UNIVERSITÄT DES SAARLANDES

Airports and Railways: Facility Location Meets Network Design

Tobias Mömke
Saarland University

Joint work with Anna Adamaszek and Antonios Antoniadis

NII Shonan Meeting No. 071
Shonan Village Center, 11–14 April 2016

Background

Facility Location:

Bipartition between demands and facilities

Network design:

Find network with specific properties

Relevant problems in between

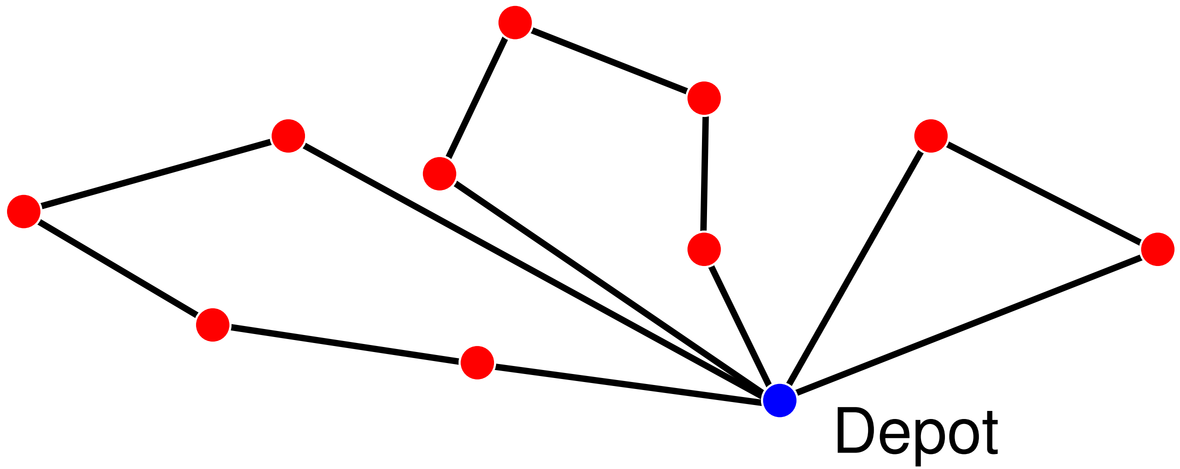
CVRP

Capacitated Vehicle Routing

Given: set of cities

Output: set of tours of size at most k each that cover all the cities

$k = 4$



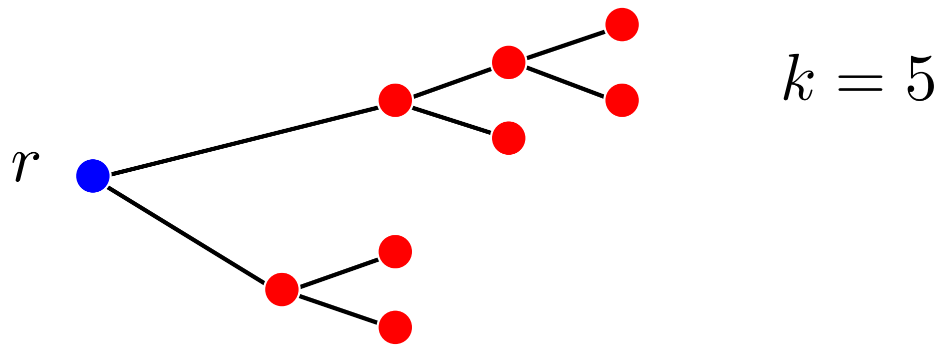
CMST

Capacitated Minimum Spanning Tree

Given: Edge-weighted graph G, root r,
vertex demands, capacity k

Feasible solutions: Forests such that for each tree,
total demand of its vertices at most k

Goal: Min. edge costs $+$ connections of trees to root r



Known Results

Capacitated Minimum Spanning Tree

Current best approx., also in Euclidean setting:

3.15-approximation [Jothi–Raghavachari TALG ’05]

Capacitated Vehicle Routing (CVRP)

(i) PTAS for very large capacity ($k = \Omega(n)$)

