Randomized heuristics for handover minimization

Mauricio G. C. Resende
AT&T Research

Joint work with Luis Morán-Mirabal, José Luis González-Velarde, and Ricardo M. A. Silva

Talk given at INFORMS 2012
Phoenix - Arizona
October 15, 2012

Outline

- Handover minimization problem (HMP)
- Integer programming formulation
- Special case of generalized quadratic assignment problem
- GRASP with evolutionary path-relinking for HMP
- Two other randomized heuristics for HMP
- Experiments

Paper

RANDOMIZED HEURISTICS FOR HANDOVER MINIMIZATION IN MOBILITY NETWORKS

L.F. MORÁN-MIRABAL, J.L. GONZÁLEZ-VELARDE, M.G.C. RESENDE, AND R.M.A. SILVA

ABSTRACT. A mobile device connects to the cell tower (base station) from which it receives the strongest signal. As the device moves it may connect to a series of towers. The process in which the device changes the base station it is connected to is called handover. A cell tower is connected to a radio network controller (RNC) which controls many of its operations, including handover. Each cell tower handles an amount of traffic and each radio network controller has capacity to handle a maximum amount of traffic from all base stations connected to it. Handovers between base stations connected to different RNCs tend to fail more often than handovers between base stations connected to the same RNC. Handover failures result in dropped connections and therefore should be minimized. The HANDOVER MINIMIZATION PROBLEM is to assign towers to RNCs such that RNC capacity is not violated and the number of handovers between base stations connected to different RNCs is minimized. We describe an integer programming formulation for the handover minimization problem and show that state-of-the-art integer programming solvers can solve only very small instances of the problem. We propose several randomized heuristics for finding approximate solutions of this problem, including a GRASP with path-relinking for the generalized quadratic assignment problem, a GRASP with evolutionary path-relinking, and a biased random-key genetic algorithm. Computational results are presented.

1. INTRODUCTION

A cellular (or mobility) network consists of fixed base stations (cell towers) and mobile transceivers (e.g., mobile phones and tablet computers). A radio signal between the mobile transceiver and the base station allows communication between the transceiver and other transceivers as well as with other devices in the network. Each base station covers an area called a *cell*. As a mobile transceiver moves between cells, it may need to connect over time to several base stations. The transfer of connection from one base station to another is called a *handover*.

Each base station is controlled by a *radio network controller* or RNC. Each base station is connected to one RNC. The amount of traffic between transceivers and each base station depends strongly on the location of the base station. For example, a base station located in a city center will usually have more traffic than one located in a rural area distant from the city center. Each RNC can handle a maximum amount of traffic. This constraint limits the subsets of base stations that can connect to each RNC.

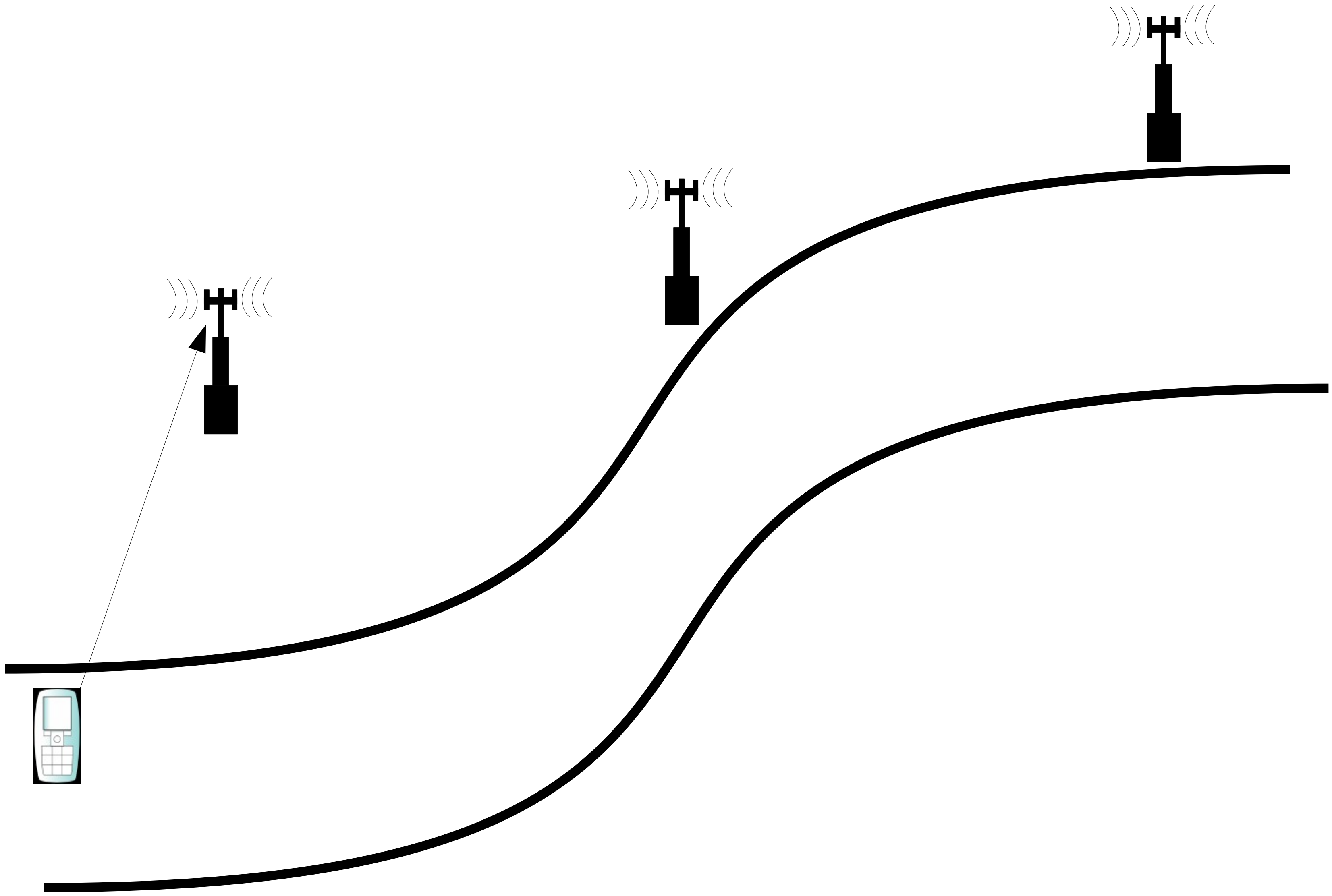
Date: August 2, 2012.
Key words and phrases. Mobility networks, handover minimization, randomized heuristics, GRASP, biased random-key genetic algorithm.
AT&T Labs Research Technical Report.

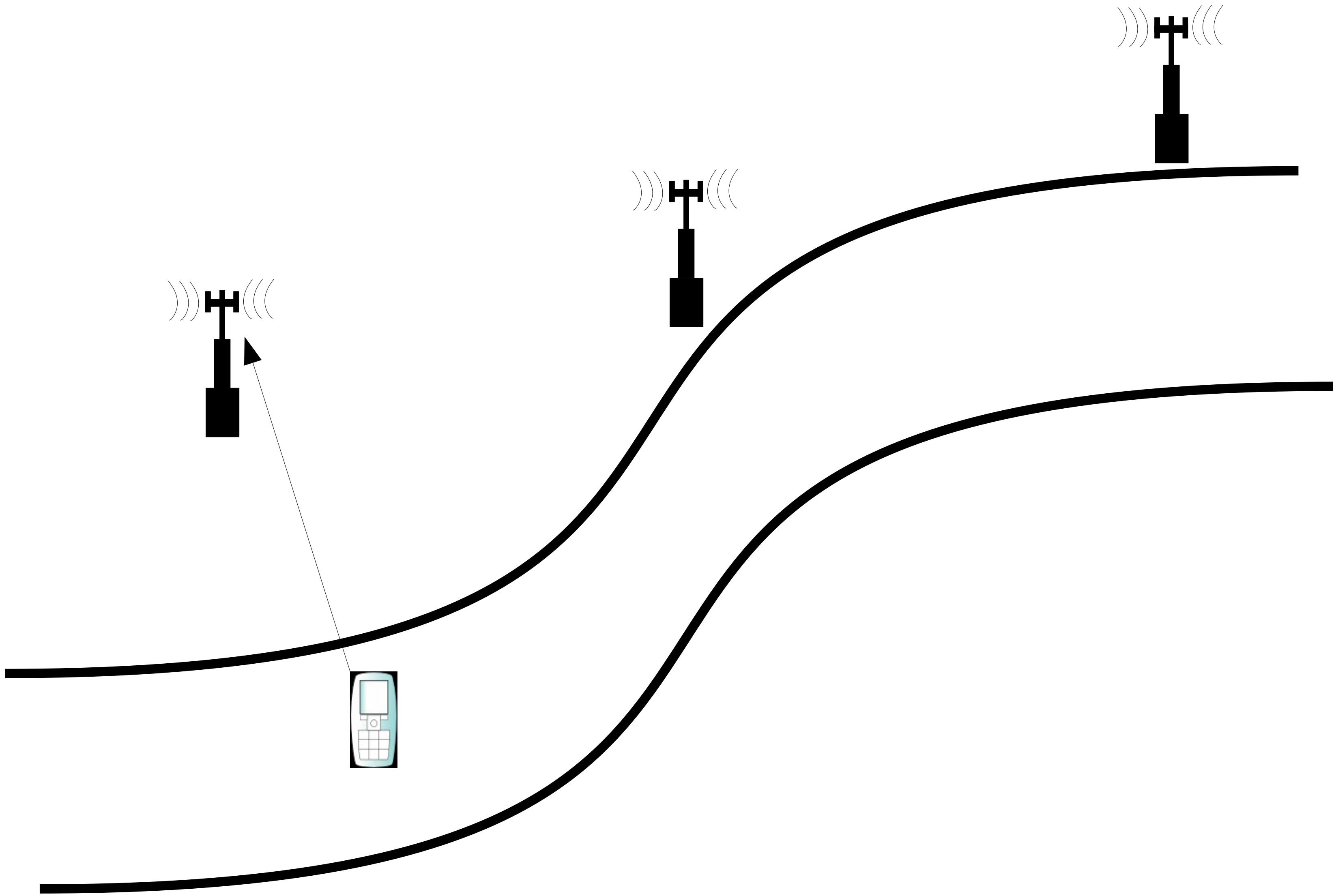
1

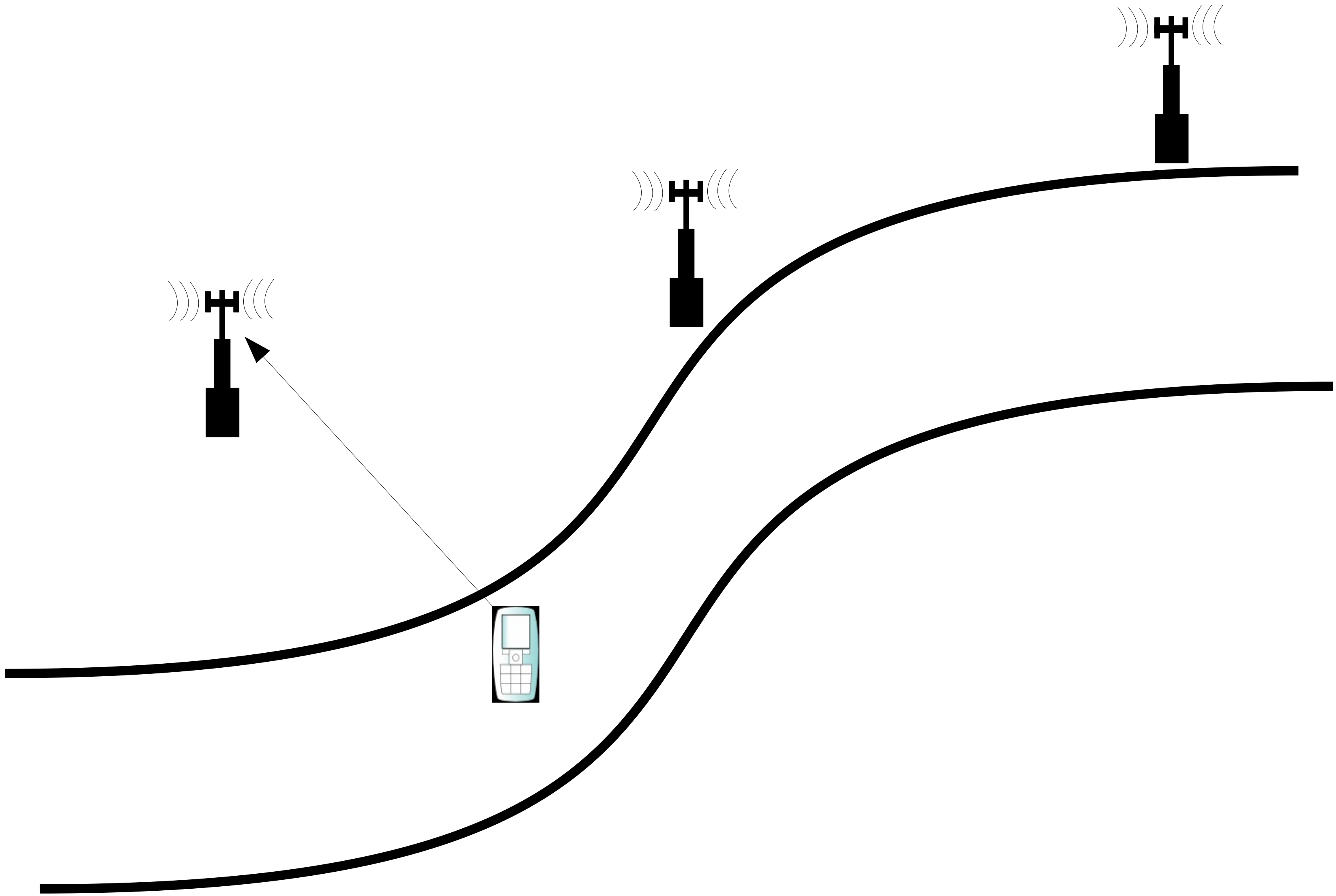
L.F. Morán-Mirabal, J.L. González-Velarde, MGCR, and R.M.A. Silva, "Randomized heuristics for handover minimization in mobility networks", AT&T Labs Research Technical Report, August 2012

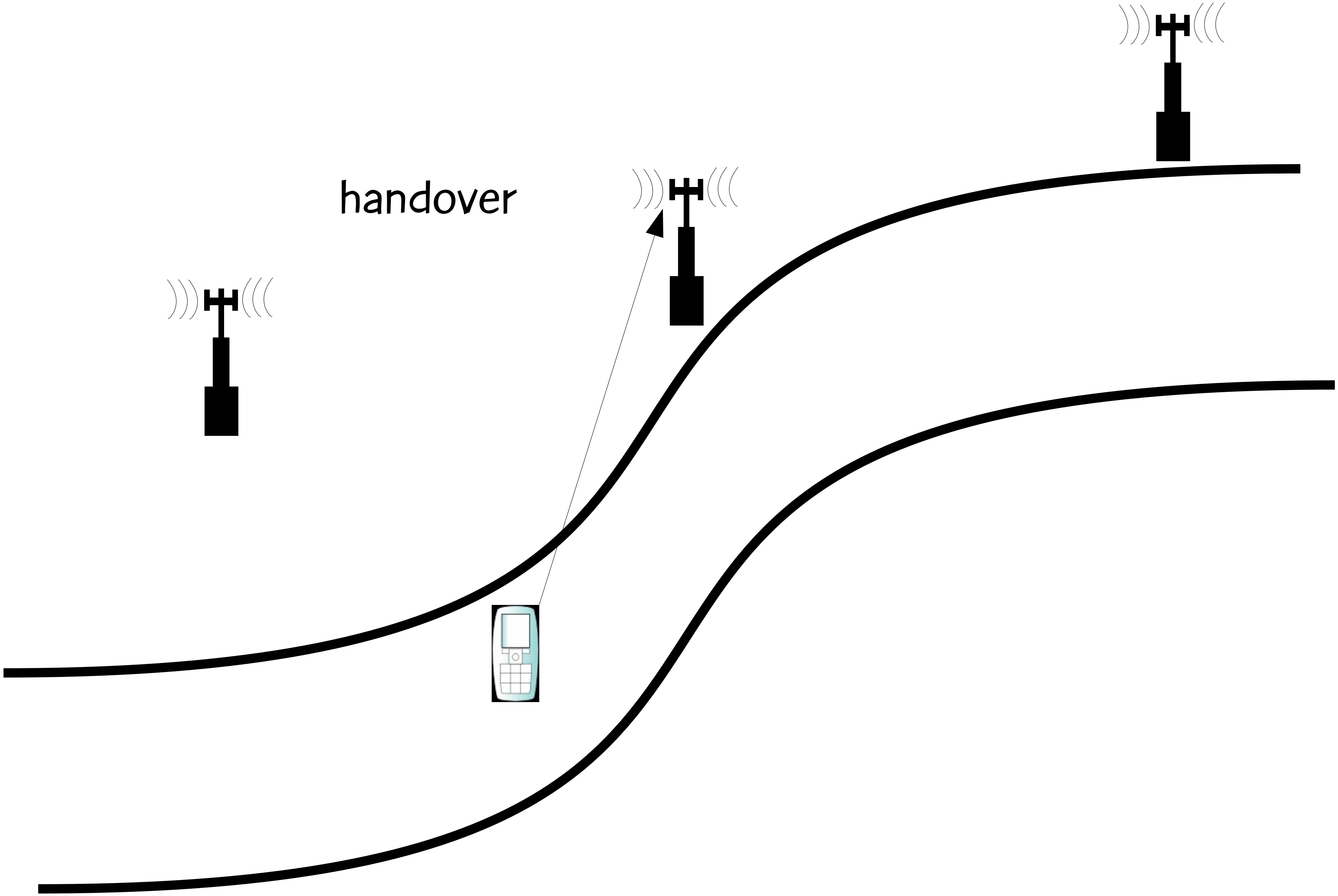
http://www.research.att.com/~mgcr/doc/randh-mhp.pdf

