Exact and approximation algorithms for geometric and capacitated set cover problems

Piotr Berman

Pennsylvania State University

Marek Karpinski

Bonn University

Andrzej Lingas

Lund University

The setting of the problem

there is a central point where we can place directional antennas
there are customer points
customers have bandwidth demands
a possible antenna can serve customers in an angle sector
it has a capacity bound.

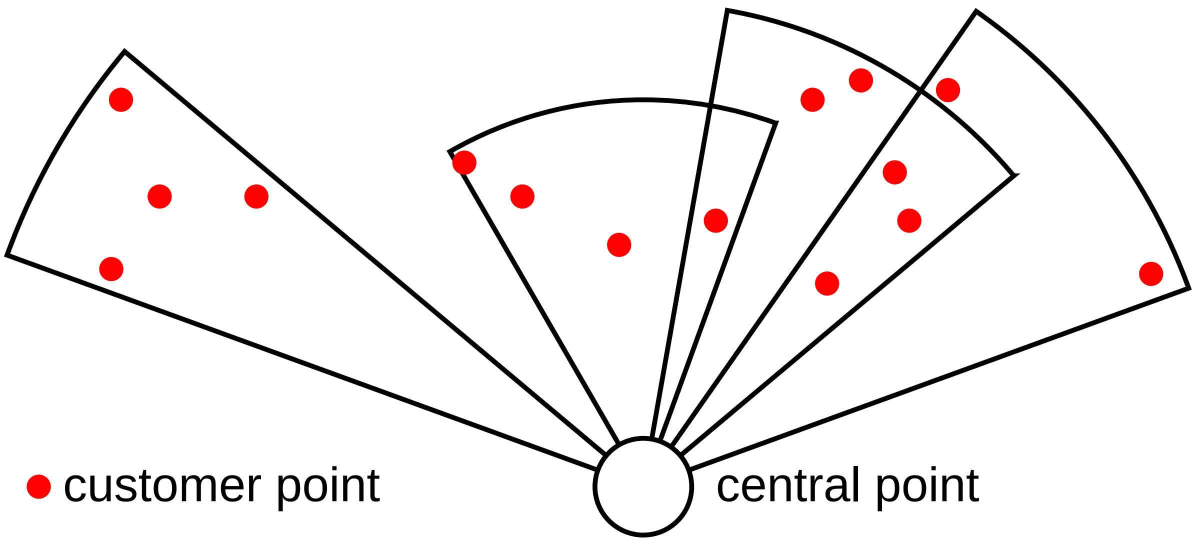
Fixed: capacity of the antennas
Minimize: the number of antennas

Fixed: the number of antennas
Minimize: maximum load of the antennas

The setting of the problem, details

An antenna can serve customers in an angle sector of variable width.

There is a trade-off: how far can an antenna reach and how wide the angle is. For example, the area of the angle sector can be constant.



Problem variations, results

Minimize number, or minimize load (Load)

Variable width (Var), constant width

negligible demands (U), demands matter (capacitated)

- MinAnt, 1.5 approximation, better implies P $=$ NP
- MinAntVar, we improve 3 to 2.357
- UMinAnt, easy exact solution
- UMinAntVar, we improve 2 to exact, $O(n^4)$ time and $O(n^2)$ space.
- MinAntLoad, we show a PTAS

Nature of MinAntVar problem: capacitated set cover

Instance: universe $\mathcal{U}$, family of subsets $\mathcal{S}$,
weight function $w : \mathcal{U} \rightarrow \mathbf{R}_+$.

Valid solution: a partition of $\mathcal{U}$ into $A_1, \ldots, A_k$ so that for each A_i

- there exists $B \in \mathcal{S}$ such that $A_i \subset B$
- $w(A_i) \leq 1$

Minimize: k.

If all subsets of $\mathcal{U}$ are in $\mathcal{S}$, this is Bin Packing Problem.

For many geometrically defined set familites, the Set Cover problem can be solved either exactly, or has a good approximation.