[Asano–Katoh–Tamaki–Tokuyama STOC ’97]

(ii) PTAS for small capacity ($k \leq 2^{\log^{o(1)} n}$)

[Adamaszek–Czumaj–Lingas ISAAC ’09]

(iii) QPTAS for all k [Das–Mathieu SODA ’10]

Framework

Aim: Formalize connection between the two problem types (i. e., facility location and network design)

- Unify solutions to problems in between
- Create tool for solving such problems

Airports and Railways

Given: set of cities
create pairwise connection with
network of airports and railways

Each city:
cost for building an airport
Each railway line:
cost proportional to distance

Each airport: serves at most k cities

Goal: minimize cost for building airports and railways

Formal Definition

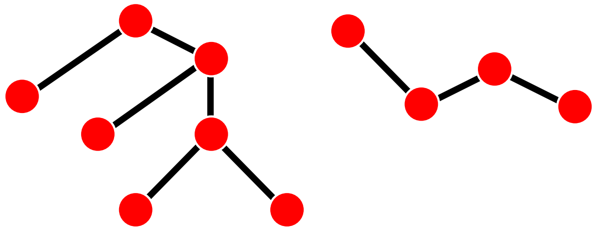
Given: Graph G with vertex costs $a(v)$, edge costs $r(e)$

Goal: Compute a min.-cost network of
$A \subseteq V(G)$ airports and $R \subseteq E(G)$ railways
connecting all the cities, where
each connected component contains
at most k vertices, and at least one airport

Subproblems of AR

Natural Problems in Framework:

Network: forest or paths
AR_F, AR_P



Capacitated or uncapacitated
AR_X, AR_X^∞

Arbitrary or uniform airport cost
AR_X, 1AR_X

$$\forall v, a(v) = 1$$

Motivation: PTAS

PTAS for Euclidean $\{$TSP, STP, etc.$\}$

Only QPTAS for CVRP!

Airports and Railways more general than CVRP

$\Rightarrow$ no PTAS known

Identify Source of Hardness

Split Properties into
(a) Respect Capacities
(b) Allow arbitrary costs

Results

1. Exact poly. time alg. for AR_F^∞
2. NP-hardness of AR_P^∞, 1AR_F, 1AR_P
3. PTAS for AR_P^∞, 1AR_F, 1AR_P (2D Euclidean)

$\Rightarrow$ We can handle arbitrary capacities and arbitrary airport costs separately

Hardness

$\text{AR}_F^\infty \in \text{P}$

1AR_P^∞ (and therefore AR_P, AR_P^∞, 1AR_P) NP-hard

Reduction from TSP-path

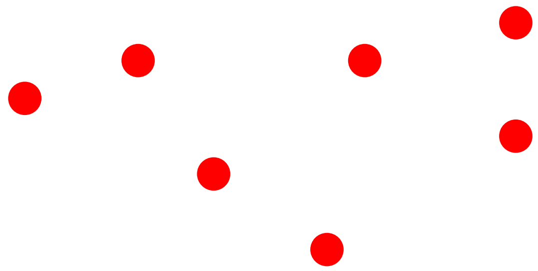
1AR_F (and therefore AR_F) NP-hard

Reduction from

Planar Monotone Cubic One-in-Three SAT

Uncapacitated Forest

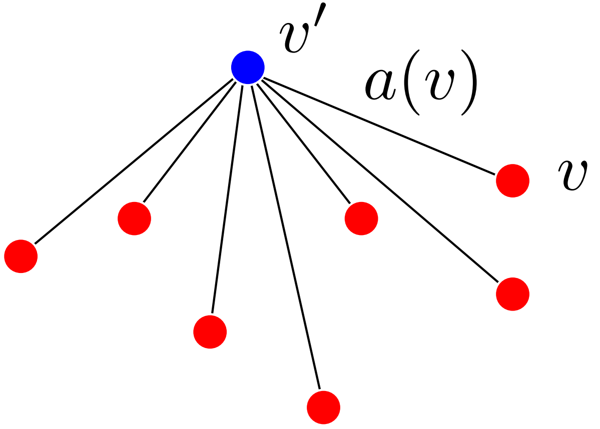
Solve MST with additional "sky vertex" v'



