_{ij} * x_{ij} \tag{1}$$

2.2 variables and constraints

The variable x can have the value 0 if the edge is not and 1 if the edge is in the solution.

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in E \tag{2}$$

For more strict constraints we need another variable y which has the value 0 if a node i is not and 1 if it is in the solution.

$$y_i \in \{0,1\} \qquad \forall i \in V \tag{3}$$

For the SCF we need a flow variable f_{ij} which can have a value between 0 and k where f_{ij} holds the flow on the edge (i,j).

$$0 \leq f_{ij} \leq k \qquad \forall (i,j) \in E \tag{4}$$

The artificial node 0 sends out a flow of value k.

$$\sum_{(0,j)\in E} f_{0j} = k \tag{5}$$

The artificial node 0 is not allowed to receive any flow.

$$\sum_{(i,0)\in E} f_{i0} = 0 \tag{6}$$

We ensure that only one edge from the artificial node to any. other node is selected.

$$\sum_{(0,j)\in E} x_{0j} = 1 \tag{7}$$

A minimum spanning tree with k nodes must have $k-1$ edges.

$$\sum_{(i,j)\in E,\ i,j\neq 0} x_{ij} = k-1 \tag{8}$$

better solution
because min is NOT a LP

The following equation states that the sum of all incoming flows minus the sum of all outgoing flows must be 0 or 1. This ensures that ~~each time a node is~~ selected, a commodity is consumed or no flow is incoming or outgoing.

$= y_v$

$$\sum_{u,(v,u)\in E} f_{vu} - \sum_{u,(u,v)\in E} f_{uv} = \min(1, \sum_{u,(v,u)\in E} f_{vu}) \qquad \forall v \neq 0, v \in V \tag{9}$$

If we have a flow on the edge (i, j), then this edge has to be selected and therefore must be in the solution

$$f_{ij} \leq k * x_{ij} \qquad \forall (i,j) \in E \tag{10}$$

If the edge (i, j) is in the solution, then also both end nodes i and j have to be in the solution.

$$x_{ij} \leq y_i \qquad \forall (i,j) \in E \tag{11}$$

$$x_{ij} \leq y_j \qquad \forall (i,j) \in E \tag{12}$$

The following equation ensures that only one direction of an edge is in the solution.

$$\cancel{y_i} + x_{ij} + x_{ji} \leq y_j + \cancel{1} \qquad \forall (i,j) \in E \tag{13}$$

It's the same without it

Exactly k+1 nodes, including the artificial node, must be selected.

$$\sum_{i \in V} y_i = k + 1 \tag{14}$$

To strenghten the constraints and to further ensure that no clycles will be in the solution we need to go sure that each node has exactly 1 or none incoming edges. This equation is not necessary but will increase the performance dramatically.

$$\sum_{i \in V} x_{in} \leq 1 \qquad \forall n \in V \backslash (0) \tag{15}$$

test instance	k	objective function value	running time	branch-and-bound nodes
data/g01.dat	2	46	0	0
data/g01.dat	5	477	0.01	0
data/g02.dat	4	373	0.02	0
data/g02.dat	10	1390	0.03	0
data/g03.dat	10	725	0.06	0
data/g03.dat	25	3074	0.07	0
data/g04.dat	14	909	0.15	17
data/g04.dat	35	3292	2.18	1476
data/g05.dat	20	1235	0.36	20
data/g05.dat	50	4898	0.4	7
data/g06.dat	40	2068	47.71	2225
data/g06.dat	100	6705	30.25	2964

Table 1: SCF results

g07/08 ?

where are data/g07 08 ?

3 Multi commodity flow (MCF)

3.1 Our Approach

The Multi Commodity Flow sends out k different flows each directed to a specific node which will be in the k-MST solution. Each flow can pass arbitrarily many other flows on it's path.

