Master's Thesis Defense

The Prize-Collecting Steiner Tree Problem and Related Problems

William Jack Lysgaard Sprent

DIKU
University of Copenhagen

31st August 2018

The fun we are having today:

Outline

Initial Problem Statement

Apply results from research on the more covered Prize-Collecting Traveling Salesman Problem to the lesser covered Prize-Collecting Steiner Tree problem.

What I spent six months doing?

What I spent six months doing?

- I read a stack of research papers about the PCSTP.
- I read a smaller stack of research papers about problems related to the PCSTP.

What I spent six months doing?

- I read a stack of research papers about the PCSTP.
- I read a smaller stack of research papers about problems related to the PCSTP.
- *I was indecisive.*

What I spent six months doing?

- I read a stack of research papers about the PCSTP.
- I read a smaller stack of research papers about problems related to the PCSTP.
- *I was indecisive.*
- I worked on a solver for the Median Tree Problem.

Revised Problem Statement

Apply results from research on the more covered Prize-Collecting Steiner Tree Problem to the lesser covered Median Tree Problem.

Goals

1. Survey research on the PCSTP
2. Identify methods worth *porting*
3. Implement these methods in a solver for the MTP

What makes a survey necessary?

What makes a survey necessary?

- A lot is written about the PCSTP, but it is unstructured and disjoint.

What makes a survey necessary?

- A lot is written about the PCSTP, but it is unstructured and disjoint.
- Some of these papers touch on very complex subjects and are sometimes short and unintuitive.

What makes a survey necessary?

- A lot is written about the PCSTP, but it is unstructured and disjoint.
- Some of these papers touch on very complex subjects and are sometimes short and unintuitive.
- The PCSTP is a good “case study” for an Graph Optimisation and ILP problem.

What makes a survey necessary?

- A lot is written about the PCSTP, but it is unstructured and disjoint.
- Some of these papers touch on very complex subjects and are sometimes short and unintuitive.
- The PCSTP is a good “case study” for an Graph Optimisation and ILP problem.

What makes a survey necessary?

- A lot is written about the PCSTP, but it is unstructured and disjoint.
- Some of these papers touch on very complex subjects and are sometimes short and unintuitive.
- The PCSTP is a good “case study” for an Graph Optimisation and ILP problem.

– There is a lot to learn.

Why the Median Tree Problem?

Why the Median Tree Problem?

We'll get back to that later.

Outline

Short (Meta) History

- Defined by Egon Balas in 1988 as a side effect.

Short (Meta) History

- Defined by Egon Balas in 1988 as a side effect.
- Subject to steady focus in the late 90’s and early 00’s.

Short (Meta) History

- Defined by Egon Balas in 1988 as a side effect.
- Subject to steady focus in the late 90's and early 00's.
- One of the subjects of the 11th DIMACs Implementation Challenge in 2014.

The Prize-Collecting Steiner Tree Problem

Problem Definition

Given an undirected graph

$$G = (V, E, c, p)$$

where $c : E \to \mathbb{R}^+$ defines edge weights, and $p : V \to \mathbb{R}^+$ defines vertex *prizes*, then the solution to the *PCSTP* is a tree

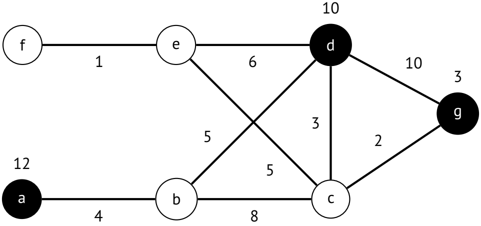
$$T = (V_T, E_T, c, p) \subseteq G$$

which minimizes

$$c(T) = \min_T \sum_{(i,j) \in E_T} c_{ij} + \sum_{v \in (V \setminus V_T)} p_v.$$

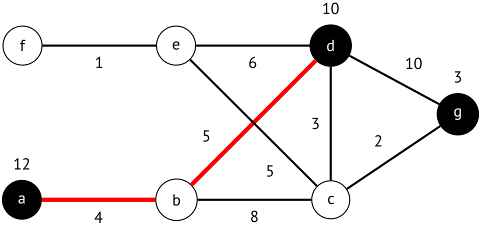
The Prize-Collecting Steiner Tree Problem

Example



The Prize-Collecting Steiner Tree Problem

Example



Outline

Contents of the PCSTP Survey:

1. The history of solving the PCSTP.
2. Preprocessing routines.
3. Two heuristic algorithms. LP-based and search based.
4. An approximation algorithm: the GW Algorithm.
5. How to separate GSECs.
6. The DHEA and SCIP-Jack solvers.

What have we learned?

The Survey

Main Points

What have we learned?

1. The PCSTP is well covered all things considered.

What have we learned?

1. The PCSTP is well covered all things considered.
2. Preprocessing is *very* good for the PCSTP.

What have we learned?

