On the Power Assignment Problem in Radio Networks

Luzius Meisser

Betreuung: Thomas Moscibroda
Content

- Content
- Problem
- Paper, Results
- Trade-Off Hops vs. Power
- Complexity
- Conclusions
Content

- Content
- Problem ← now
- Paper, Results
- Trade-Off Hops vs. Power
- Complexity
- Conclusions
Problem

N stations in mobile network, minimize power consumption while preserving connectivity with at most h hops.

→ Min d-D h-Range Assignment
Problem

Definition:

- Nodes in d-dimensional space
- Power consumption = $f(\text{sending distance}), p = \alpha \cdot (d^\beta)$
- Minimize Total Power Consumption
- Constraint: max $h$ hops
  $\Rightarrow$ Min d-D $h$-Range Assignment
If $h = \infty$, Min d-D Range Assignment
Problem

Questions:
What’s the minimal total power consumption for a given $h$?
What’s the computational complexity of finding the optimal range assignment?
Are there good approximations?
Content

• Content
• Problem
• Paper, Results ← now
• Trade-Off Hops vs. Power
• Complexity
• Conclusions
On the power assignment problem in radio networks

Adrea E.F. Clementi, Paolo Penna, Riccardo Silvestri

Clementi – Penna – Silvestri
University of Rome - 2000
Result: Complexity

Paper says:

<table>
<thead>
<tr>
<th>Problem version</th>
<th>Previous results</th>
<th>Our results</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MIN 1D RANGE ASSIGNMENT</strong></td>
<td>in P [16]</td>
<td>–</td>
</tr>
<tr>
<td><strong>MIN 2D RANGE ASSIGNMENT</strong></td>
<td>in APX [16]</td>
<td>NP-complete</td>
</tr>
<tr>
<td><strong>MIN 2D h-RANGE ASSIGNMENT, well-spread</strong></td>
<td>–</td>
<td>in APX</td>
</tr>
<tr>
<td><strong>MIN 2D h-RANGE ASSIGNMENT</strong></td>
<td>–</td>
<td>in Av-APX</td>
</tr>
<tr>
<td><strong>MIN 3D RANGE ASSIGNMENT</strong></td>
<td>NP-complete, in APX [16]</td>
<td>APX-complete</td>
</tr>
</tbody>
</table>

APX: Class of NP Problems that are approximable to a constant factor in polynomial time.

[details later]
Result: Hops vs. Power

Paper proves:
Optimal 2D h-Range Assignment is in

$$\Theta (\delta(S)^2 |S|^{1+1/h})$$

What does that mean?
Result: Hops vs. Power

$$\Theta\left(\delta(S)^2 |S|^{1+1/h}\right)$$

S: Set of nodes
h: maximal number of hops
$$\delta(S): \text{minimal distance between two nodes}$$
Result: Hops vs. Power

Example:

\[ h = 1 \]
\[ \delta(S) = 1 \]
\[ |S| = n \]

\[ \Theta \left( \delta(S)^2 |S|^{1 + 1/h} \right) \]
Content

• Content
• Problem
• Paper, Results
• Trade-Off Hops vs. Power ← now
• Complexity
• Conclusions
## Hops vs. Power

<table>
<thead>
<tr>
<th>Problem version</th>
<th>Previous results</th>
<th>Our results</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{MIN 1D Range Assignment}</td>
<td>in P [16]</td>
<td>–</td>
</tr>
<tr>
<td>\textit{MIN 2D Range Assignment}</td>
<td>in APX [16]</td>
<td>\begin{itemize}</td>
</tr>
<tr>
<td>\textit{MIN 2D $h$-Range Assignment, well-spread}</td>
<td>–</td>
<td>\item NP-complete</td>
</tr>
<tr>
<td>\textit{MIN 2D $h$-Range Assignment}</td>
<td>–</td>
<td>\item in APX</td>
</tr>
<tr>
<td>\textit{MIN 3D Range Assignment}</td>
<td>NP-complete, in APX [16]</td>
<td>\item in Av-APX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\end{itemize}</td>
</tr>
</tbody>
</table>

\[14/73\]
Hops vs. Power

We want to show:
Optimal 2D h-Range Assignment is in

$$\Theta\left(\delta(S)^2 |S|^{1+1/h}\right)$$
Hops vs. Power

Oh-Notation

- Lower bound: $n^2$ is in $\Omega(n)$
- Upper bound: 1 is in $O(n)$
- Optimum: $n$ is in $\Theta(n)$ because $n$ is in both, in $\Omega(n)$ and $O(n)$
Hops vs. Power