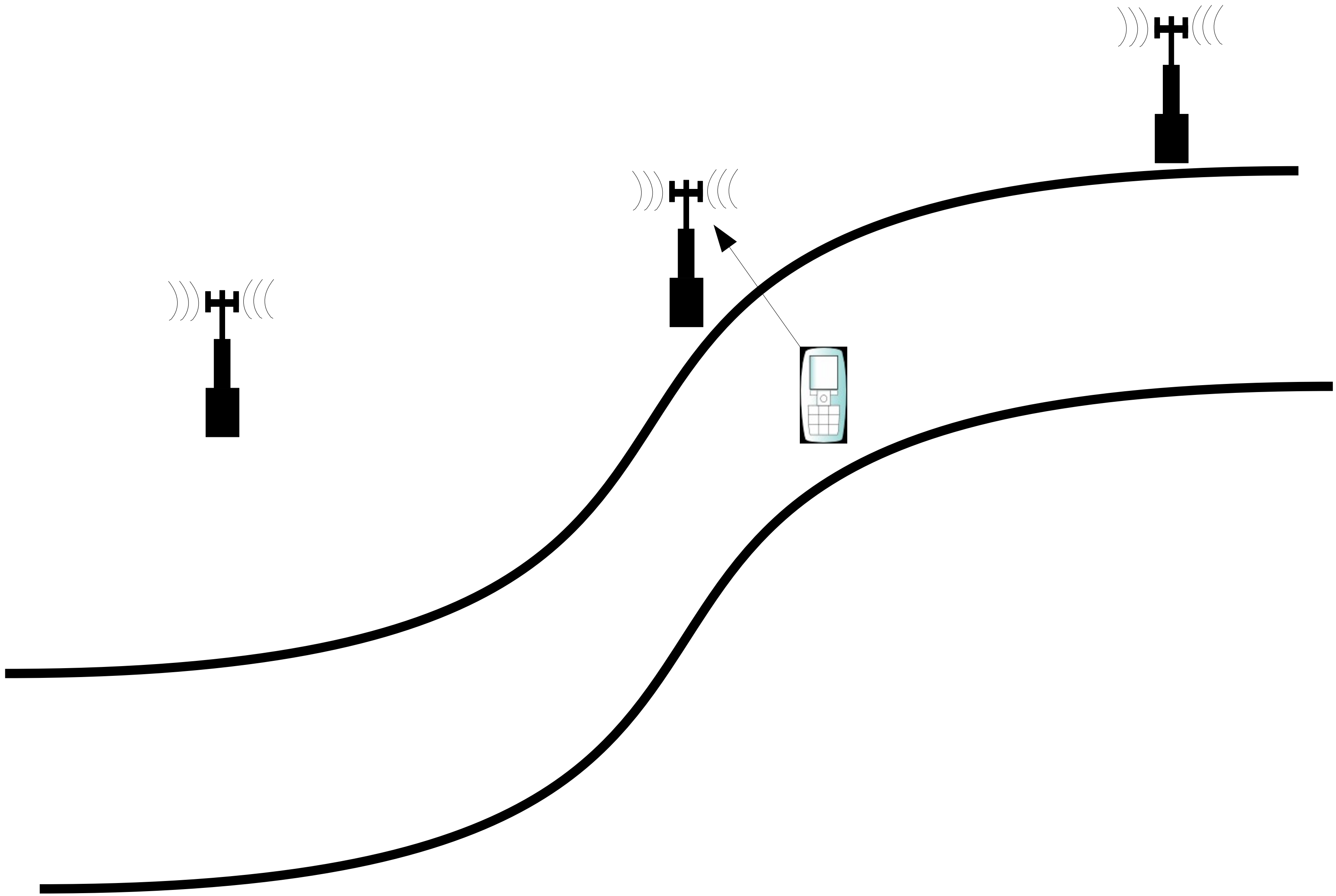
Handover minimization

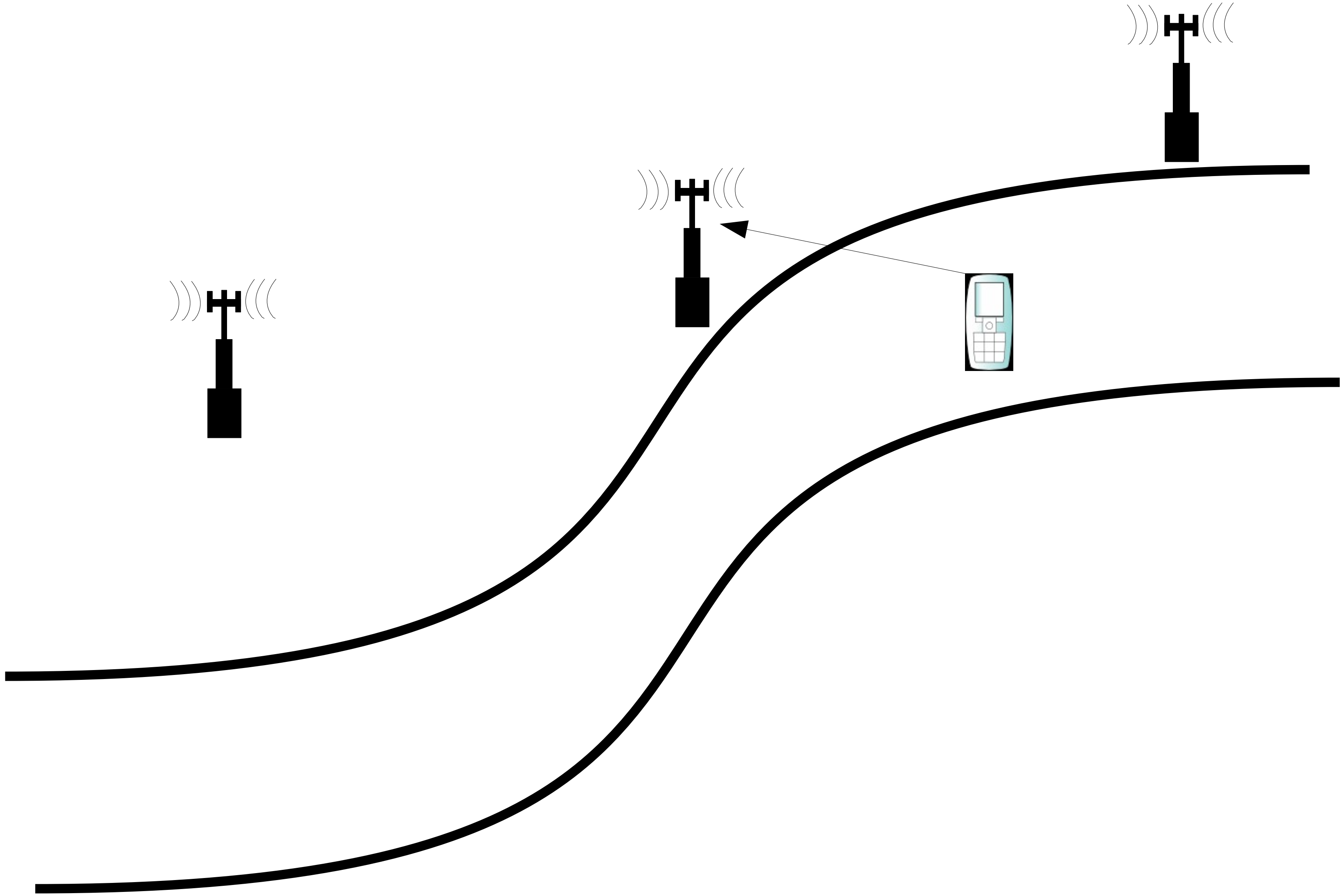


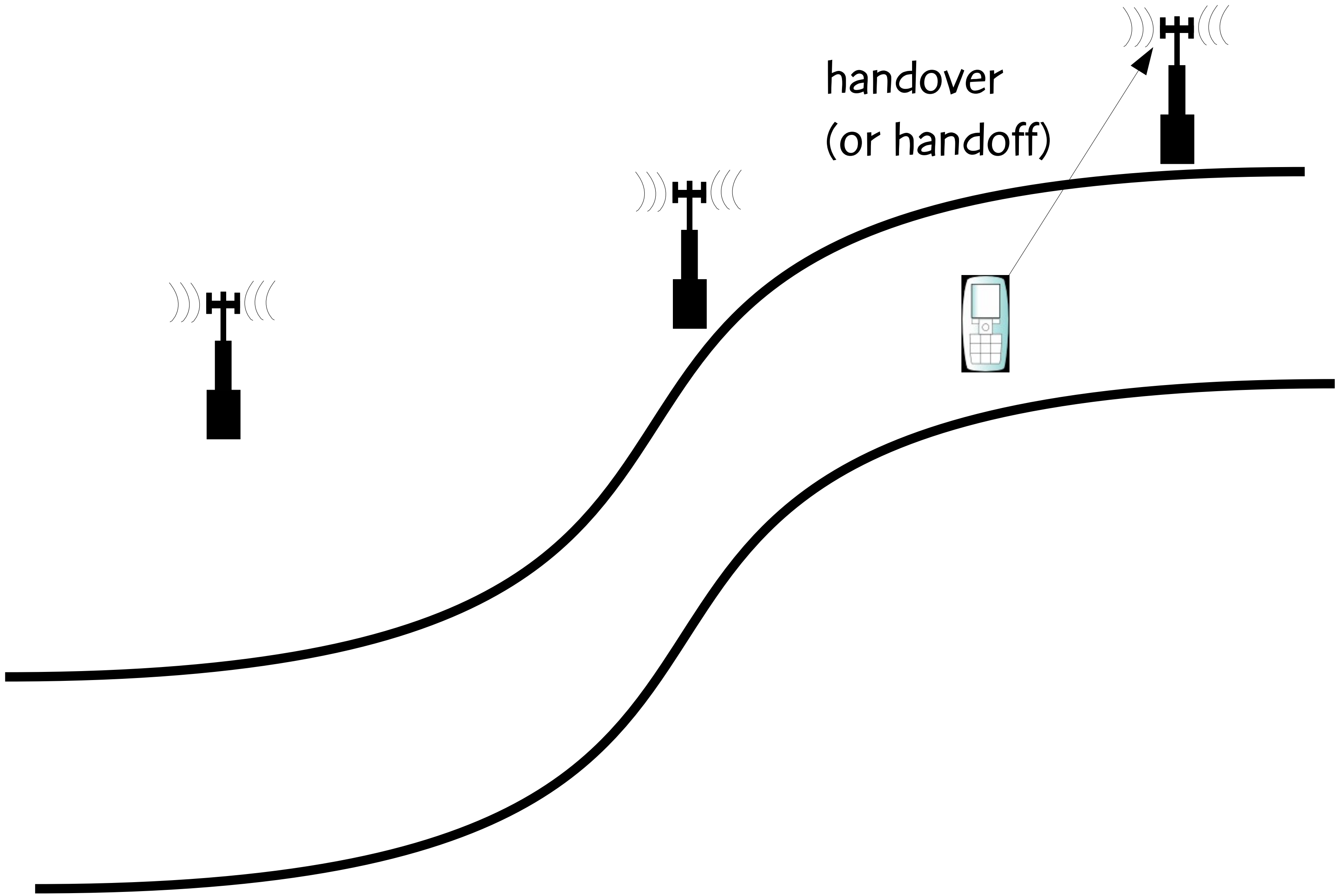


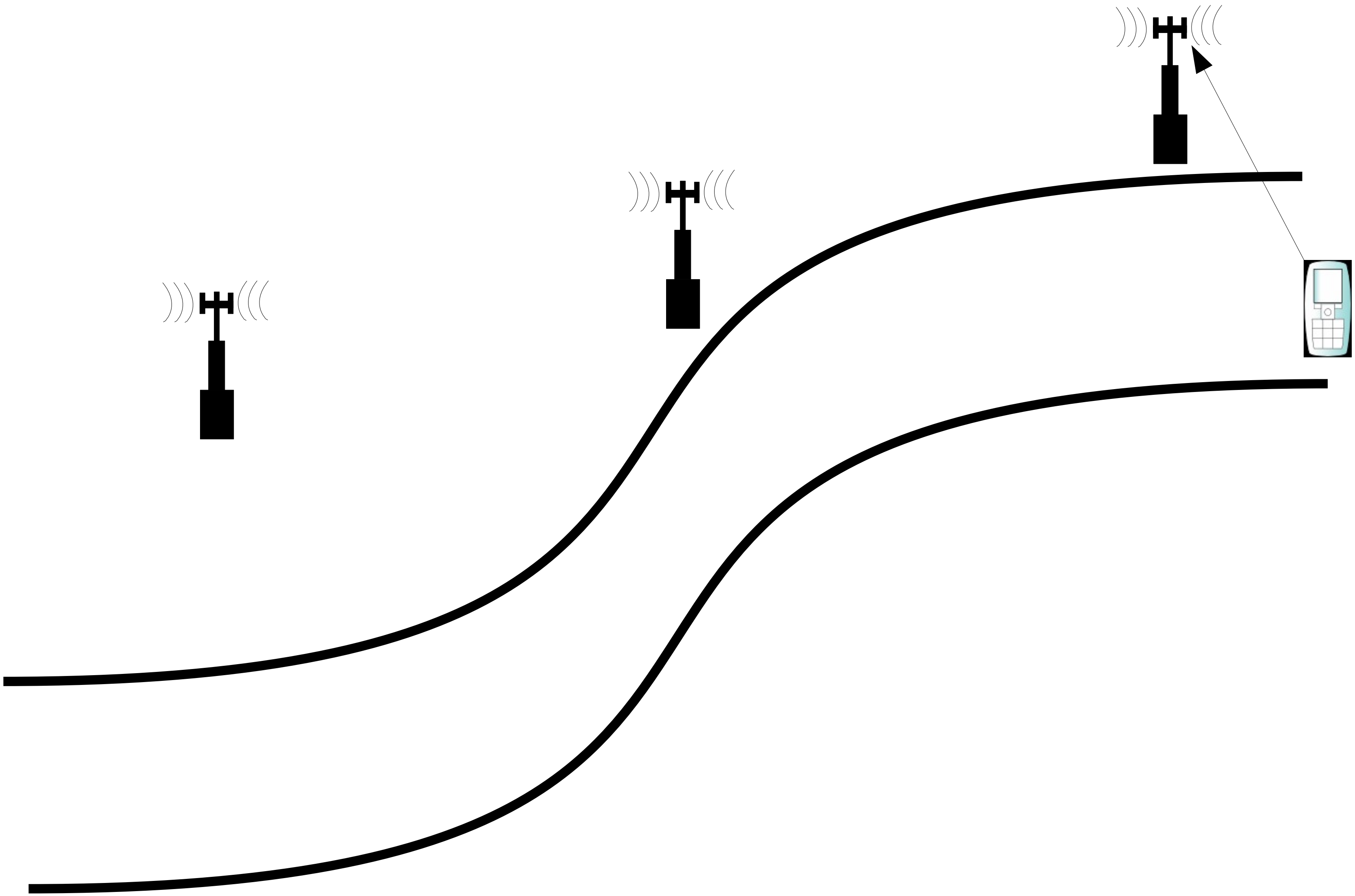


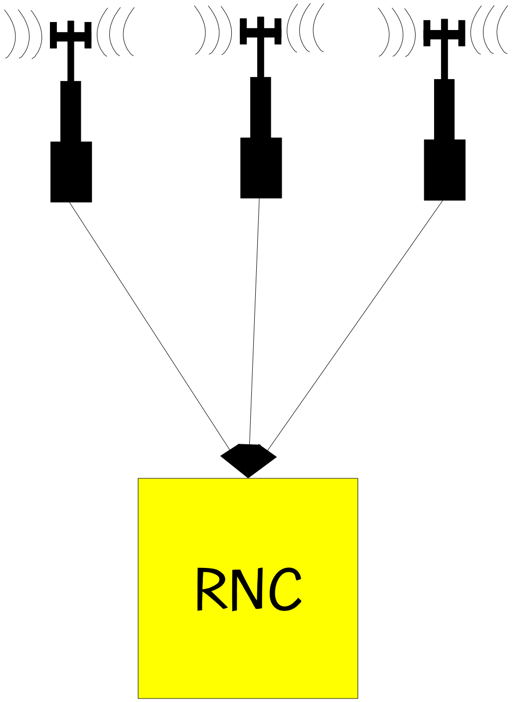

handover



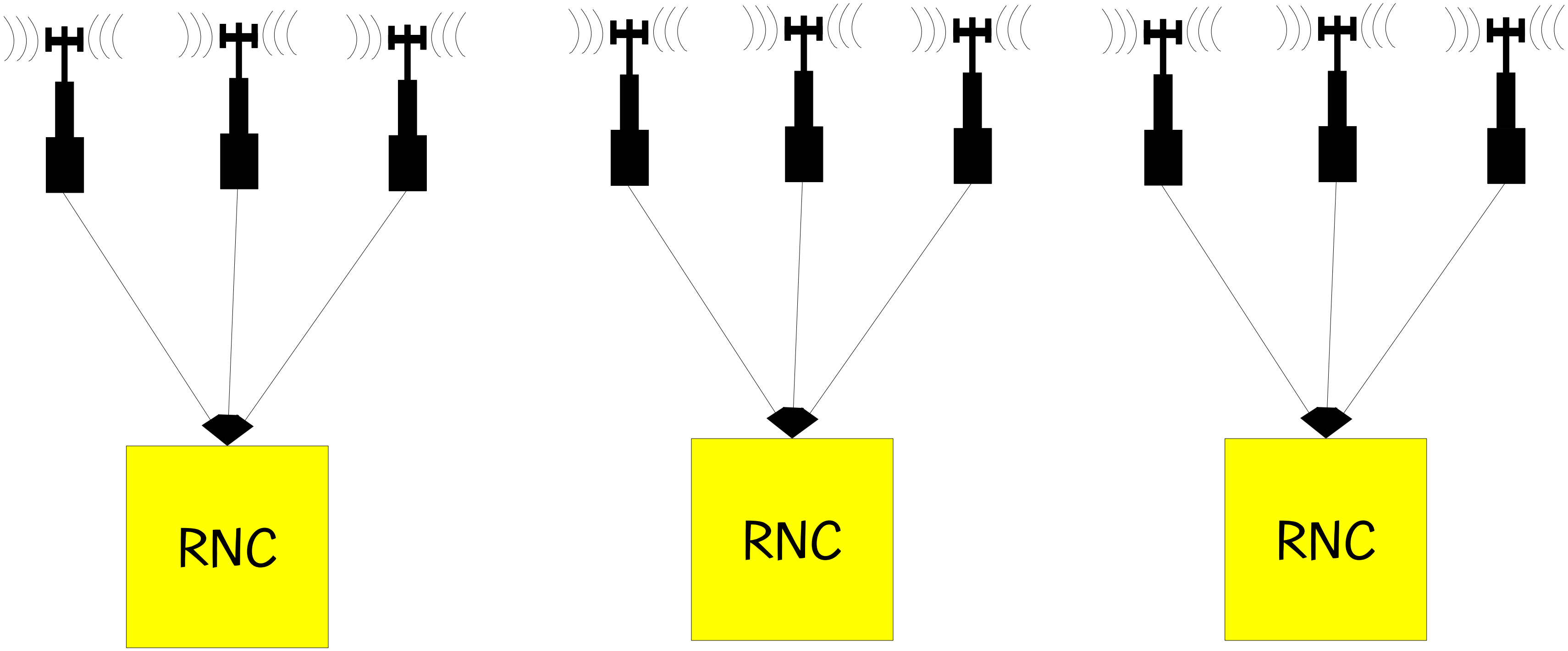



handover
(or handoff)