For this problem we introduce the same deciscion variable $x_{ij} \forall x_{ij} \in \{0,1\}$ and the same objective function as we did for the SCF.

$$min \sum_{(i,j)\in E,\ i,j\neq 0} w_{ij} * x_{ij} \tag{16}$$

3.2 Variables and Constraints

The variable x can have the value 0 if the edge is not and 1 if the edge is in the solution.

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in E \tag{17}$$

We need another variable y which has the value 0 if a node i is not and 1 if it is in the solution.

$$y_i \in \{0,1\} \qquad \forall i \in V \tag{18}$$

For the MCF we need, like in the SCF, a flow variable, but we have n different flows where each flow f_{ij}^n has a value beween 0 and 1 and is directed to a specific node n.

$$0 \leq f_{ij}^n \leq 1 \qquad \forall (i,j) \in E, \forall n \in V \tag{19}$$

The following equation ensures that the artificial node 0 sends out a flow to each different node n with a value between 0 and 1.

$$\sum_{j,j\neq 0} f_{0j}^n \leq 1 \qquad \forall n \in V \backslash \{0\} \tag{20}$$

This equation guarantees that exactly k flows are sent to graph.

$$\sum_{(0,j)\in E} f_{0j}^n = k \qquad \forall n \in V \tag{21}$$

Here we assure that the sum of all flows which are incoming in the destiny node must have a value between 0 and 1.

$$\sum_{i,i\neq n} f_{in}^n \leq 1 \qquad \forall n \in V \backslash \{0\} \tag{22}$$

We need go sure, that the sum of all incoming flows on the destination nodes must be equal to the sum of the artificial node which has sent a value of k.

$$\sum_{n=1}^{V} \sum_{i,(i,n)\in E} f_{in}^{n} = k \tag{23}$$

Each flow which is incoming in an node other then the destination node must be sent further.

$$\sum_{i,(i,j)\in E} f_{ij}^{n} - \sum_{i,(j,i)\in E} f_{ji}^{n} = 0 \qquad \forall n \in V\backslash\{0\}, \forall j, j \neq n \tag{24}$$

We need to go sure, that if a flow greater then 0 is on an edge than the edge must be selected.

$$f_{ij}^{n} \leq x_{ij} \qquad \forall (i,j) \in E, \forall n \in V \tag{25}$$

Further we need to ensure that the artificial node 0 sends all k different flows over one specific node. Therefore only one edge can be selected.

$$\sum_{(0,j)\in E} x_{0j} = 1 \tag{26}$$

To be sure that each solution is truly a minimum spanning tree we need to be sure that there are no cylces. We accomplish that by the following equation that guarantees us that there can only be $k-1$ edges selected.

$$\sum_{(i,j)\in E, i,j\neq 0} x_{ij} = k-1 \tag{27}$$

If the edge (i,j) is in the solution, then also both end nodes i and j have to be in the solution.

$$x_{ij} \leq y_i \qquad \forall (i,j) \in E \tag{28}$$

$$x_{ij} \leq y_j \qquad \forall (i,j) \in E \tag{29}$$

The following equation ensures that only one direction of an edge is in the solution.

$$y_i + x_{ij} + x_{ji} \leq y_j + 1 \qquad \forall (i,j) \in E \tag{30}$$

Exactly k+1 nodes, including the artificial node, must be selected.

$$\sum_{i\in V} y_i = k+1 \tag{31}$$

To strenghten the constraints and to further ensure that no clycles will be in the solution we need to go sure that each node has exactly 1 or none incoming edges. This equation is not necessary but will increase the performance.

$$\sum_{i\in V} x_{in} \leq 1 \qquad \forall n \in V\backslash(0) \tag{32}$$

test instance	k	objective function value	running time	branch-and-bound nodes
data/g01.dat	2	46	0.03	0
data/g01.dat	5	477	0.02	0
data/g02.dat	4	373	0.33	11
data/g02.dat	10	1390	0.13	0
data/g03.dat	10	725	3.31	0
data/g03.dat	25	3074	8.18	0
data/g04.dat	14	909	91.96	39
data/g04.dat	35	3292	76.38	0
data/g05.dat	20	1235	368.58	0
data/g05.dat	50	4898	362.9	0

Table 2: MCF results

4 Miller Tucker Zemlin (MTZ)

4.1 Our Approach

Miller Tucker Zemlin uses sequential node ordering. We use again an artificial node 0 starting with the order 0 and increment the order every time we reach a new node by one.