Power assignment algorithm:

Set $h=2$, $n=256$,
Side of square: $l$
Hops vs. Power

Power assignment algorithm:

Set \( h = 2 \), \( n = 256 \),
Side of square: \( l \)
Divide area into \( k^2 \) subsquares, with \( k = n^{(1/2h)} \)
\[ \Rightarrow k = 4 \]
Hops vs. Power

Power assignment algorithm (centralized):

h = 2, n = 256
k = 4
→ 16 squares
Hops vs. Power

Power assignment algorithm (centralized):

h=2, n=256

Choose 1 station in each square and give it global transmission range.
Hops vs. Power

Power assignment algorithm (centralized):

$$h=2, \, n=256$$

→ All the blue nodes are connected.

Cost so far: $$l^2 \times k^2$$
Hops vs. Power

Power assignment algorithm (centralized):

h=2, n=256

Now recursively solve the problem in each subsquare with h decreased by 1.
Hops vs. Power

Power assignment algorithm (centralized):

Sub-Problem:
\[ h=1, \ n=16 \]

\[ \rightarrow \text{all nodes get a range of } l/k \]

\[ \rightarrow \text{cost of all subsquares:} \]
\[ k^2 \cdot (n/k^2) \cdot (l/k)^2 = n \cdot (l/k)^2 \]
Hops vs. Power

Power assignment algorithm (centralized):

→ All nodes are connected with at most 2 hops.

Total Cost:
\[ l^2 k^2 + n (l/k)^2 \]
\[ = l^2 (n^{1/2}) + (n^{1/2}) l^2 \]
\[ = 2 l^2 n^{1/2} \]

\[ \Theta(\delta(S)^2 \sqrt{|S|^{1+1/h}}) \]
Hops vs. Power

But what about this network?
Hops vs. Power

But what about this network?

\[ \text{cost is in } O(n^*(\delta(S)*n)^2) = O(n^3* \delta(S)^2) \]

\[ \Theta(\delta(S)^2 |S|^{1+1/h}) \]
Hops vs. Power

But what about this network?

- cost is in $O(n^* (\delta(S)^2 n)^2) = O(n^3 \delta(S)^2)$
- formula only holds for "well-spread" instances.

$$\Theta\left(\delta(S)^2 |S|^{1+1/h}\right)$$
Hops vs. Power

Well Spread: $D(S) = O(\delta(S) \cdot S^{1/2})$

→ idea: close to grid

Perfectly "well spread"

„well spread“ (obtained by moving the grid nodes a little)

Not „well spread“,

Randomly distributed on a square
Hops vs. Power

For well spread instances:
\[ \Theta \left( \delta(S)^2 \frac{|S|^{1+1/h}}{n^{2h}} \right) \]

For instances that are randomly distributed on a square:
\[ \Theta \left( l^2 n^{1/h} \right) \]
[With high probability]
Content

• Content
• Problem
• Paper, Results
• Trade-Off Hops vs. Power

• Complexity
• Conclusions
Intermezzo

Flashback

Paper says:

<table>
<thead>
<tr>
<th>Problem version</th>
<th>Previous results</th>
<th>Our results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min 1D Range Assignment</td>
<td>in P [16]</td>
<td>–</td>
</tr>
<tr>
<td>Min 2D Range Assignment</td>
<td>in APX [16]</td>
<td>NP-complete</td>
</tr>
<tr>
<td>Min 2D h-Range Assignment, well-spread</td>
<td>–</td>
<td>in APX</td>
</tr>
<tr>
<td>Min 2D h-Range Assignment</td>
<td>–</td>
<td>in Av-APX</td>
</tr>
<tr>
<td>Min 3D Range Assignment</td>
<td>NP-complete, in APX [16]</td>
<td>APX-complete</td>
</tr>
</tbody>
</table>

APX: Class of NP Problems that are approximable to a constant factor in polynomial time.