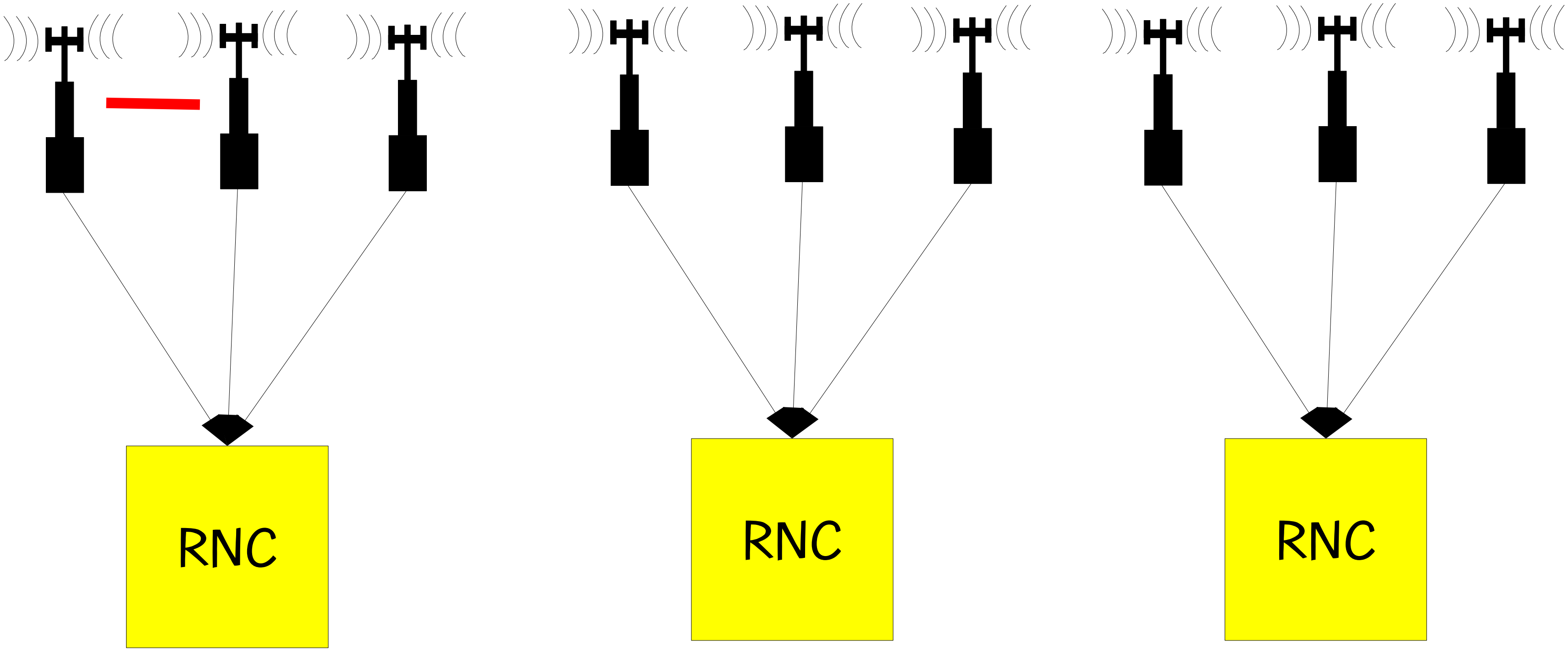




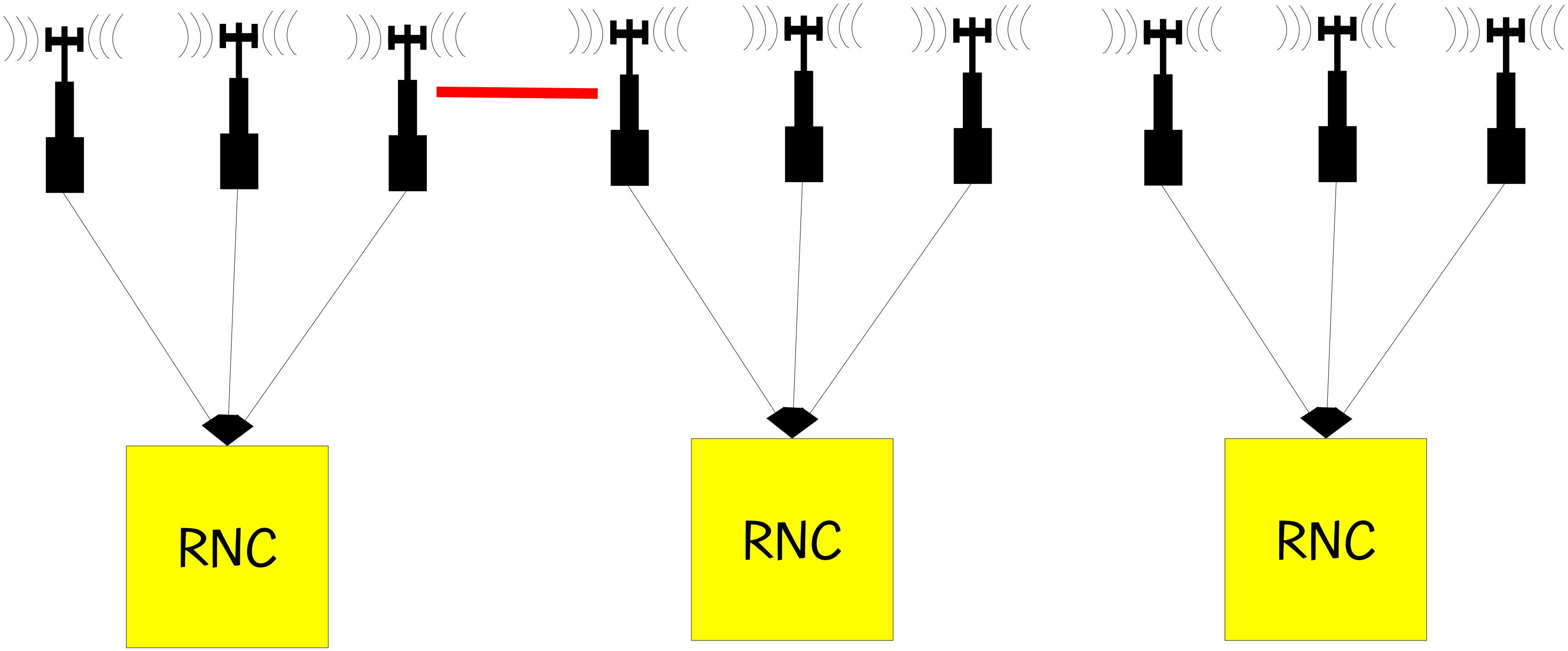
- Each cell tower has associated with it an amount of traffic.
- Each cell tower is connected to a Radio Network Controller (RNC).
- Each RNC can have one or more cell towers connected to it.
- Each RNC can handle a given amount of traffic … this limits the subsets of cell towers that can be connected to it.
- An RNC controls the cell towers connected to it.



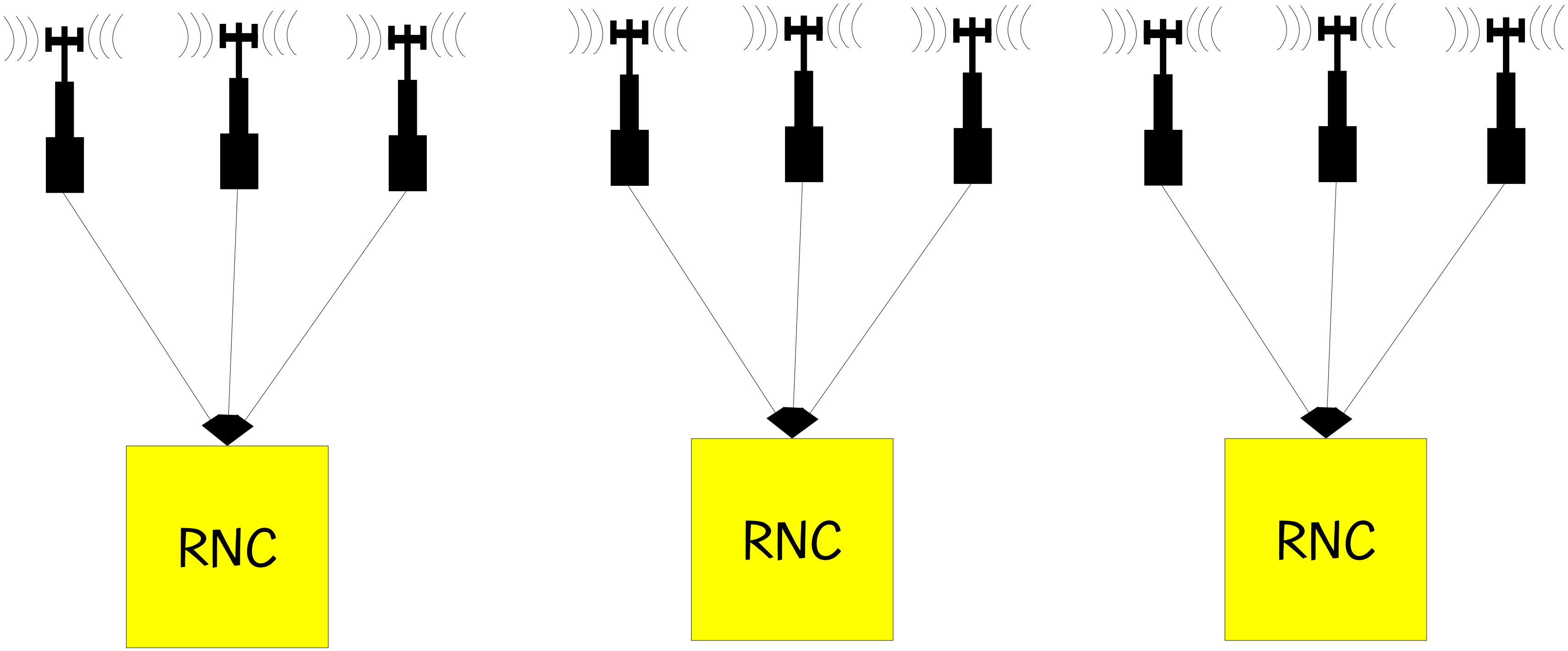
- Handovers can occur between towers



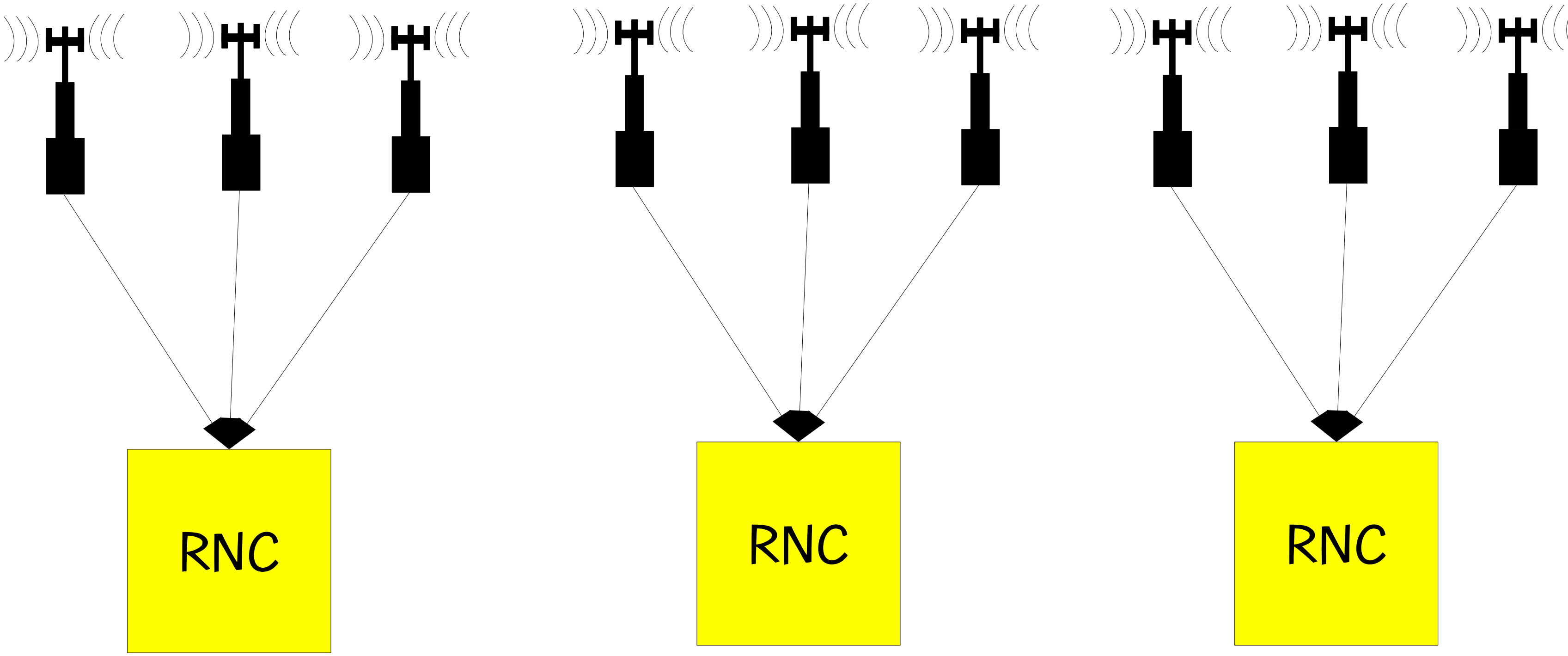
- Handovers can occur between towers
  - connected to the same RNC



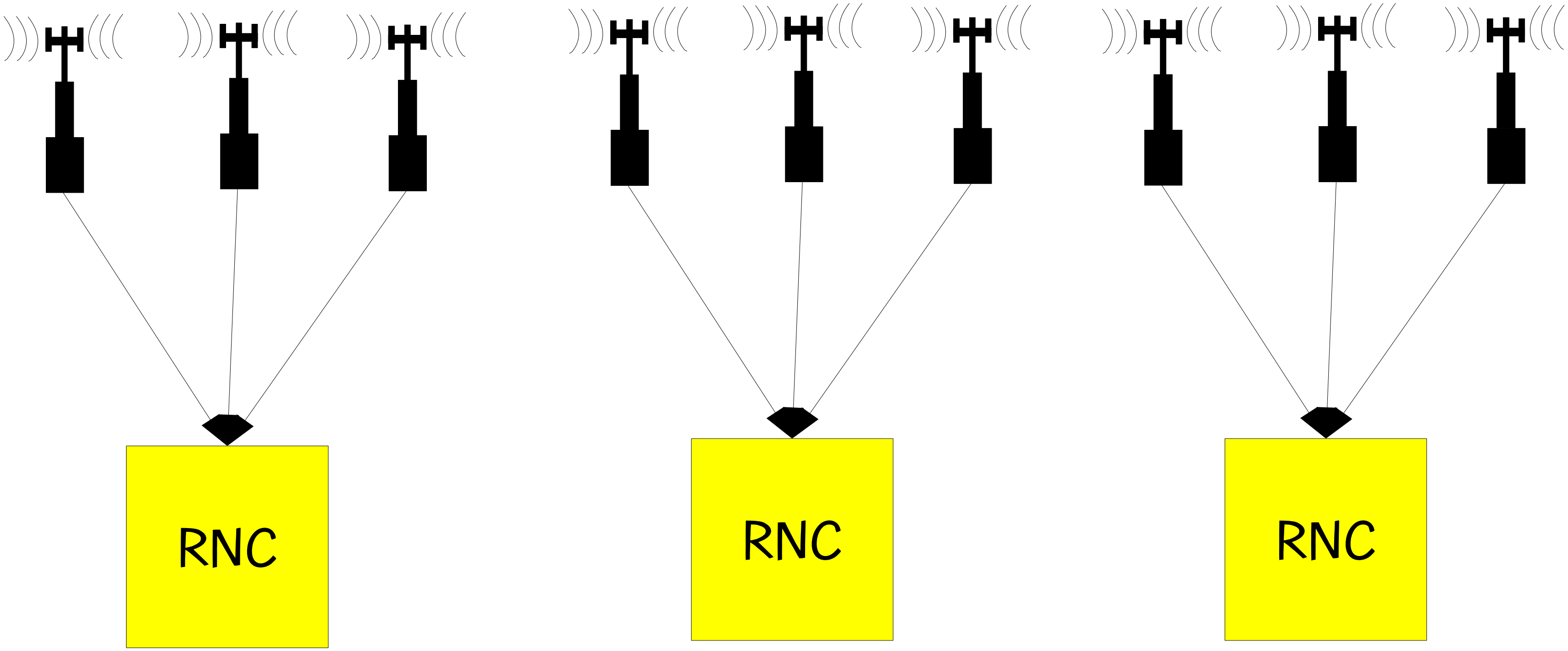
- Handovers can occur between towers
  - connected to the same RNC
  - connected to different RNCs



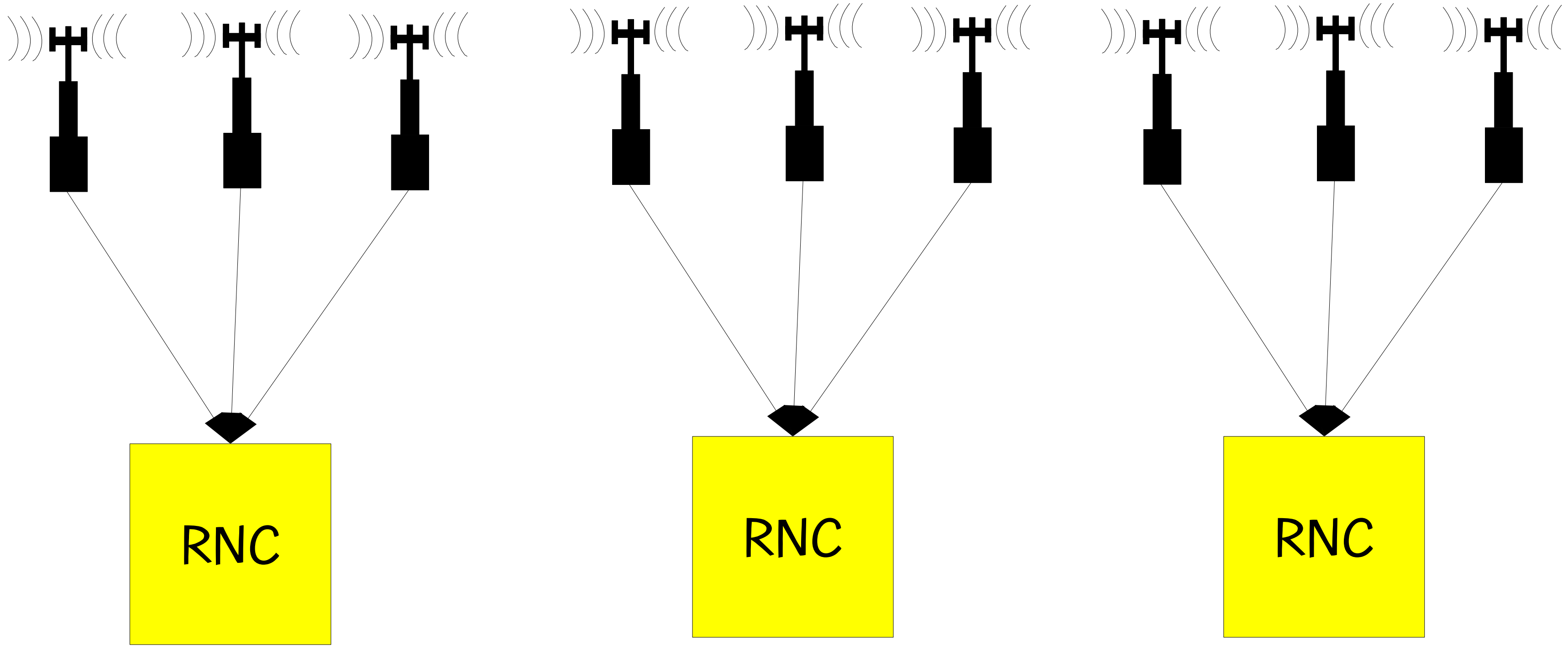
- Handovers between towers connected to different RNCs tend to fail more often than handovers between towers connected to the same RNC.
- Handover failure results in dropped call!



- If we minimize the number of handovers between towers connected to different RNCs we may be able to reduce the number of dropped calls.



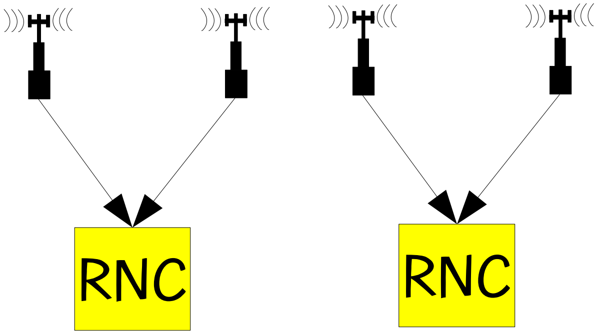
- HANDOVER MINIMIZATION: Assign towers to RNCs such that RNC capacity is not violated and number of handovers between towers assigned to different RNCs is minimized.