Our approach: a solution to Set Cover instance is converted to a solution to Capacitated, and the approx. factor increases by 1.357.

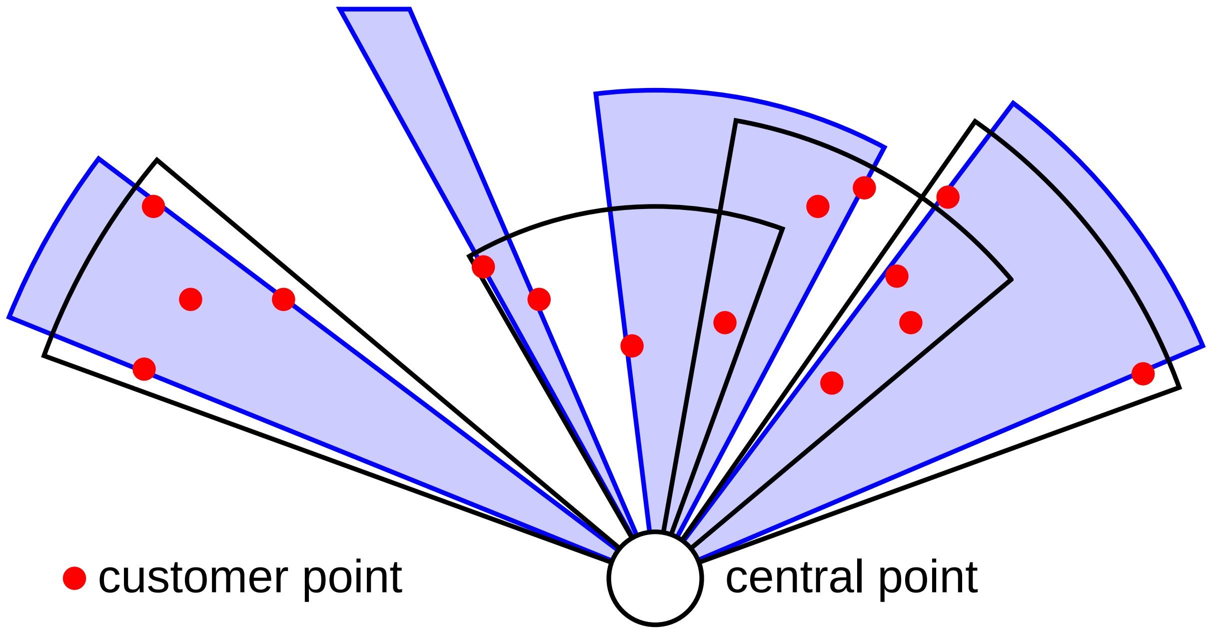
UMinAntVar: a dynamic programmings solution

We will look for the best normalized solution

1. Antennas specified by pairs of customer points that are covered by this antenna, only only by this antenna
   customers i, j with coordinates (θ_i, r_i) and $(\theta_j, r_j) \rightarrow$
   antenna with angle interval (θ_i, θ_j) and the furthest possible reach
2. A consequence: angle intervals of antennas are nested — disjoint, or one is subset of another

UMinAntVar: a dynamic programmings solution

Example of normalization: black → blue



Principle: if the solution is not normalized, we can shift boundaries

UMinAntVar: a dynamic programmings solution

Generic subproblem

consider the antenna A_{ij} specified by customers i and j

there is a set of customers B_{ij} within angular range of the antenna and outside its radial range (too far to reach)

task: cover B_{ij} with as few antennas as possible

basic case: $B_{ij} = \emptyset$, cost 0

non-basic case: use solved subproblems for the subranges to form a graph where solutions correspond to paths, find a shortest path.