Extension: ensure components of *average* size k
(via matroid intersection)

Uncapacitated Forest

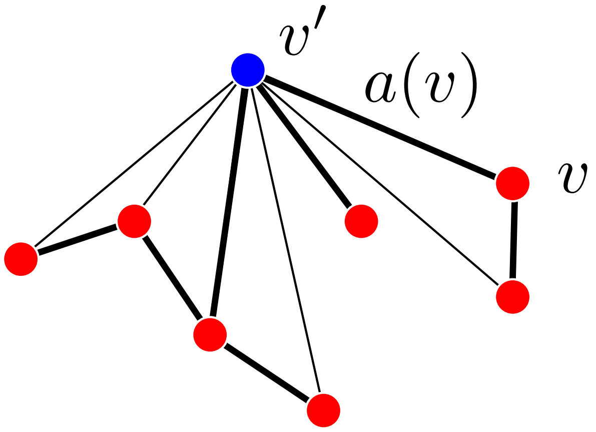
Solve MST with additional “sky vertex” v'



Extension: ensure components of *average* size k
(via matroid intersection)

Uncapacitated Forest

Solve MST with additional "sky vertex" v'



Extension: ensure components of *average* size k
(via matroid intersection)

Uncapacitated Paths

We develop a PTAS based on Arora's scheme

But almost each step needs some extra work

- Identify clusters (separate bounding boxes)
- New uncrossing
- Control number of crossings per portal
- Some extra work to combine DP cells

Uniform Airport Costs

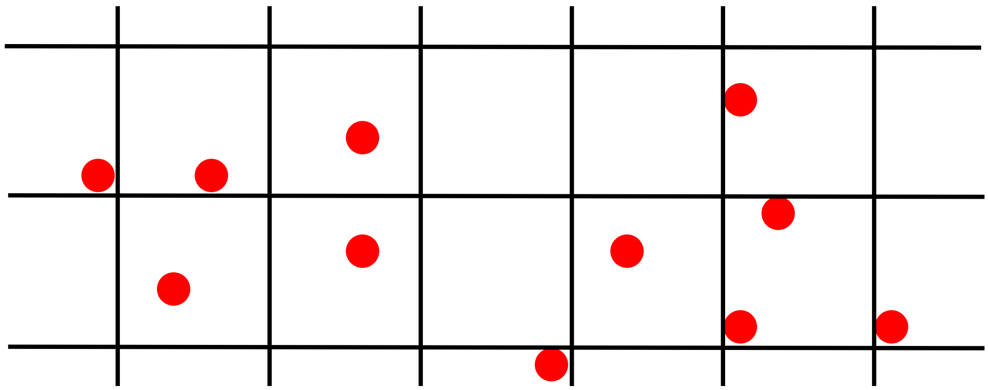
For both 1AR_F and 1AR_P:

1. Preprocess the instance
2. Subdivide into sparse and dense sub-instances
3. Solve sparse and dense substances independently
4. Recombine them to result

Preprocessing

Split instance into substances of size $\ell_i \times \ell_i$ with $\ell_i \leq 1/\varepsilon$

Randomly shift in both dimensions



Note 1:

substances of size $O_\varepsilon(1)$ each.

In Arora's scheme: $O_\varepsilon(n)$

Note2:

OPT cannot contain edges longer than 1

Proof idea:

(i) Expected total cost of removed edges at most an ε-fraction of the optimal solution.

(ii) Derandomized random shift.