- HANDOVER MINIMIZATION: Assign towers to RNCs such that RNC capacity is not violated and number of handovers between towers assigned to different RNCs is minimized.

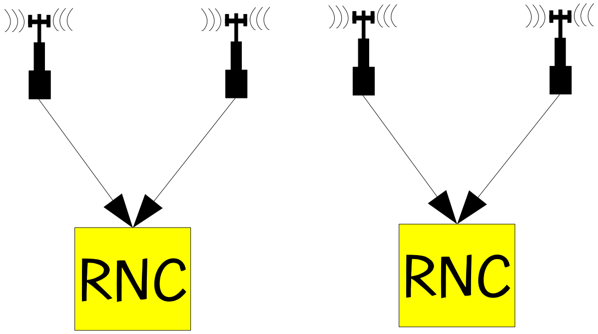
Node-capacitated graph partitioning problem

Example

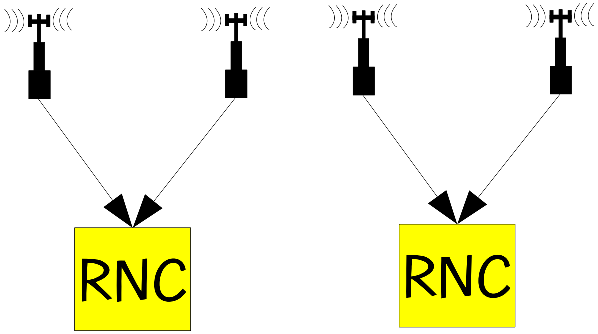


- 4 towers: $t(1) = 25$; $t(2) = 15$; $t(3) = 35$; $t(4) = 25$
- 2 RNCs: $c(1) = 50$; $c(2) = 60$
- Handover matrix:

	1	2	3	4
1	0	100	10	0
2	100	0	200	50
3	10	200	0	500
4	0	50	500	0

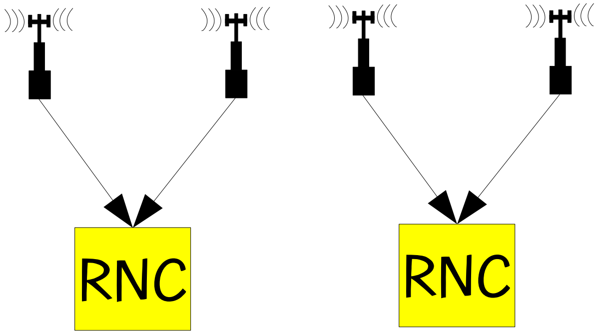


- 4 towers: t(1) = 25; t(2) = 15; t(3) = 35; t(4) = 25
- 2 RNCs: c(1) = 50; c(2) = 60
- Given this traffic profile and RNC capacities the feasible configurations are:
  - RNC(1): { 1, 2 }; RNC(2): { 3, 4 }
  - RNC(1): { 2, 3 }; RNC(2): { 1, 4 }
  - RNC(1): { 2, 4 }; RNC(2): { 1, 3 }
  - RNC(1): { 1, 4 }; RNC(2): { 2, 3 }



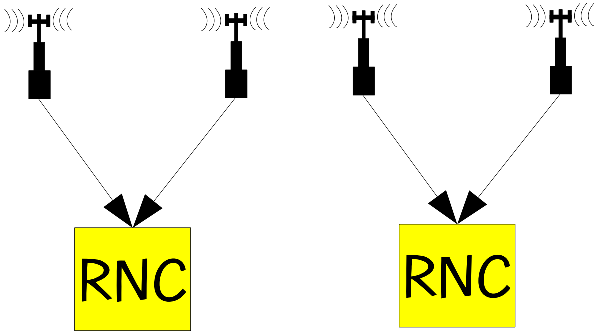
	1	2	3	4
1	0	100	10	0
2	100	0	200	50
3	10	200	0	500
4	0	50	500	0

- Total handover for each configuration:



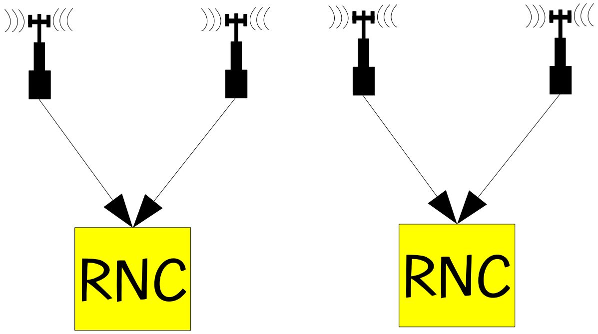
	1	2	3	4
1	0	100	10	0
2	100	0	200	50
3	10	200	0	500
4	0	50	500	0

- Total handover for each configuration:
  - RNC(1): { 1, 2 }; RNC(2): { 3, 4 }: h(1,3) + h(1,4) + h(2,3) + h(2,4) = 10 + 0 + 200 + 50 = 260



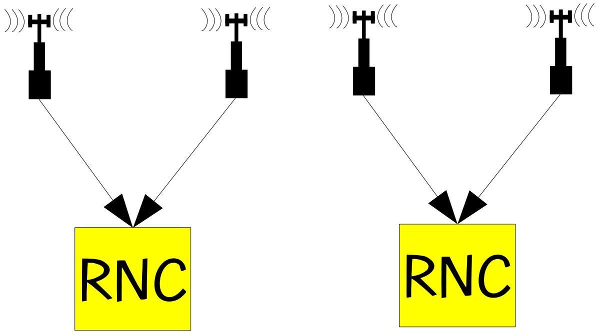
	1	2	3	4
1	0	100	10	0
2	100	0	200	50
3	10	200	0	500
4	0	50	500	0

- Total handover for each configuration:
  - RNC(1): { 1, 2 }; RNC(2): { 3, 4 }: h(1,3) + h(1,4) + h(2,3) + h(2,4) = 10 + 0 + 200 + 50 = 260
  - RNC(1): { 2, 3 }; RNC(2): { 1, 4 }: h(2,1) + h(2,4) + h(3,1) + h(3,4) = 100 + 50 + 10 + 500 = 660



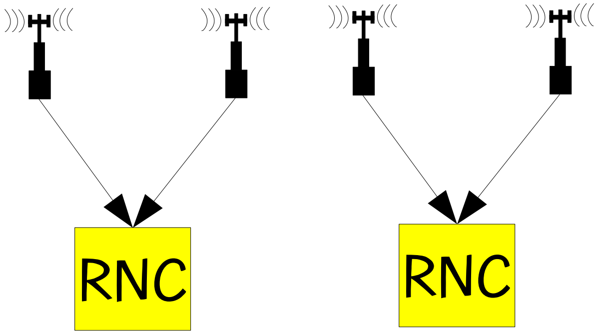
	1	2	3	4
1	0	100	10	0
2	100	0	200	50
3	10	200	0	500
4	0	50	500	0

- Total handover for each configuration:
  - RNC(1): { 1, 2 }; RNC(2): { 3, 4 }: h(1,3) + h(1,4) + h(2,3) + h(2,4) = 10 + 0 + 200 + 50 = 260
  - RNC(1): { 2, 3 }; RNC(2): { 1, 4 }: h(2,1) + h(2,4) + h(3,1) + h(3,4) = 100 + 50 + 10 + 500 = 660
  - RNC(1): { 2, 4 }; RNC(2): { 1, 3 }: h(2,1) + h(2,3) + h(4,1) + h(4,3) = 100 + 200 + 0 + 500 = 800



	1	2	3	4
1	0	100	10	0
2	100	0	200	50
3	10	200	0	500
4	0	50	500	0

- Total handover for each configuration:
  - RNC(1): { 1, 2 }; RNC(2): { 3, 4 }: h(1,3) + h(1,4) + h(2,3) + h(2,4) = 10 + 0 + 200 + 50 = 260
  - RNC(1): { 2, 3 }; RNC(2): { 1, 4 }: h(2,1) + h(2,4) + h(3,1) + h(3,4) = 100 + 50 + 10 + 500 = 660
  - RNC(1): { 2, 4 }; RNC(2): { 1, 3 }: h(2,1) + h(2,3) + h(4,1) + h(4,3) = 100 + 200 + 0 + 500 = 800
  - RNC(1): { 1, 4 }; RNC(2): { 2, 3 }: h(1,2) + h(1,3) + h(4,2) + h(4,3) = 100 + 10 + 50 + 500 = 660

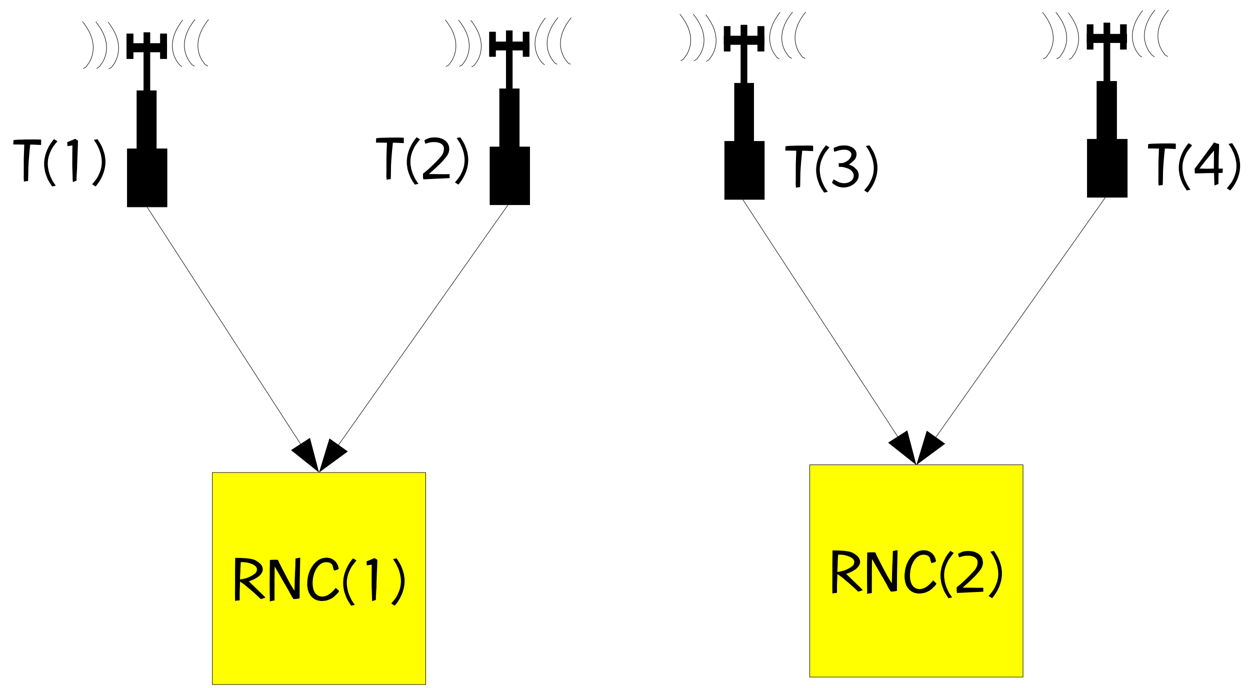


	1	2	3	4
1	0	100	10	0
2	100	0	200	50
3	10	200	0	500
4	0	50	500	0

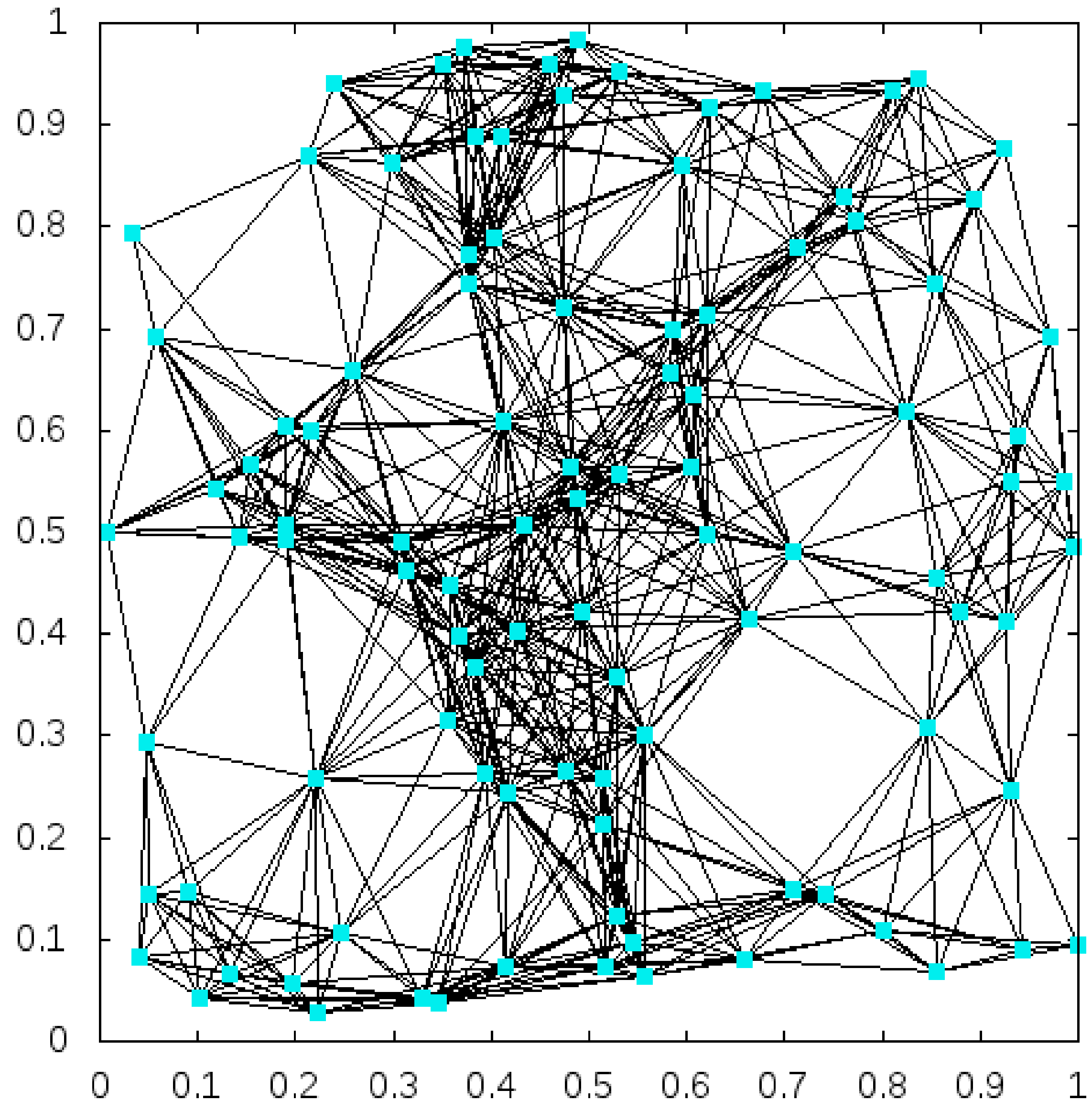
- Total handover for each configuration:
  - RNC(1): { 1, 2 }; RNC(2): { 3, 4 }: h(1,3) + h(1,4) + h(2,3) + h(2,4) = 10 + 0 + 200 + 50 = **260**
  - RNC(1): { 2, 3 }; RNC(2): { 1, 4 }: h(2,1) + h(2,4) + h(3,1) + h(3,4) = 100 + 50 + 10 + 500 = 660
  - RNC(1): { 2, 4 }; RNC(2): { 1, 3 }: h(2,1) + h(2,3) + h(4,1) + h(4,3) = 100 + 200 + 0 + 500 = 800
  - RNC(1): { 1, 4 }; RNC(2): { 2, 3 }: h(1,2) + h(1,3) + h(4,2) + h(4,3) = 100 + 10 + 50 + 500 = 660

	1	2	3	4
1	0	100	10	0
2	100	0	200	50
3	10	200	0	500
4	0	50	500	0

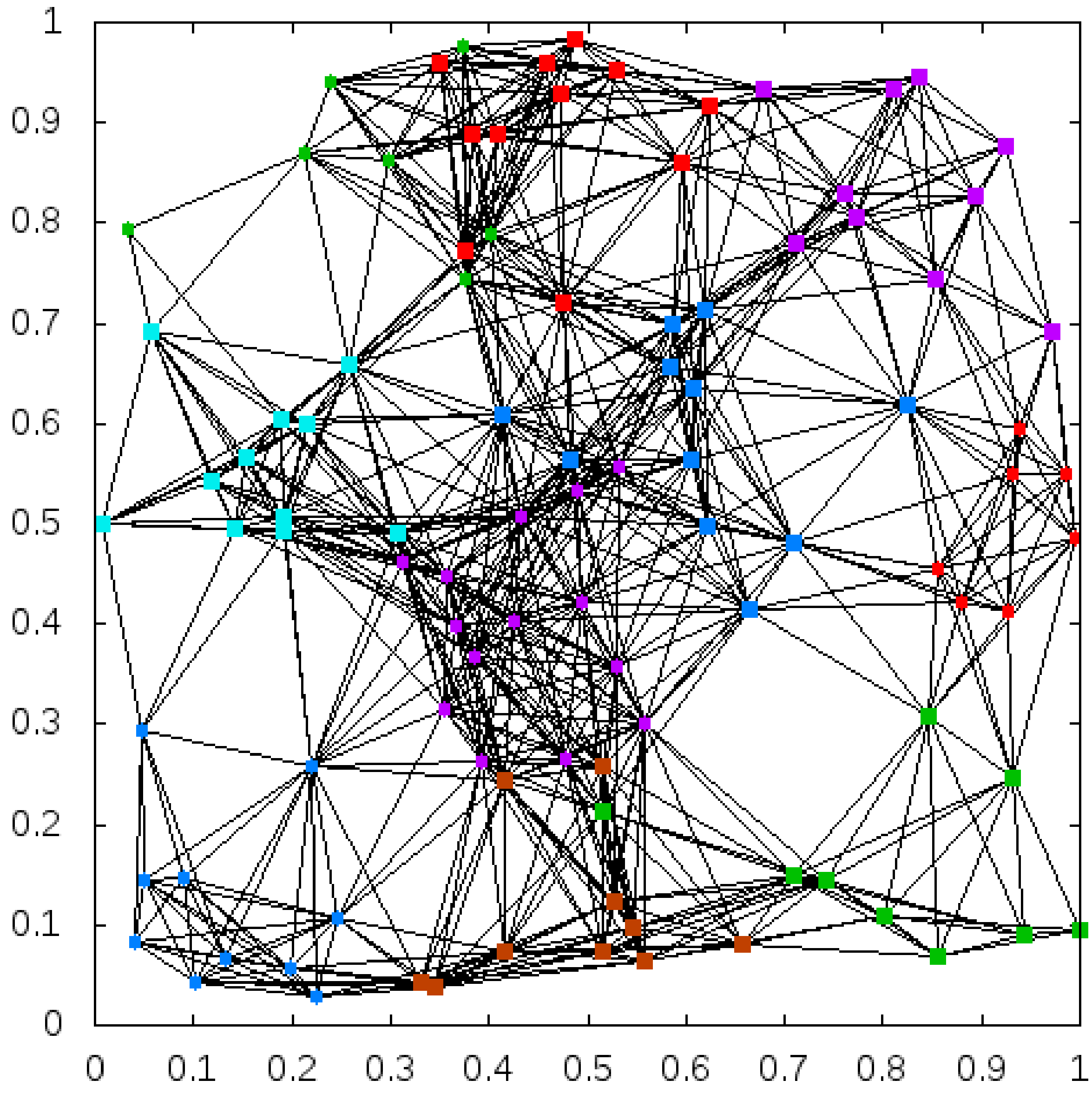
Optimal configuration:



G=(T,E)   Nodeset T are the towers; Edgeset: (i,j)∈E iff h(i,j)+h(j,i) > 0



Tower are assigned to RNCs indicated by distinct colors/shapes



Mixed integer programming formulation

- T is the set of towers
- R is the set of RNCs
- $x_{e,k} = 1$ if edge $e = (i,j)$ has both endpoints in RNC k
- $y_{i,k} = 1$ if tower i is assigned to RNC k

Mixed integer programming formulation

Each tower can only be assigned to one RNC:

$$\text{sum}_{\{k \in R\}}\, y_{i,k} = 1, \text{ for all } i \in T$$

Mixed integer programming formulation

Each $e=(i,j)$ cannot be in RNC k if either of its endpoints is not assigned to RNC k:

$$x_{e,k} \leq y_{i,k}\text{, for all } e=(i,j) \in E, k \in R$$

$$x_{e,k} \leq y_{j,k}\text{, for all } e=(i,j) \in E, k \in R$$

$$x_{e,k} \geq y_{i,k} + y_{j,k} - 1\text{, for all } e=(i,j) \in E, k \in R$$

Mixed integer programming formulation

Each RNC k can only accommodate c_k units of traffic:

$$\text{sum}_{\{i \in T\}} t_i y_{i,k} \leq c_k, \text{ for all } k \in R$$

Mixed integer programming formulation

Minimize handover between towers assigned to different RNCs is equivalent to maximize handover between towers assigned to the same RNC.