[details later]
Intermezzo

Complexity Classes

Short overview over the classes P, NP, APX, as well as over the concepts of Hardness and Completeness.
Intermezzo

Complexity Classes

P: Class of Problems that can be solved in polynomial time.
Intermezzo

Complexity Classes

NP (or NPO): Class of Problems whose objective function can be calculated in polynomial time.
Intermezzo

Complexity Classes

Example of an NP problem:
Min Vertex Cover

Given a graph:
Color the minimal number of vertices blue such that every edge is connected to a blue vertex.
Intermezzo
Complexity Classes

Example of an NP problem:
Min Vertex Cover

cover

no cover
Intermezzo

Complexity Classes

APX: Class of NP Problems that are approximable to a constant factor in polynomial time.

Examples: Min Vertex Cover restricted to cubic graphs, Travelling Salesman in Euclidean Space
Intermezzo
Complexity Classes

NP

APX

P
Intermezzo

Complexity Classes

Hardness/Completeness

NP

NP-Hard

APX

P

NP-Complete
**Intermezzo**

Complexity Classes

Hardness/Completeness

- **NP**
- **APX**
- **APX-Hard**
- **APX-Complete**
- **P**
Intermezzo

Complexity Classes

Hardness/Completeness

- NP
- APX
- P
- NP-Hard
- APX-Hard
- NP-Complete
- APX-Complete
Intermezzo

Complexity Classes

Recipe to prove APX-Completeness of a problem A:

- show that A is in APX by giving an approximation algorithm
- show that A is APX-Hard by reducing it to another problem B that is known to be APX-Hard
- since A is in APX and APX-Hard, it follows that A is APX-Complete

(Replace „APX“ by „NP“ for NP-Completeness)
Intermezzo

Complexity Classes

Reducibility:

A is reducible to B if: given an polynomial time algorithm that solves instances of A, we can provide a polynomial time algorithm that solves instances of B.

Additional when reducing APX problems: Show that a constant approximation factor is preserved.
Content

- Content
- Problem
- Paper, Results
- Hop-Power Trade-Off
- Complexity of Min 3D RA \( \leftarrow \) now
- Conclusions
Complexity Proof

APX-Completeness of Min 3D RA

1. Step according to recipe:
Show that „Min 3D Range Assignment“ is in APX.
Complexity Proof

APX-Completeness of Min 3D RA

1. Step according to recipe:
Show that „Min 3D Range Assignment“ is in APX.

This has already been done by Kirousis et al.
→ We believe them, so we can proceed to step 2, hehe. 😊
Complexity Proof

APX-Completeness of Min 3D RA

2. Step according to recipe:

Reduce „Min 3D Range Assignment“ to a problem which is known to be APX-Hard.

We pick „Min Vertex Cover restricted to cubic graphs“
Complexity Proof

APX-Completeness of Min 3D RA
Complexity Proof

APX-Completeness of Min 3D RA

1. Each edge in Vertex Cover is replaced by a chain of radio stations

\[ \delta \]
Complexity Proof

APX-Completeness of Min 3D RA

2. We add a „gadget“ to each chain

Gadget

3

\[ \delta + \delta \]

\[ \delta \]

\[ \delta \]
Complexity Proof

APX-Completeness of Min 3D RA

Optimal power assignment?
Complexity Proof

APX-Completeness of Min 3D RA

Intuitive (and correct) assignment
Complexity Proof

APX-Completeness of Min 3D RA

Equivalent Solution:
Complexity Proof

APX-Completeness of Min 3D RA

What would this graph look like when converted?

```
   o -- o -- o
```
Complexity Proof

APX-Completeness of Min 3D RA
Complexity Proof

APX-Completeness of Min 3D RA

Optimal Solution: Candidate A
Complexity Proof

APX-Completeness of Min 3D RA

Optimal Solution: Candidate B
Complexity Proof

APX-Completeness of Min 3D RA

We have implicitly found the Min Vertex Cover by solving the Range Assignment Problem.
Complexity Proof

APX-Completeness of Min 3D RA

After having converted the graph, can we guarantee that we are still only a constant factor away from the optimal solution?

\[ \text{solRA} = \text{solVC} \times (\delta + \varepsilon)^2 + m \times \varepsilon^2 + n \times (\delta + \varepsilon)^2 \]
Complexity Proof

APX-Completeness of Min 3D RA

\[ \text{apxRA} = \text{apxVC}^* (\delta + \epsilon)^2 + m^* \epsilon^2 + n^* (\delta + \epsilon)^2 \]

Can be made smaller than any given constant.
Complexity Proof

APX-Completeness of Min 3D RA

\[ \text{apxRA} = \text{apxVC} \times c^2 + 1 + n \times c^2 \]