UMinAntVar: a dynamic programmings solution

The graph of a non-basic subproblem

Nodes: the sequence of the customers to cover, from left to right and the extra END

After renumbering, Nodes $= \{0, 1, \ldots, m\}$

Edges: if an antenna can cover customers i, j, the problem specified by this antenna has an already computed cost, say k, and we define edge $(i, j+1)$ with cost $k+1$

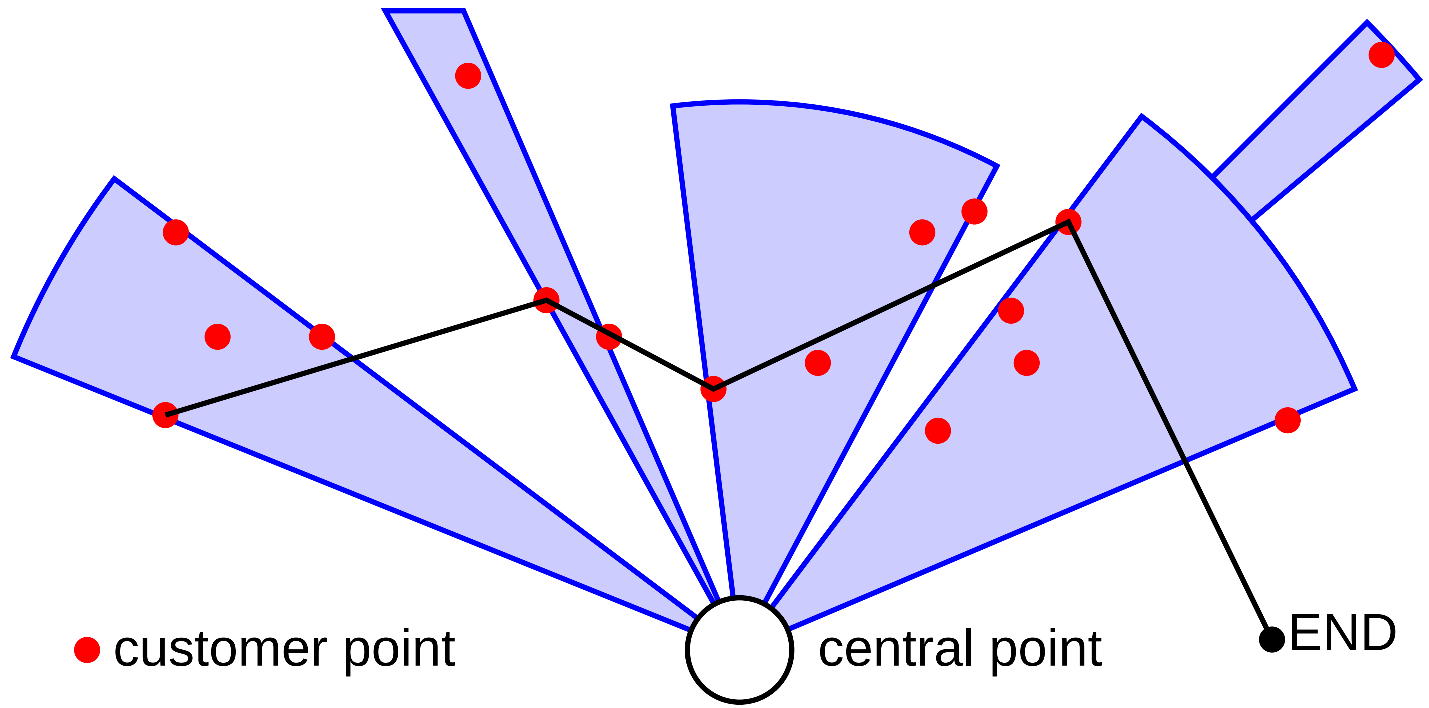
Solution: a shortest path from 0 to m, corresponds to the "outermost" antennas in the nesting of angle intervals

Solving a subproblem: the graph has at most n nodes and n^2 edges.

It is acyclic, so we can run Disjkstra's in $O(n^2)$ time.

UMinAntVar: a dynamic programmings solution

A path in the graph of subproblems (edges have costs 1,1,1, 2):



UMinAntVar: a dynamic programmings solution

Top subproblems

Suppose the cycle of customer angular positions is $(\theta_0, \ldots, \theta_{n-1})$

Assume that in an optimimum solution customer i defines an outermost interval on its left side

Then we can form a graph for the customer sequence $(i, i+1, \ldots, n-1, 0, \ldots, i-1, \text{END})$ and the optimum solution corresponds to a shortest path in this graph.