Objective function:

$$\max \{ \text{sum}_{\{k \in R\}} \{ \text{sum}_{\{e=(i,j) \in E\}} h(i,j)\ x_{e,k} \} \}$$

at&t Your world. Delivered.

CPLEX MIP solver

Towers	RNCs	BKS	CPLEX	time (s)
20	10	7602	7602	18.8
30	15	18266	18266	25911.0
40	15	29700	29700	101259.9
100	15	19000	49270	1 day
100	25	36412	58637	1 day
100	50	60922	70740	1 day

CPLEX MIP solver

Towers	RNCs	BKS	CPLEX	time (s)
20	10	7602	7602	18.8
30	15	18266	18266	25911.0
40	15	29700	29700	101259.9
100	15	19000	49270	1 day
100	25	36412	58637	1 day
100	50	60922	70740	1 day

We would like to solve instances with 1000 towers.

CPLEX MIP solver

Towers	RNCs	BKS	CPLEX	time (s)
20	10	7602	7602	18.8
30	15	18266	18266	25911.0
40	15	29700	29700	101259.9
100	15	19000	49270	1 day
100	25	36412	58637	1 day
100	50	60922	70740	1 day

We would like to solve instances with 1000 towers.

Need heuristics!

Generalized quadratic assignment problem

Generalized quadratic assignment

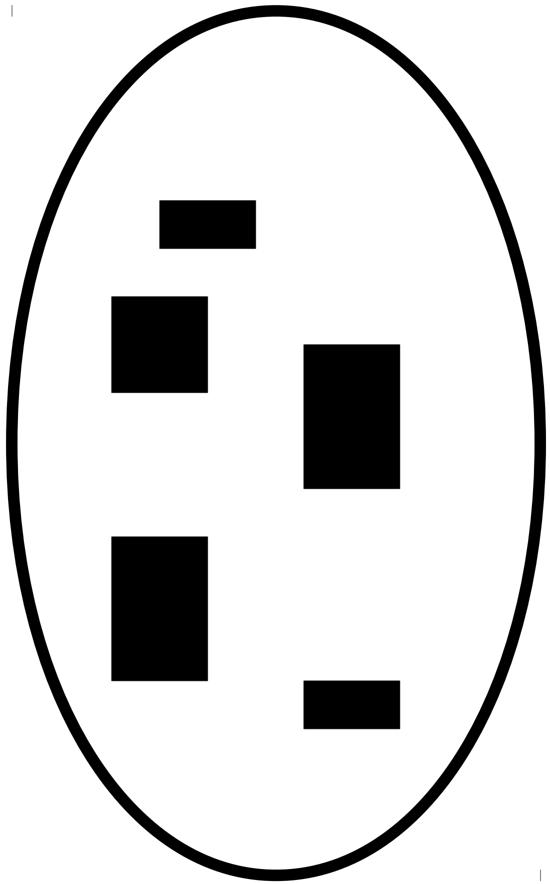
Quadratic assignment problem (QAP): Assign n facilities to n locations minimizing sum of products of flow between facilities and distance between locations over all assignments.

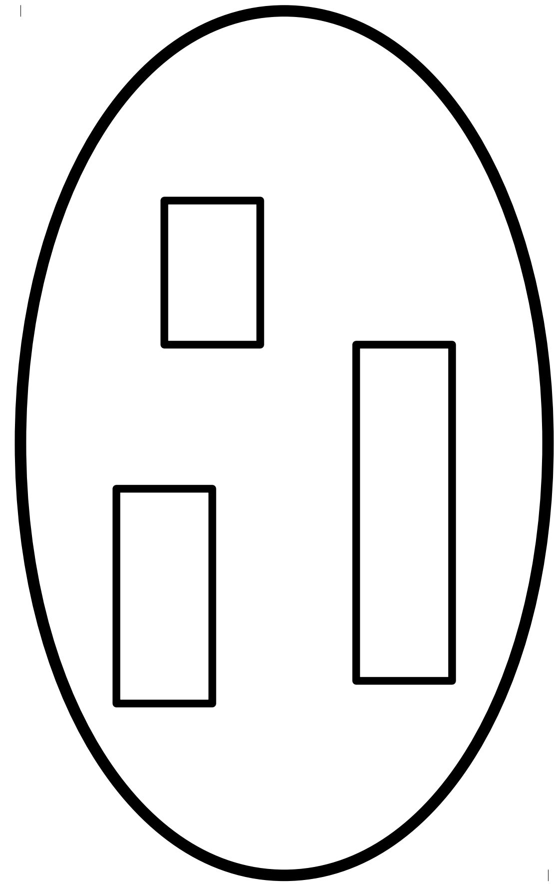
GQAP is a generalization of the QAP.

Multiple facilities can be assigned to a single location as long as the capacity of the location allows.

N: set of n facilities

M: set of m locations

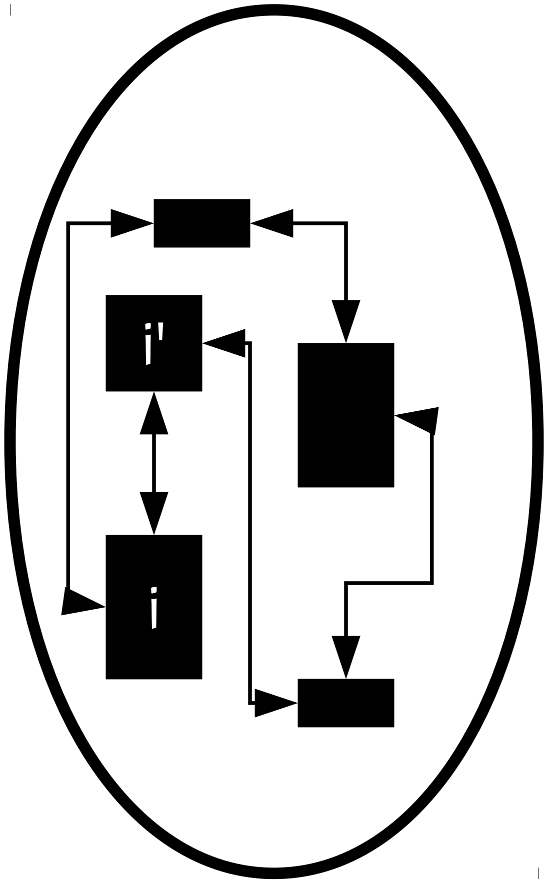




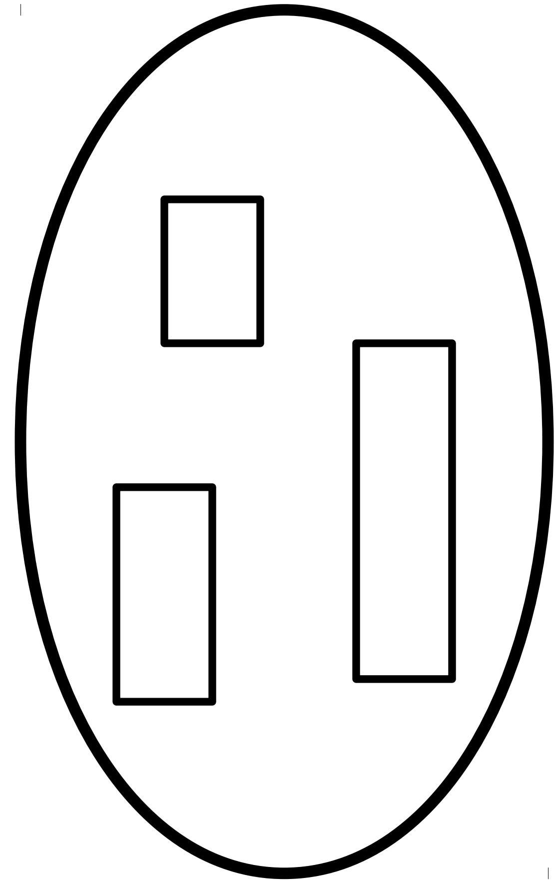
d_i : capacity demanded by facility $i \in N$

Q_j : capacity of location $j \in M$

N: set of n facilities

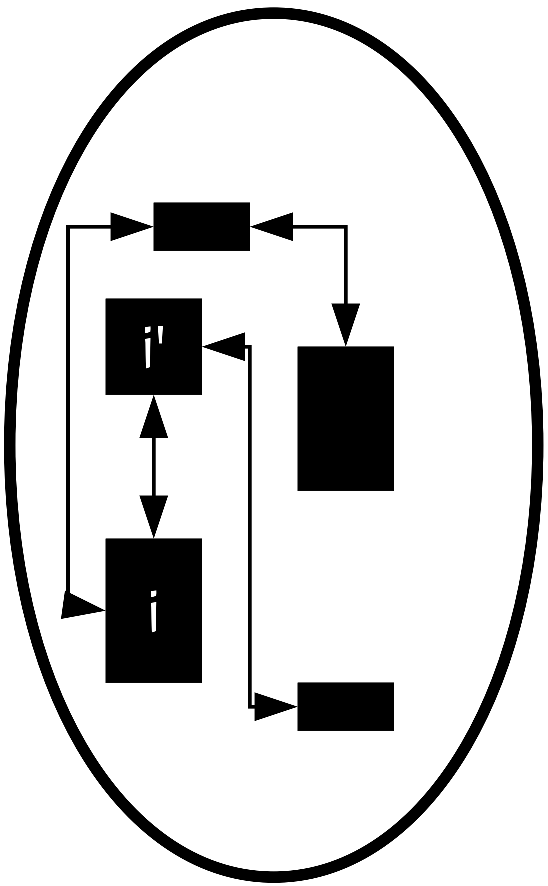


M: set of m locations

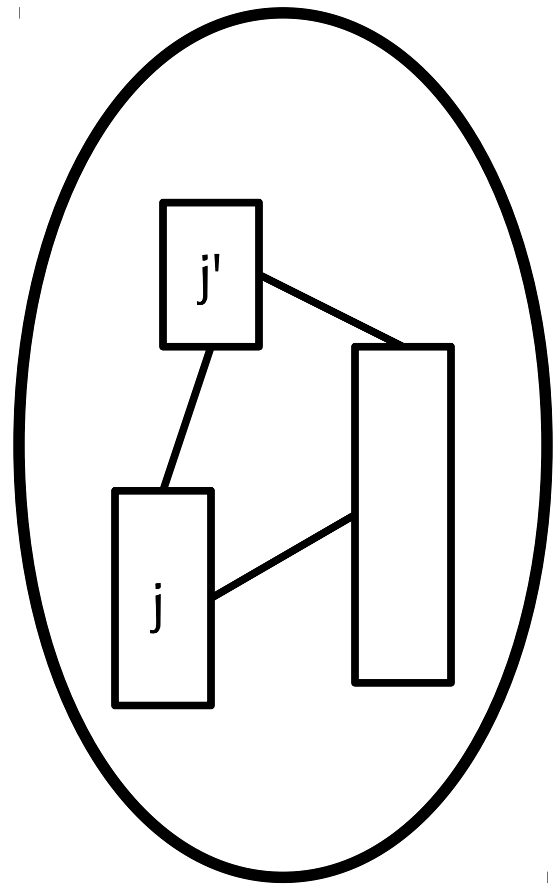


$A_{nxn}=(a_{ii'})$: flow between facilities

N: set of n facilities



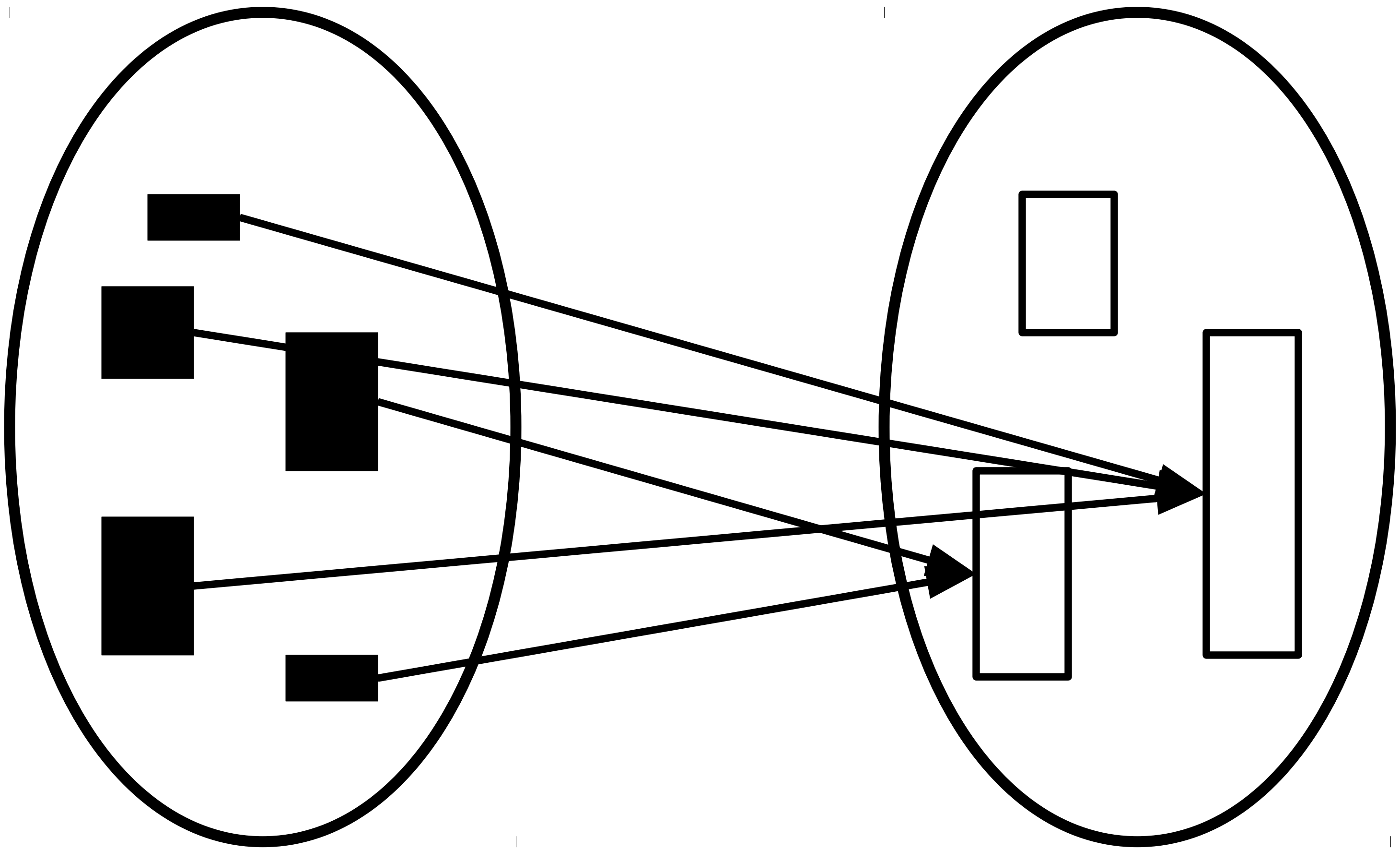
M: set of m locations



$A_{nxn} = (a_{ii'})$: flow between facilities $B_{mxm} = (b_{jj'})$: distance between locations

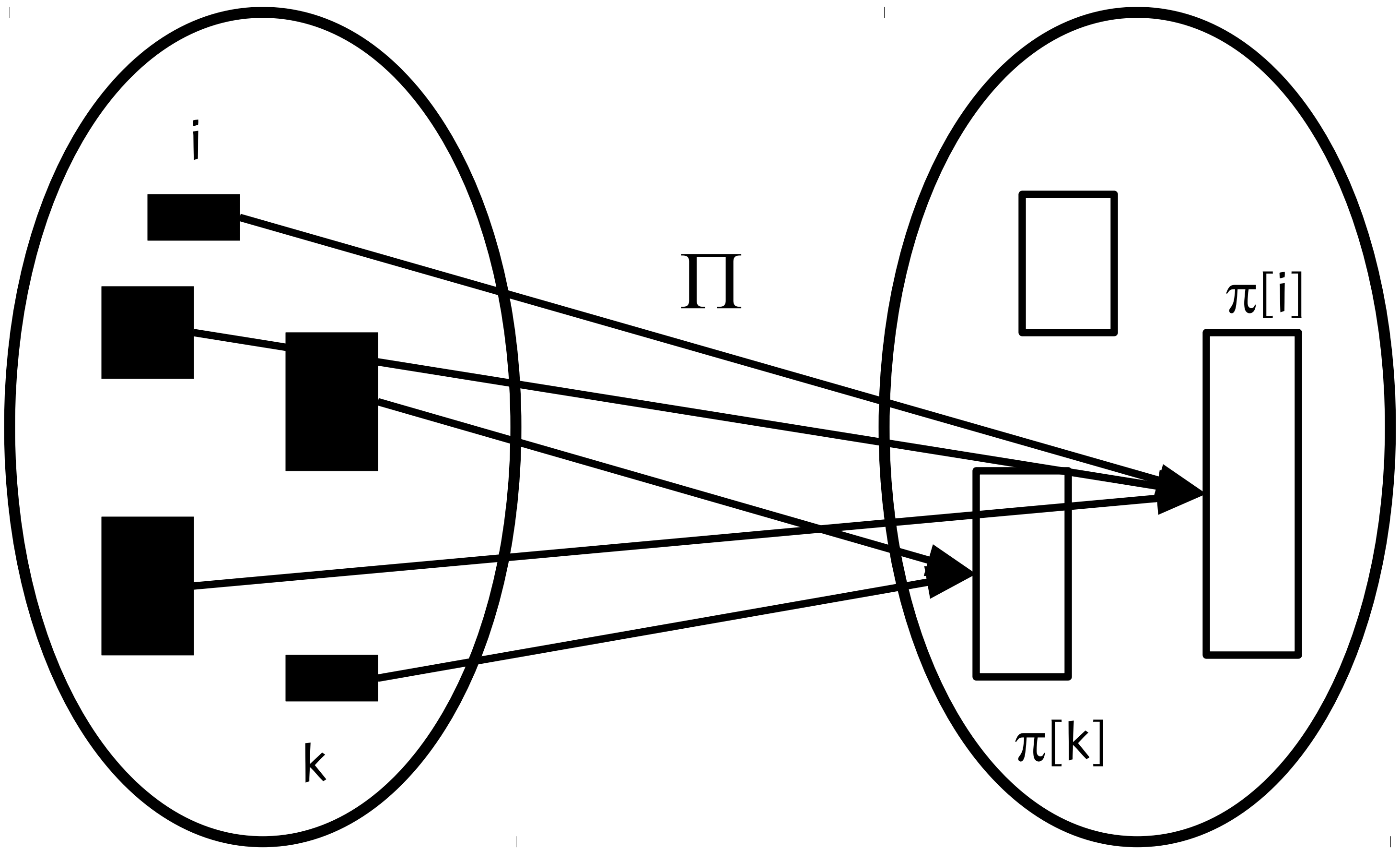
at&t
Your world. Delivered.