1. The PCSTP is well covered all things considered.
2. Preprocessing is *very* good for the PCSTP.
3. Directed formulations of the problem are preferable for branch and bound.

What have we learned?

1. The PCSTP is well covered all things considered.
2. Preprocessing is *very* good for the PCSTP.
3. Directed formulations of the problem are preferable for branch and bound.
4. Heuristics are aplenty.

Curiosities

Curiosities

- Not a lot of focus on applications
- Linear progress turns to lateral progress
- Somewhat general methods besides preprocessing

Curiosities

- Not a lot of focus on applications
- Linear progress turns to lateral progress
- Somewhat general methods besides preprocessing

On to other problems.

Three main variants:

1. The Prize-Collecting Travelling Salesman Problem
2. The Orienteering Problem
3. The Profitable Tour Problem

Three main variants:

1. The Prize-Collecting Travelling Salesman Problem
2. The Orienteering Problem
3. The Profitable Tour Problem

Some Notes:

- Main point of the Balas paper in 88
- Shares approximation algorithms
- Apart from the OP, not well covered

Notes

- Assignment Problem
- Different shapes of facility
- Median Trees are only research on shaped graphs

Summary

- Two axes of similarity: structure of solution and type of prize function
- PCSTP is the most well researched problem in the family

Outline

Median Tree Problem

Motivation

Question: How do I best make use of the survey?

Median Tree Problem

Motivation

Question: How do I best make use of the survey?

Options

- Stay with the PCSTP

Median Tree Problem

Motivation

Question: How do I best make use of the survey?

Options

- Stay with the PCSTP
- Look at the Profitable Tour Problem

Question: How do I best make use of the survey?

Options

- Stay with the PCSTP
- Look at the Profitable Tour Problem
- Look at the Median Tree Problem

Median Tree Problem

Motivation

Median Trees instead of Prize-Collecting Tours

- A feasible solution to the PCSTP is a feasible solution to the MTP
- Collect Prize vs. Assignment: similar – although not the exact same – trade offs

Median Trees instead of Prize-Collecting Tours

- A feasible solution to the PCSTP is a feasible solution to the MTP
- Collect Prize vs. Assignment: similar – although not the exact same – trade offs

And, of course, a splash of subjectivity.

Median Tree Problem

Problem Definition

Let $G = (V, E, c, d)$ be an undirected graph. Denote $c : E \to \mathbb{R}^+$ as an *edge cost* function and $d : V \times V \to \mathbb{R}^+$ be an *assignment cost* function where we have

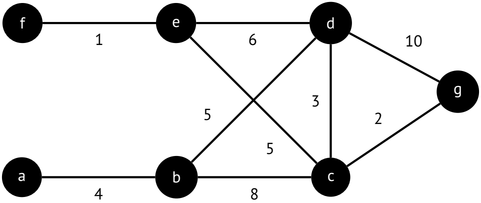
$$d_{ii} = 0.$$

Then the *Median Tree Problem* is defined as finding a *connected subgraph* $T = (V_T, E_T)$ of G where $V_T \subseteq V$ and $E_T \subseteq E$ which minimises the cost function,

$$c(T) = \sum_{ij \in E_T} c_{ij} + \sum_{i \in V} \min_{j \in V_T} d_{ij}.$$

Median Tree Problem

Example



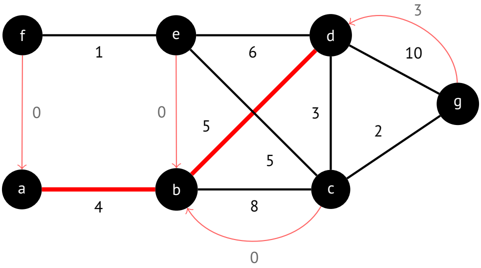
Median Tree Problem

Example

d_{ij}	a	b	c	d	e	f	g
a	0	12	12	12	12	12	12
b	0	0	0	0	0	0	0
c	0	0	0	0	0	0	0
d	10	10	10	0	10	10	10
e	0	0	0	0	0	0	0
f	0	0	0	0	0	0	0
g	3	3	3	3	3	3	0

Median Tree Problem

Example



Median Tree Problem

ILP Formulation

$$
\begin{aligned}
\underset{\mathbf{x},\mathbf{y}}{\text{minimize}} \quad & \sum_{ij \in E} c_{ij} x_{ij} + \sum_{i,j \in V} d_{ij} y_{ij} && \text{(1a)} \\
\text{subject to} \quad & \sum_{ij \in E} x_{ij} = \sum_{i \in V} y_{ii} - 1 && \text{(1b)} \\
& x(E(S)) \leq \sum_{i \in S \setminus \{s\}} y_{ii} \ \forall S \subseteq V, s \in S && \text{(1c)} \\
& \sum_{j \in V} y_{kj} = 1 \qquad \forall k \in V && \text{(1d)} \\
& y_{ik} \leq y_{kk} \qquad \forall i, k \in V && \text{(1e)} \\
& y_{kk} \leq \sum_{i \in \delta(k)} x_{ik} \qquad \forall k \in V && \text{(1f)} \\
& \mathbf{x} \in \mathbb{B}^{|E|} && \text{(1g)} \\
& \mathbf{y} \in \mathbb{B}^{|V \times V|} && \text{(1h)}
\end{aligned}
$$