Sparse vs Dense

Case 1: opt. of subinstance has $\leq 1/\epsilon^7$ components

Slight adaption of Arora's scheme,
see [Asano–Katoh–Tamaki–Tokuyama STOC '97]

Case 2: opt. of subinstance has $> 1/\epsilon^7$ components

More interesting case

Dense Instances (Idea)

Start with an infinite capacity solution

Split instance into $\varepsilon^2 \times \varepsilon^2$ cells

Cut each component of the infinite capacity solution
into εk-vertex chunks,
associate each with a cell

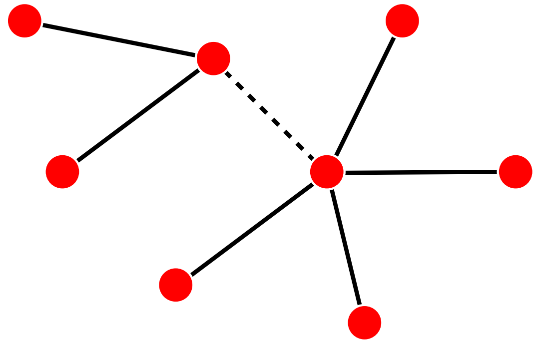
Connect chunks greedily but cheaply

Show:
(i) Existence of almost optimal “chunk-respecting” solution
(ii) Find good “chunk-respecting” solution”

$1\mathrm{AR}_F$

Return trees of sizes between εk and $5\varepsilon k$

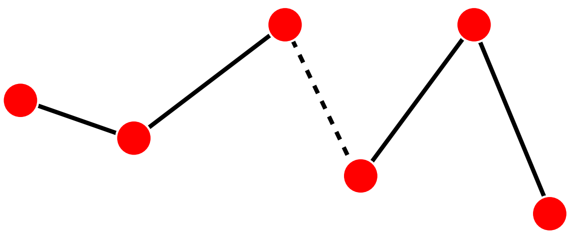
MST in Euclidean plane: degree at most 5 (kissing number)



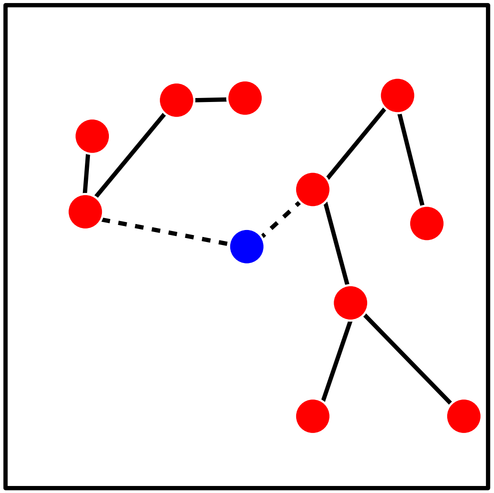
Collect the trees bottom-up

$1\mathrm{AR}_P$

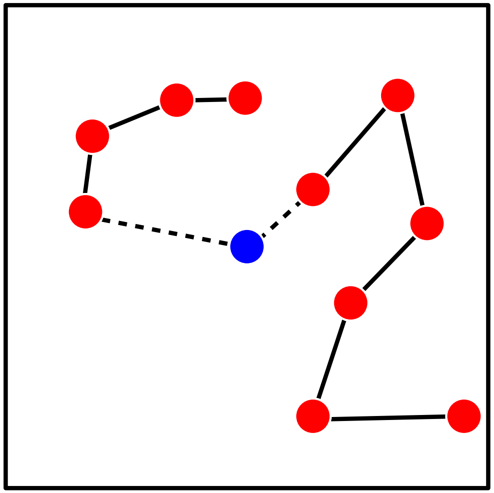
Cut each path into pieces of length εk



Associate Chunks with Cells



Forest:
pick an arbitrary vertex for each chunk



Paths:
Assign each endpoint of each chunk

Assembling cunks

Forest Case: greedily collect chunks for each cell

Path Case: follow a path, connect endpoint
to another endpoint in same cell

Proof Sketch

Polynomial running time + Feasibility ✓

Solutions for the uncapacitated case are lower bound (up to $(1+\varepsilon)$) for the optimal solution

Airport cost: Roughly as many airports as OPT,
"stuck" at most once per cell

Edge cost: Added at most $1/\varepsilon$ edges
each of cost at most $\varepsilon^2\sqrt{2}$;
each component has cost at least 1

Open Problems

Find PTAS for general versions AR$_F$ and AR$_P$
in Euclidean space, planar graphs, etc.

Other metrics

Further problems in the Airport and Railway framework?

Other special instance classes of AR$_P$ and AR$_F$?