The generalized quadratic assignment problem



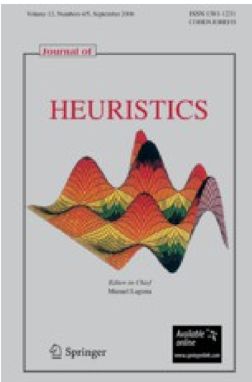
GQAP seeks a assignment, without violating the capacities of locations, that minimizes the sum of products of flows and distances.

The generalized quadratic assignment problem



cost[Π] = sum(i=1,n) sum (i≠k=1,n) F[i,k]*D[π[i],π[k]]

Paper and java code



G.R. Mateus, R.M.A. Silva, and M.G.C. Resende, GRASP with path-relinking for the generalized quadratic assignment problem, J. of Heuristics 17 (527-565) 2011

http://www.research.att.com/~mgcr/doc/gpr-gqap.pdf

We developed a Java implementation of the algorithm.

Handover minimization is a special case of the GQAP

- Towers $\leftrightarrow$ Facilities
  - tower traffic is facility demand
- RNCs $\leftrightarrow$ Locations
  - RNC capacity is Location capacity
- Handovers between towers $\leftrightarrow$ Flows between facilities
- Distance between each pair of RNC = 1

GRASP with evolutionary path-relinking for handover minimization

GRASP with evolutionary path-relinking

- Algorithm maintains an elite set of diverse good-quality solutions found during search
- Repeat
  - build tower-to-RNC assignment π' using a randomized greedy algorithm
  - apply local search to find local min assignment π near π'
  - select assignment π' from elite pool and apply path-relinking operator between π' and π and attempt to add result to elite set
- Apply evolutionary path-relinking to elite set once in while during search

Randomized greedy construction

- Open one RNC at a time …
  – use heuristic A to assign first tower to RNC
  – while RNC can accommodate an unassigned tower
    - use heuristic B to assign next tower to RNC
- If all available RNCs have been opened and some tower is still unassigned, open one or more artificial RNCs having capacity equal to the max capacity over all real RNCs

Randomized greedy construction:
Heuristic A to assign first tower to RNC

- Let $H(i) = \text{sum}_{(j = 1,\ldots,T)}\ h(i,j) + h(j,i)$
- Let Ω be the set of unassigned towers that fit in RNC
- Choose tower i from Ω with probability proportional to its H(i) value and assign i to RNC

Randomized greedy construction:
Heuristic B to assign remaining towers to RNC

- Let $g(i) = \text{sum}_{(j \in RNC)} h(i,j) + h(j,i)$
- Let Ω be the set of unassigned towers that fit in RNC
- Select tower i from Ω with probability proportional to its $g(i)$ value and assign i to RNC

Local search

- Repeat until no improving reassignment of tower to RNC exists:
  - Let { i, j, k } be such that tower i is assigned to RNC j, RNC k has available capacity to accommodate tower i and moving i from RNC j to RNC k reduces the number of handovers between towers assigned to different RNCs
  - If { i, j, k } exists, then move tower i from RNC j to RNC k

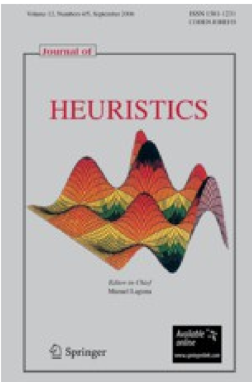
Path-relinking

Intensification strategy exploring trajectories connecting elite solutions (Glover, 1996)

Originally proposed in the context of tabu search and scatter search.

Paths in the solution space leading to other elite solutions are explored in the search for better solutions.

Recent survey paper on path-relinking

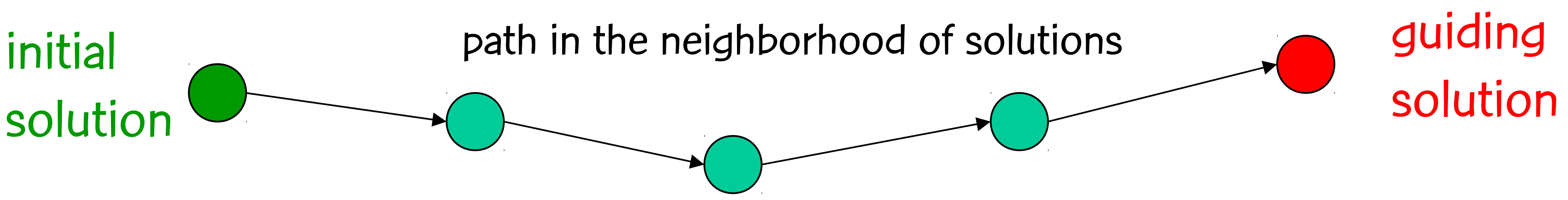


C.C. Ribeiro and M.G.C. Resende, Path-relinking intensification methods for stochastic local search algorithms, J. of Heuristics, vol. 18, pp. 193-214, 2012.

http://www.research.att.com/~mgcr/doc/spr.pdf

Path-relinking

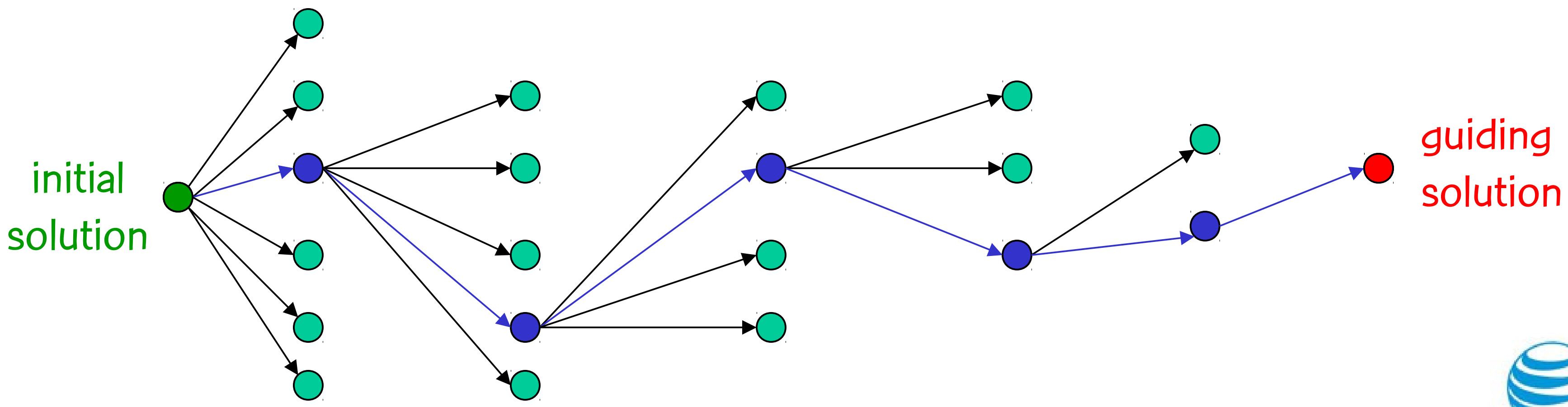
Exploration of trajectories that connect high quality (elite) solutions:



Path-relinking

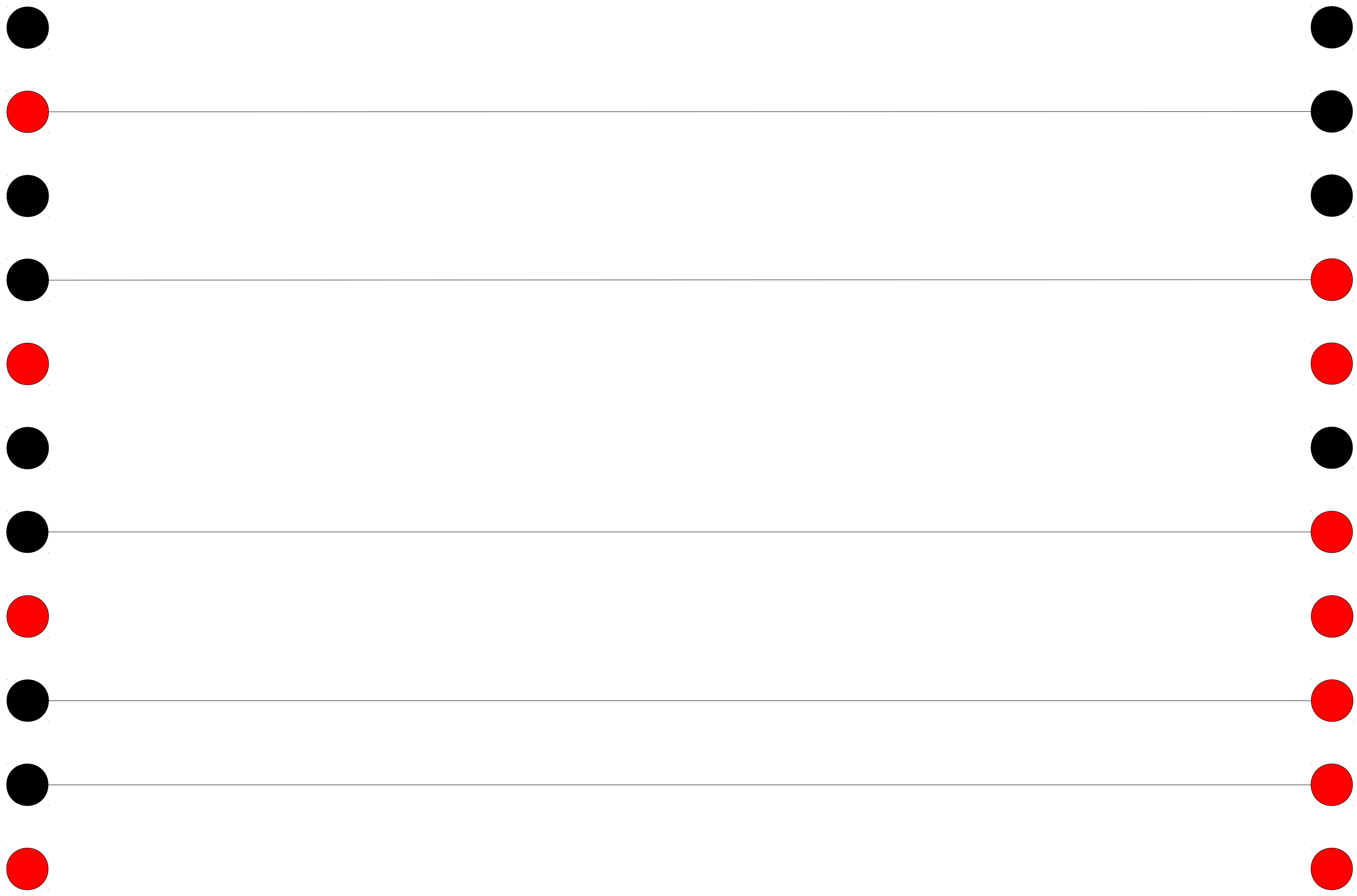
Path is generated by selecting moves that introduce in the initial solution attributes of the guiding solution.

At each step, all moves that incorporate attributes of the guiding solution are evaluated and the best move is selected:



starting solution | # PR example | guiding solution

Example with two RNCs



PR example

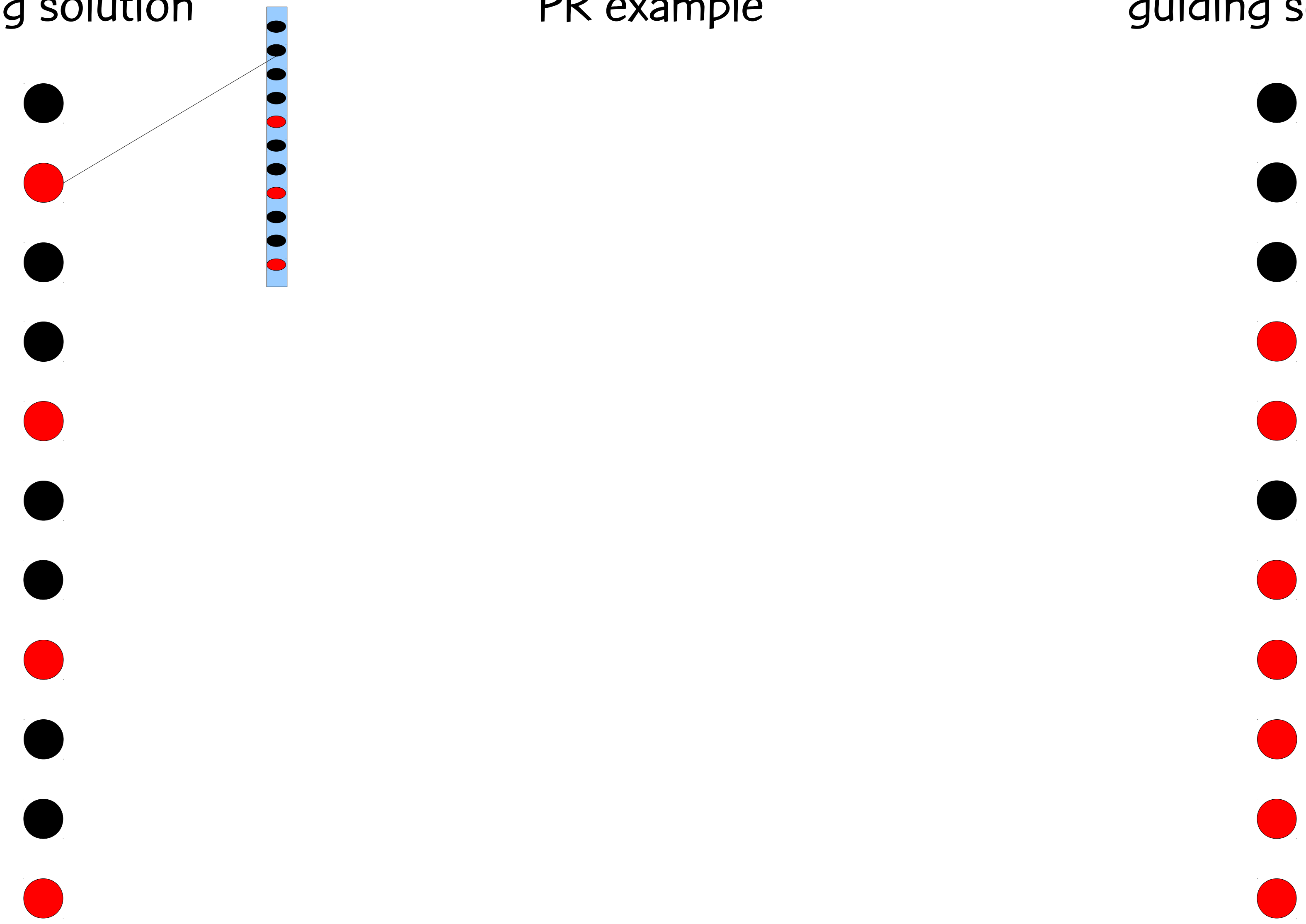
$|\Delta(x,y)| = 5$

Example with two RNCs

starting solution

PR example

guiding solution

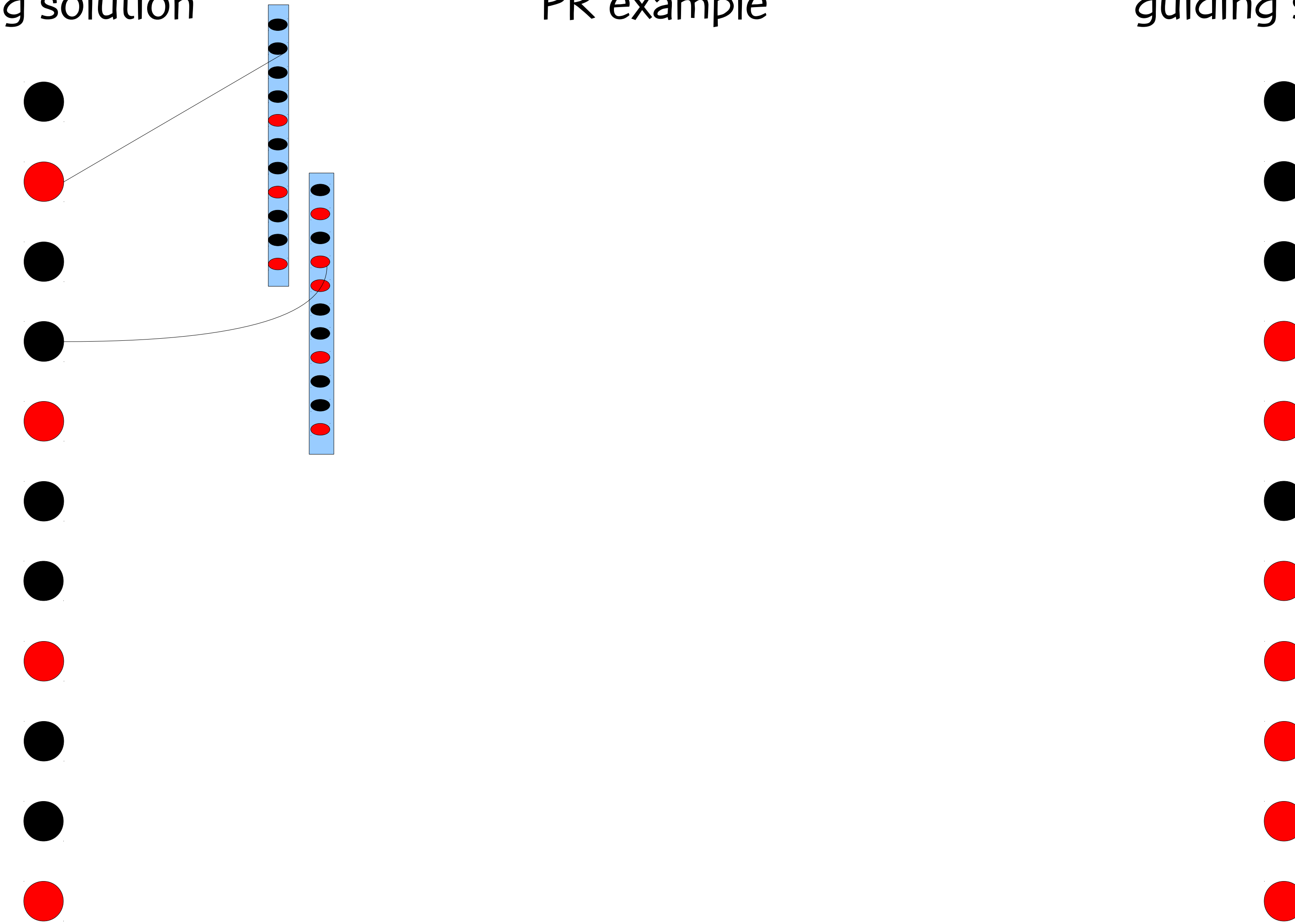


Example with two RNCs

PR example

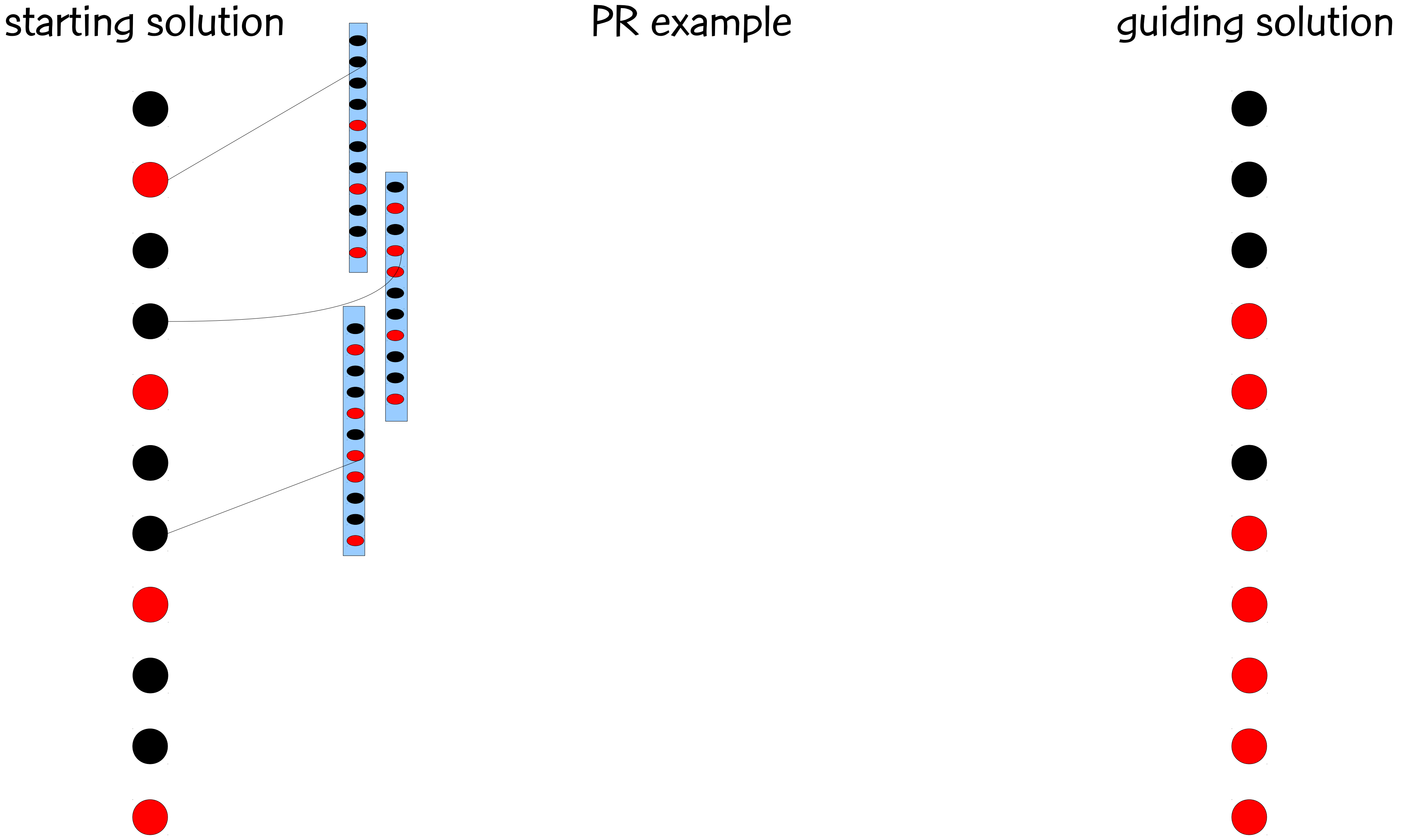
starting solution

guiding solution



Example with two RNCs

PR example

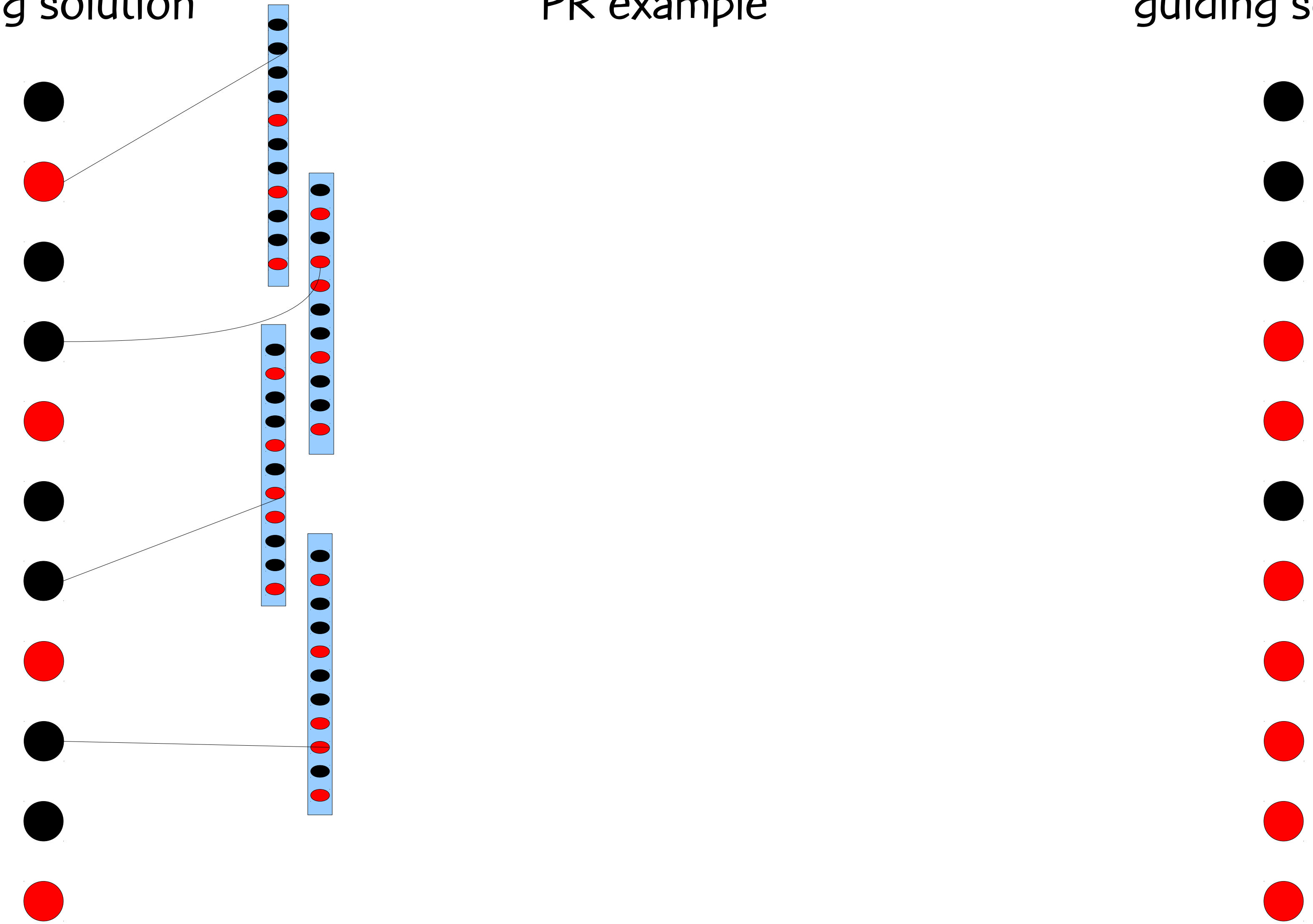


Example with two RNCs

PR example

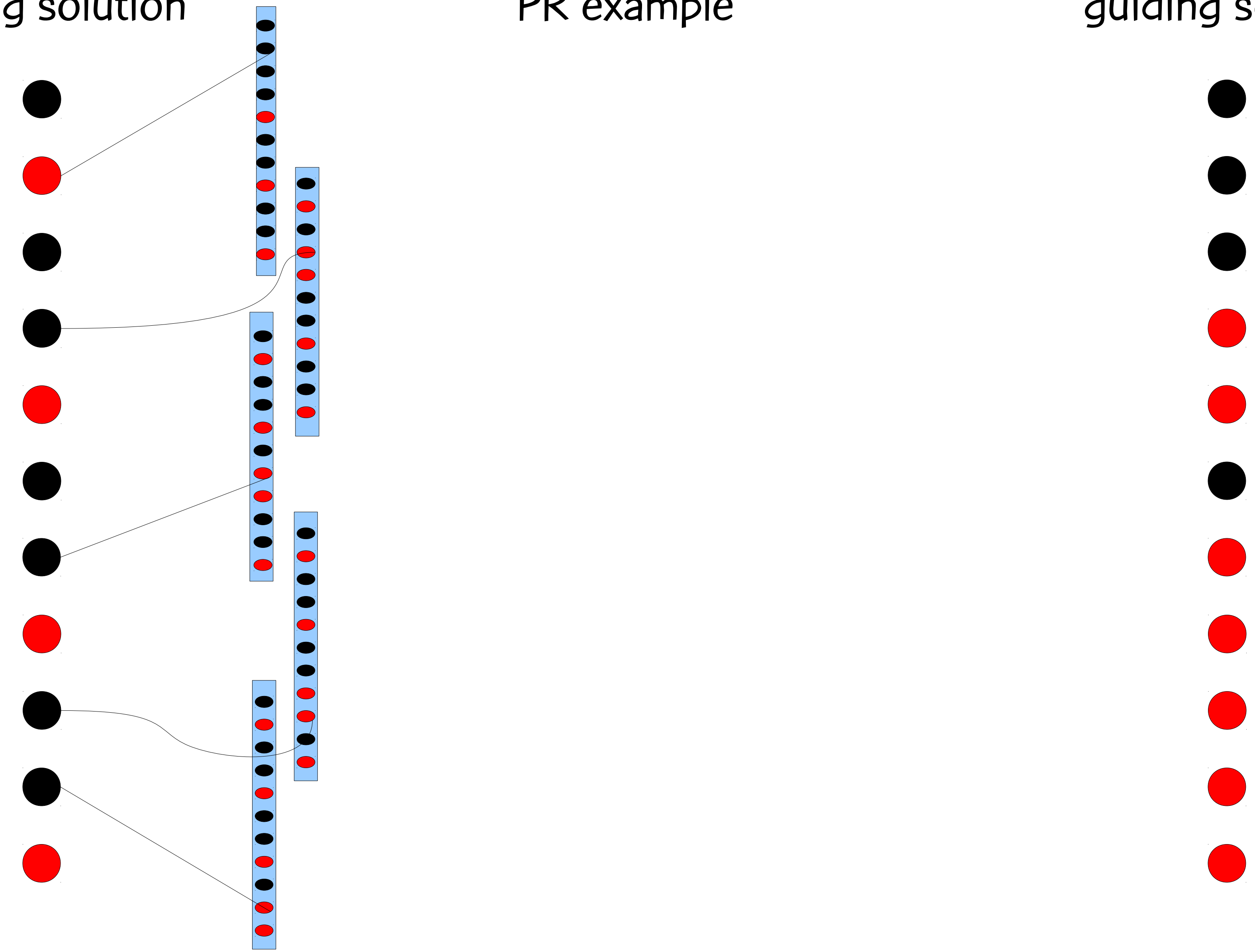
starting solution

guiding solution



Example with two RNCs

PR example

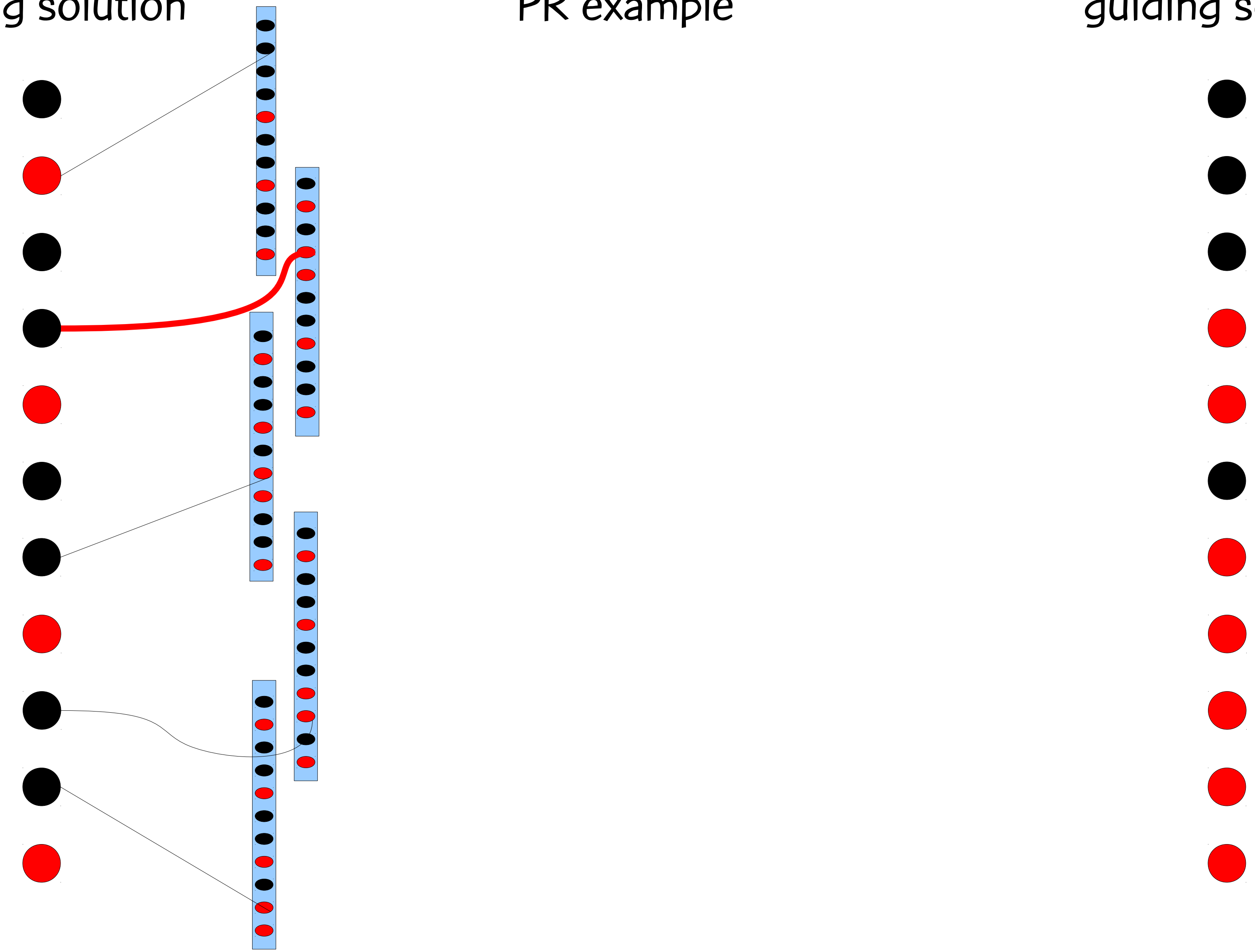


Example with two RNCs

starting solution