Again we introduce the same deciscion variable $x_{ij} \forall x_{ij} \in \{0,1\}$ and the same objective function as we did for the SCF.

$$min \sum_{(i,j)\in E,\ i,j\neq 0} w_{ij} * x_{ij} \tag{33}$$

4.2 Variables and Constraints

The variable x can have the value 0 if the edge is not and 1 if the edge is in the solution.

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in E \tag{34}$$

We need another variable y which has the value 0 if a node i is not and 1 if it is in the solution.

$$y_i \in \{0,1\} \qquad \forall i \in V \tag{35}$$

For MTZ we introduce a new variable u_i which is the order of the every node i. It can have a value between 0 and k, because the highest order there can be is exactly k.

$$0 \leq u_i \leq k \qquad \forall i \in V \backslash (0) \tag{36}$$

The artifical node 0 has the order 0.

$$u_0 = 0 \tag{37}$$

The artificial node can have only one outgoing edge selected.

$$\sum_{(0,j)\in E} x_{0j} = 1 \tag{38}$$

This equation ensures that an edge can only exist between a node with a lower order than the targeting node. This guarantees us that there will be no cycles.

$$u_i + x_{ij} \leq u_j + k * (1 - x_{ij}) \qquad \forall (i,j) \in E \tag{39}$$

To ensure that no order will be skipped we restict the sum of orders.

$$\sum_{i\in V} u_i = \frac{k * (k+1)}{2} \tag{40}$$

slows down

macht langsam

We also need to ensure that if a node has a order greater than 0 then the node has to be selected.

test instance	k	objective function value	running time	branch-and-bound nodes
data/g01.dat	2	46	0	0
data/g01.dat	5	477	0	0
data/g02.dat	4	373	0.03	15
data/g02.dat	10	1390	0.05	23
data/g03.dat	10	725	0.25	137
data/g03.dat	25	3074	1.81	2559
data/g04.dat	14	909	0.85	94
data/g04.dat	35	3292	2.62	1171
data/g05.dat	20	1235	0.98	339
data/g05.dat	50	4898	5.26	3811
data/g06.dat	40	2068	119.58	20930
data/g06.dat	100	6705	110.34	22861

Table 3: MTZ results

$$u_i \leq k * y_i \qquad \forall i \in V \tag{41}$$

The k-MST solution must have exactly $k-1$ edges selected.

$$\sum_{(i,j)\in E,\ i,j\neq 0} x_{ij} = k-1 \tag{42}$$

Further the solution must have exactly k nodes selected.

$$\sum_{i\in V, i\neq 0} y_i = k \tag{43}$$

To strenghten the constraints and to further ensure that no clycles will be in the solution we need to go sure that each node has exactly 1 or none incoming edges.

$$\sum_{i\in V} x_{in} \leq 1 \qquad \forall n \in V\backslash(0) \tag{44}$$

A node can only have an outgoing edge selected, if the node has a order greater than 0.

$$x_{ij} \leq u_i \qquad \forall (i,j) \in E, i \neq 0 \tag{45}$$

A node can only have an incoming edge selected, if the node has a order greater than 0.

$$x_{ij} \leq u_j \qquad \forall (i,j) \in E, j \neq 0 \tag{46}$$

The following equation ensures that only one direction of an edge is in the solution.

$$y_i + x_{ij} + x_{ji} \leq y_j + 1 \qquad \forall (i,j) \in E, i \neq 0, j \neq 0 \tag{47}$$

5 results

Comparing the results we can see that SCF performs better then MTZ and both are performing way better then MCF.