Between \( \text{apxVC} \) and \( 3 \times \text{apxVC} \)
Complexity Proof

**APX-Completeness of Min 3D RA**

\[
apxRA = c_3 \ast apxVC \ast c_2 + 1
\]

→ changing \(apxRA\) by a constant factor also changes \(apxVC\) by a constant factor.
Complexity Proof

APX-Completeness of Min 3D RA

\[ \text{apxRA} = c_3 \times \text{apxVC} \times c_2 + 1 \]

The paper proves this in a correct way and concludes that:

\[ f_{VC} = 5 \times f_{RA} - 4 \]
### Complexity Proof

#### Flashback

<table>
<thead>
<tr>
<th>Problem version</th>
<th>Previous results</th>
<th>Our results</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MIN 1D RANGE ASSIGNMENT</strong></td>
<td>in P [16]</td>
<td>–</td>
</tr>
<tr>
<td><strong>MIN 2D RANGE ASSIGNMENT</strong></td>
<td>in APX [16]</td>
<td>NP-complete</td>
</tr>
<tr>
<td><strong>MIN 2D $h$-RANGE ASSIGNMENT, well-spread</strong></td>
<td>–</td>
<td>in APX</td>
</tr>
<tr>
<td><strong>MIN 2D $h$-RANGE ASSIGNMENT</strong></td>
<td>–</td>
<td>in Av-APX</td>
</tr>
<tr>
<td><strong>MIN 3D RANGE ASSIGNMENT</strong></td>
<td>NP-complete, in APX [16]</td>
<td>APX-complete</td>
</tr>
</tbody>
</table>
### Complexity Proof

#### Flashback

<table>
<thead>
<tr>
<th>Problem version</th>
<th>Previous results</th>
<th>Our results</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIN 1D RANGE ASSIGNMENT</td>
<td>in P [16]</td>
<td>–</td>
</tr>
<tr>
<td>MIN 2D RANGE ASSIGNMENT</td>
<td>in APX [16]</td>
<td>NP-complete</td>
</tr>
<tr>
<td>MIN 2D $h$-RANGE ASSIGNMENT, well-spread</td>
<td>–</td>
<td>in APX</td>
</tr>
<tr>
<td>MIN 2D $h$-RANGE ASSIGNMENT</td>
<td>–</td>
<td>in Av-APX</td>
</tr>
<tr>
<td>MIN 3D RANGE ASSIGNMENT</td>
<td>NP-complete, in APX [16]</td>
<td>APX-complete</td>
</tr>
</tbody>
</table>

Same proof for Min 2D Range Assignment?
Complexity Proof

Flashback

<table>
<thead>
<tr>
<th>Problem version</th>
<th>Previous results</th>
<th>Our results</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIN 1D RANGE ASSIGNMENT</td>
<td>in P [16]</td>
<td>–</td>
</tr>
<tr>
<td>MIN 2D RANGE ASSIGNMENT</td>
<td>in APX [16]</td>
<td>NP-complete</td>
</tr>
<tr>
<td>MIN 2D (h)-RANGE ASSIGNMENT, well-spread</td>
<td>–</td>
<td>in APX</td>
</tr>
<tr>
<td>MIN 2D (h)-RANGE ASSIGNMENT</td>
<td>–</td>
<td>in Av-APX</td>
</tr>
<tr>
<td>MIN 3D RANGE ASSIGNMENT</td>
<td>NP-complete, in APX [16]</td>
<td>APX-complete</td>
</tr>
</tbody>
</table>

Same proof for Min 2D Range Assignment?

-> only for NP-Completeness
Content

• Content
• Problem
• Paper, Results
• Hop-Power Trade-Off
• Complexity
• Conclusions
Conclusions

No „Min 2D h-Range Assignment“ algorithm will ever consume less energy than $O(n^{(1+1/h)})$
Conclusions

Relevance of paper:
Superficial measurement: has been cited in 12 papers so far (all self-citations) -> low impact.
But: We can now judge the quality of distributed algos better, since we know the optimum.
Conclusions

My impression:
- A provably wrong statement
- A prove we did not understand
→ I do not entirely trust every detail in the paper (e.g. does it really work for all betas?)
Open Questions

Is „Min 2D h-Range Assignment“ APX-Complete?
Distributed Algorithm?
Not much known about „Min d-D h-Range Assignments“ in general, even for the 1 dimensional case (is it in P? in NP? In APX?)
→ maybe in newer papers of Clementi et al.
Questions