Median Tree Problem

Valid Inequalities

Median Tree Problem

Valid Inequalities

Forced self-assignment:

$$y_{ii} \geq x_{ji} \qquad \forall i \in V, \forall j \in \delta(i).$$

Median Tree Problem

Valid Inequalities

Forced self-assignment:

$$y_{ii} \geq x_{ji} \qquad \forall i \in V, \forall j \in \delta(i).$$

Degree of Nonterminals:

$$\sum_{j \in \delta(i)} x_{ij} \geq 2x_{ik} \qquad \forall i \in N,\ \forall k \in \delta(i)$$

Specifications

- Based on the Gurobi MIP solver.

Median Tree Problem

The Solver

Specifications

- Based on the Gurobi MIP solver.
- Callbacks written in Python 3.

Specifications

- Based on the Gurobi MIP solver.
- Callbacks written in Python 3.
- Applies methods from the PCSTP survey:

Specifications

- Based on the Gurobi MIP solver.
- Callbacks written in Python 3.
- Applies methods from the PCSTP survey:
  - Primal heuristic from the DHEA solver.
  - User cuts based on GSEC separation from an article by Lucena and Resende for the PCSTP.

Results

Results

- Primal Heuristics: Ambivalent performance

Results

- Primal Heuristics: Ambivalent performance
- Valid Inequalities: Ambivalent performance

Results

- Primal Heuristics: Ambivalent performance
- Valid Inequalities: Ambivalent performance
- User Cuts: *Terrible* performance

Results

- Primal Heuristics: Ambivalent performance
- Valid Inequalities: Ambivalent performance
- User Cuts: *Terrible* performance

Results

- Primal Heuristics: Ambivalent performance
- Valid Inequalities: Ambivalent performance
- User Cuts: *Terrible* performance

Why?

Outline

What did I manage to do?

- Survey of the PCSTP
- ILP formulation for the MTP
- Dataset for the MTP
- Solver for the MTP

Survey criticism

- I approach approximation algorithms, exact algorithms, and heuristics the same way

Survey criticism

- I approach approximation algorithms, exact algorithms, and heuristics the same way
- I spend too much time with a “zoomed in” perspective

Survey criticism

- I approach approximation algorithms, exact algorithms, and heuristics the same way
- I spend too much time with a “zoomed in” perspective
- More figures outside the preprocessing section

Survey criticism

- I approach approximation algorithms, exact algorithms, and heuristics the same way
- I spend too much time with a “zoomed in” perspective
- More figures outside the preprocessing section
- Too little focus on application – may be a feature of the body of research

Solver section criticism

- Python is fun and fast for me, but slow for the PC

Solver section criticism

- Python is fun and fast for me, but slow for the PC
- Something is deeply wrong with the GSEC seperation routine

Solver section criticism

- Python is fun and fast for me, but slow for the PC
- Something is deeply wrong with the GSEC seperation routine
- I should have measured performance in other ways than end to end execution time

Solver section criticism

- Python is fun and fast for me, but slow for the PC
- Something is deeply wrong with the GSEC seperation routine
- I should have measured performance in other ways than end to end execution time
- The dataset is a bit uninteresting

Solver section criticism

- Python is fun and fast for me, but slow for the PC
- Something is deeply wrong with the GSEC seperation routine
- I should have measured performance in other ways than end to end execution time
- The dataset is a bit uninteresting

Reflections

Improvements

Solver section criticism

- Python is fun and fast for me, but slow for the PC
- Something is deeply wrong with the GSEC seperation routine
- I should have measured performance in other ways than end to end execution time
- The dataset is a bit uninteresting

Overall: slight lack of a red thread

Thoughts about the subject field

- There is a limit on how much research on an arbitrary problem is interesting without real world motivation

Thoughts about the subject field

- There is a limit on how much research on an arbitrary problem is interesting without real world motivation
- Same with laterally defining new problems

Thoughts about the subject field

- There is a limit on how much research on an arbitrary problem is interesting without real world motivation
- Same with laterally defining new problems
- There could be a greater focus on generalising results as widely as possible

Make the solver as good as it can be

Perform a more thorough investigation of the shortcomings in the MTP solver, and use these results to reimplement the solver.

Apply

Operationalise the research done on the PCSTP (or MTP) for a real-world problem field, and apply it to a specific scenario.

Postface

Postface

Questions?