Thus we solve a top subproblem for every i, and the best of their solutions will be the optimum solution.

From UMinAntVar solution to MinAntVar

A solution obtain without regard for customer demands may assign to an antenna a set of customers with the sum of demands that exceeds 1, the antenna bandwidth capacity. In such a case we will add more antennas.

In doing that we just use a general method, i.e. one that could be applied for any family of sets.

The simplest approach is to consider the sets of the solutions of the UMinAntVar one at the time, $A_1, A_2, \ldots$.

As long as the sum of weights of elements of A_i exceeds 1, we "pack" another copy of A_i using FFD method (First Fit Decreasing).

This method can increase the number of sets/antennas by a factor about 1.6905.

From UMinAntVar solution to MinAntVar

Analysis of FFD

We can obtain a better approximation ratio by using FFD as the basis, and improve critical cases.

One can divide elements into weight classes:

class P_i has weights in the interval $\left(\frac{1}{i+1}, \frac{1}{i}\right]$

each u of P_i obtains slack $s(u) = \frac{1}{i} - \frac{1}{i+1}$, so $w(u) + s(u) > \frac{1}{i}$

For a set A from an optimum solution we can show
$w(A) + s(A) \leq 1.6905$

For a set B produced by FFD, we can “almost” show that
$w(B) + s(B) \geq 1$

From UMinAntVar solution to MinAntVar

Analysis of FFD, cont.

The largest possible slack of a set with weight 1:

weights: $\frac{1}{2} + \frac{1}{3} + \frac{1}{7} + \frac{1}{43} + \ldots$

slacks: $\frac{1}{2} + \frac{1}{6} + \frac{1}{42} + \frac{1}{42 \times 43} + \ldots$

It is very easy to show for a set that consists of elements of only one weight class: we can pack at least i elements of class P_i, and for each $u \in P_i$ we have $w(u) + s(u) \geq \frac{1}{i}$.

For other sets it takes a page of boring inequalities to show that deficits, if any, are smaller then surplusses.

Approach of our improvement: search for better ways to pack elements of P_1 and P_2 — the big slacks. Use the fact that an optimal set can contain at most two of these.

Interesting open problems

Our dynamic programming for UMinAntVar uses $O(n^4)$ time and $O(n^2)$ memory.

Mihail Patrascu posted a dynamic program that uses $O(n^3)$ time and $O(n^3)$ memory.

There is a "feeling" that there should be a yet more efficient algorithm.

Patrascu consider covering of points (x_i, y_i) such that $y_i \geq 0$ by rectangles of the form $[x_{low}, x_{high}] \times [0, \frac{1}{x_{high} - x_{low}}]$
i.e. oriented rectangles with area equal to 1 and adjacent to X-axis.

This problem is basically equivalent to MinAntVar.

Interesting open problems, n^3 dynamic programming

Assume that points are ordered by x-coordinate. Patrascu's subproblems are of the form

problem i, j, k: cover $\{x_\ell : i \leq \ell \leq j$ and $y_\ell \geq y_k\}$

ordering of problems:

first, from the smallest to largest $j - i$
second, from the smallest to largest y_k (assume uniqueness for simplicity)

basic case: we can cover with a single rectangle

basic case: y_k is minimal in the interval $i, \ldots, j$ (solve in linear time)
(either use the rectangle from i to j and a shorter problem, or a split into two shorter problems)

Interesting open problems, n^3 dynamic programming

normal case of problem i, j, k:
we already solved problem i, j, k' for the minimally smaller $y_{k'}$

possible solutions:

- the same, with the same cost
- different, hence, one that does not cover (x_k, y_k)

let $y_{k''}$ be minimally larger

check if the sum of costs of problems $i, k-1, k''$ and $k+1, j, k''$ is smaller