PR example

guiding solution

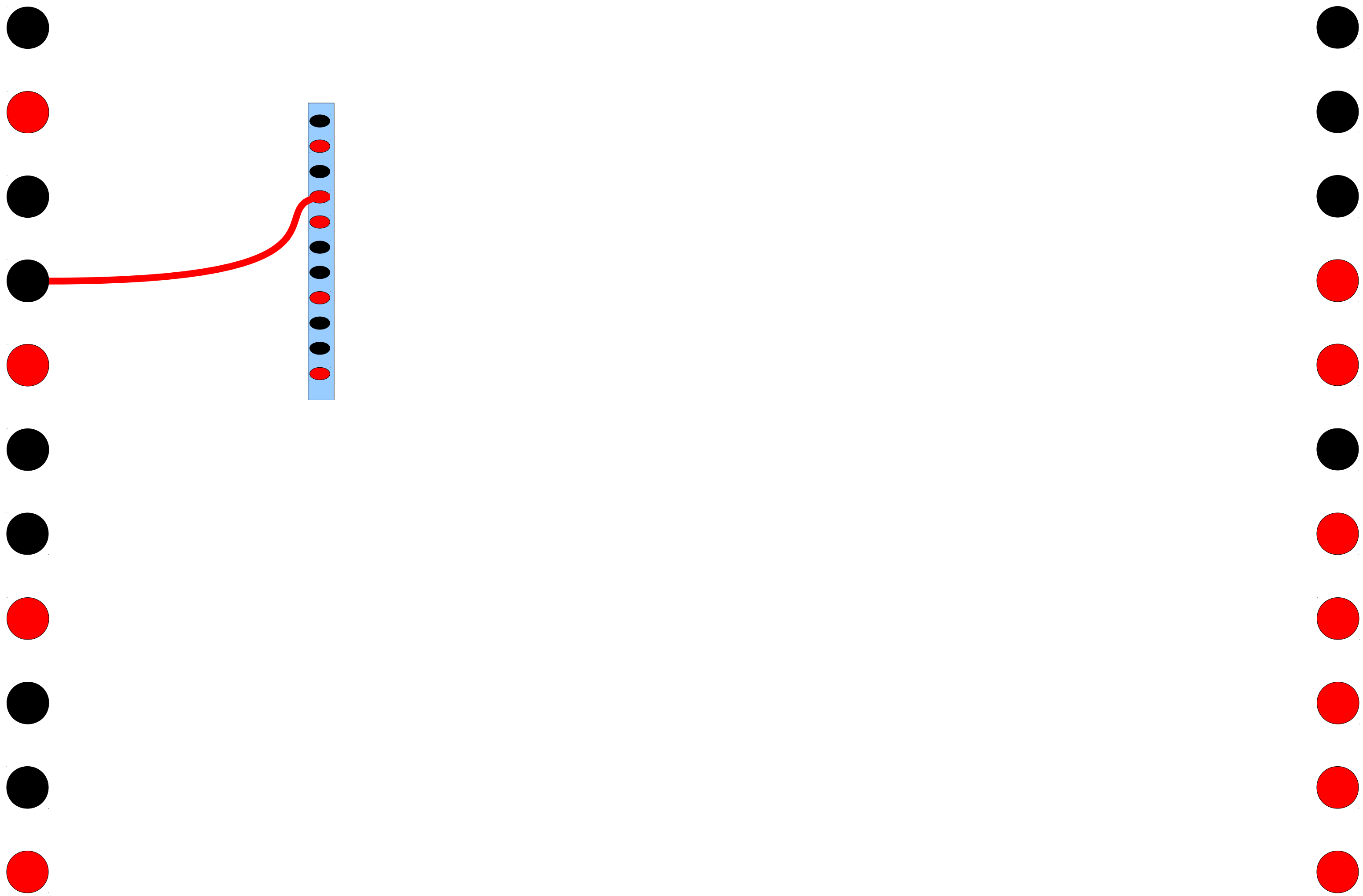


Example with two RNCs

starting solution

PR example

guiding solution

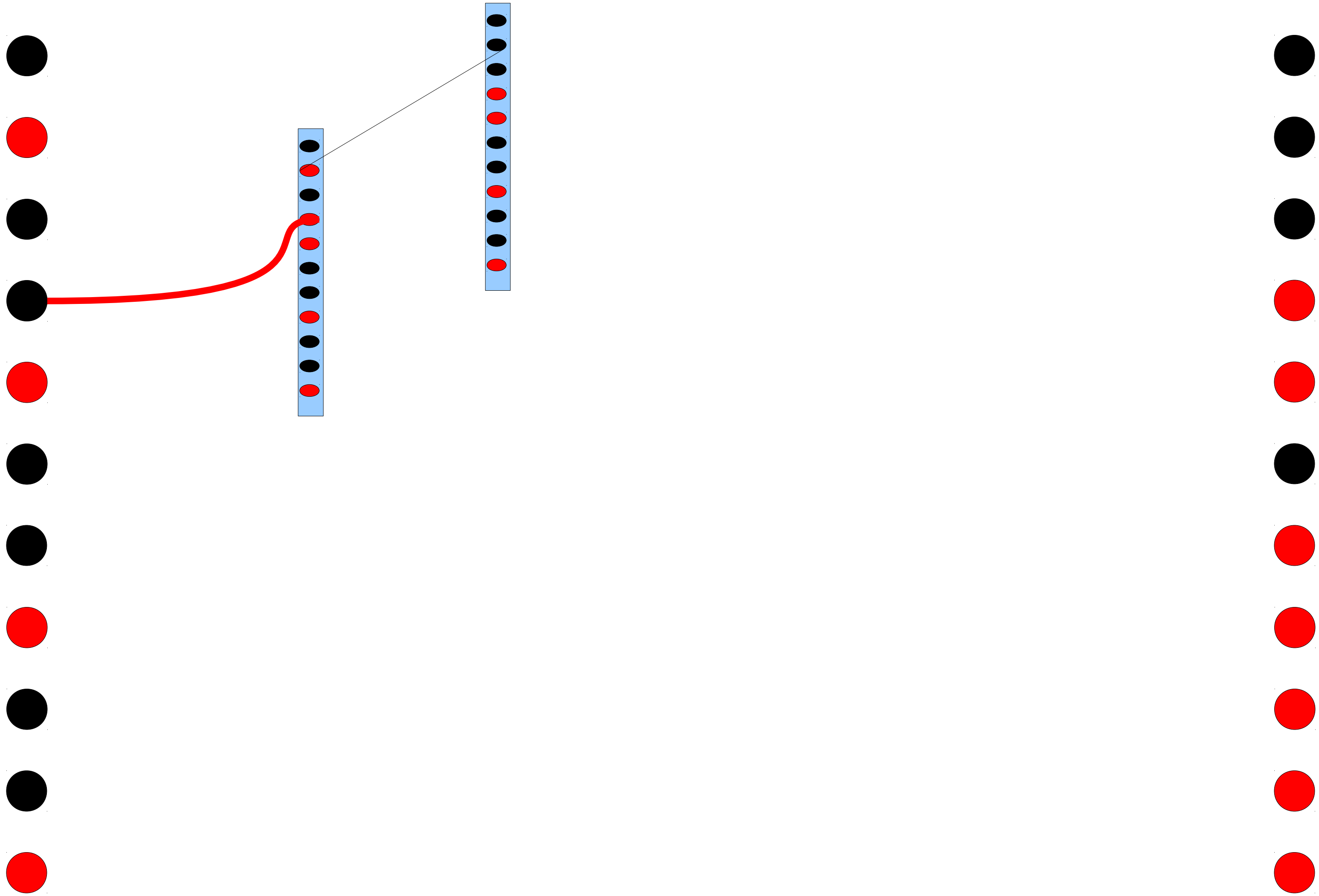


Example with two RNCs

starting solution

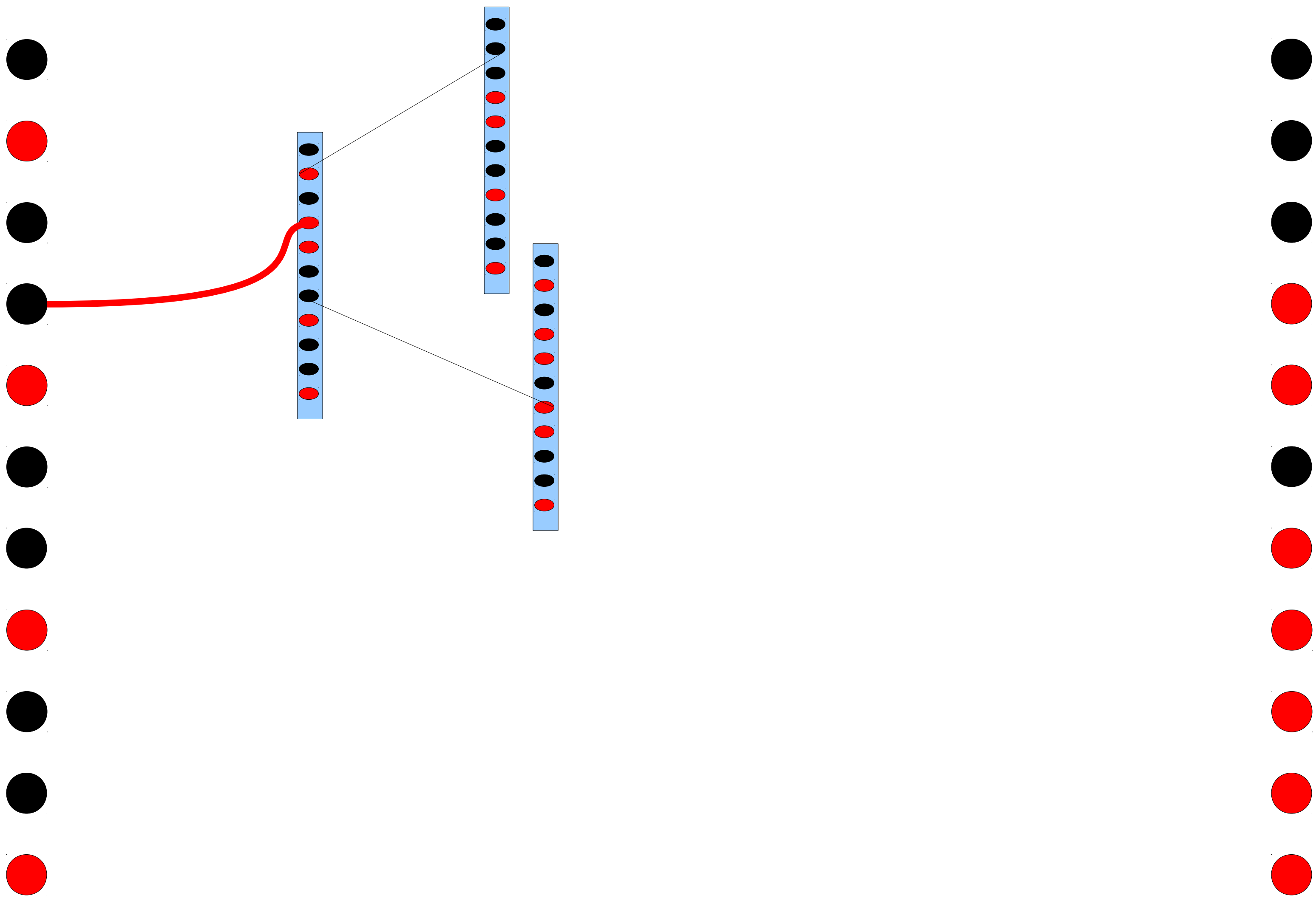
PR example

guiding solution



Example with two RNCs

starting solution PR example guiding solution

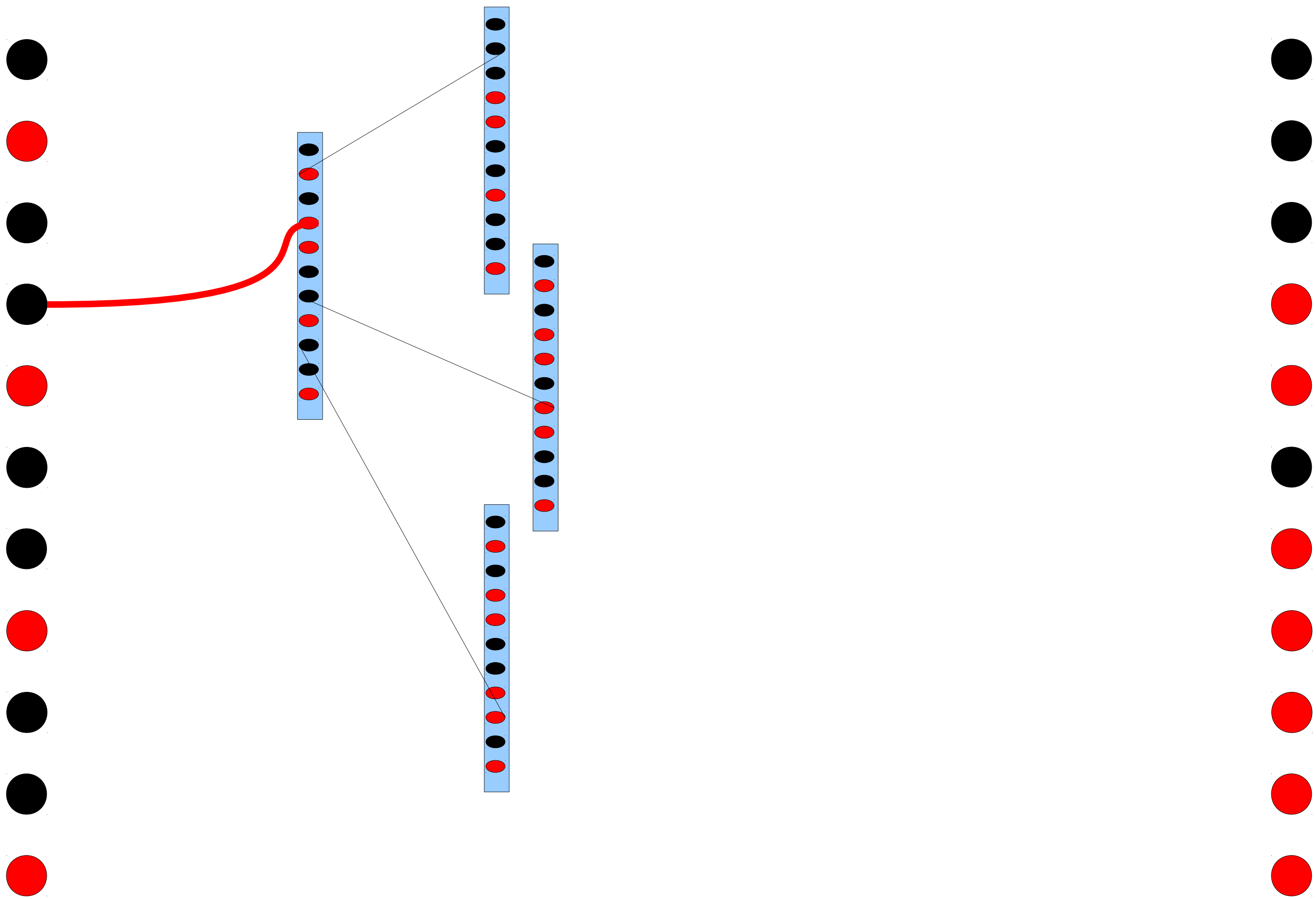


Example with two RNCs

PR example

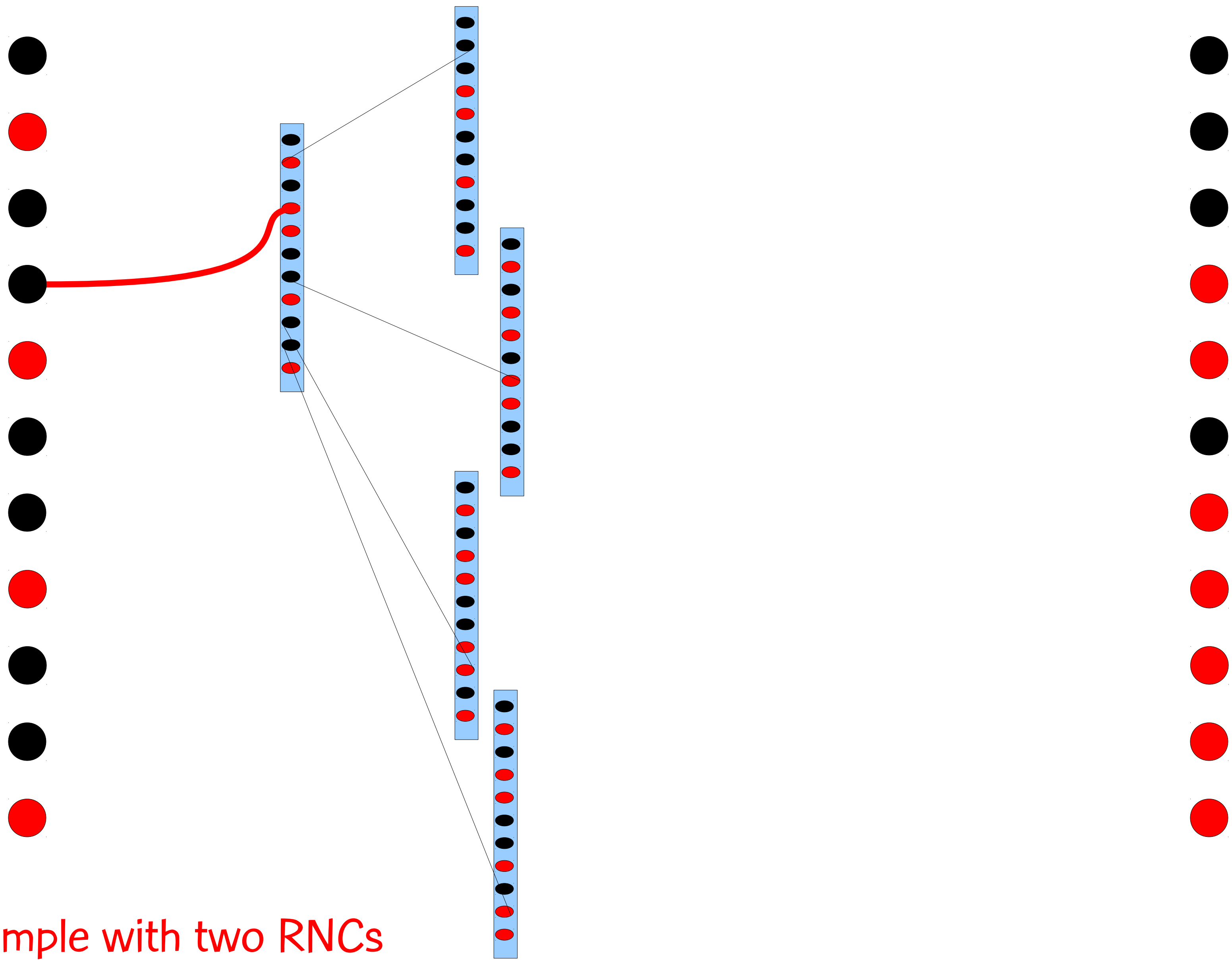
starting solution

guiding solution



Example with two RNCs

starting solution | PR example | guiding solution

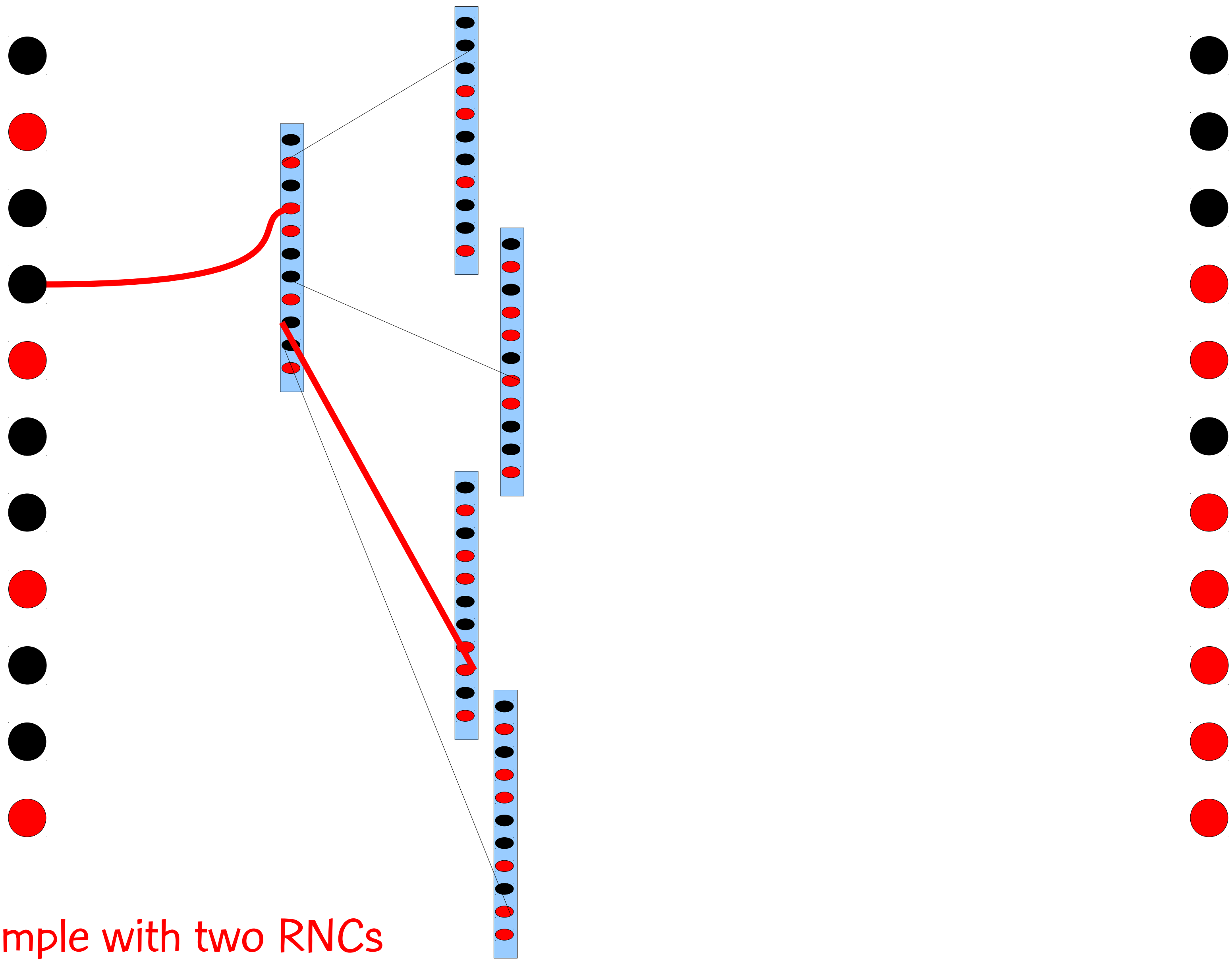


Example with two RNCs

starting solution

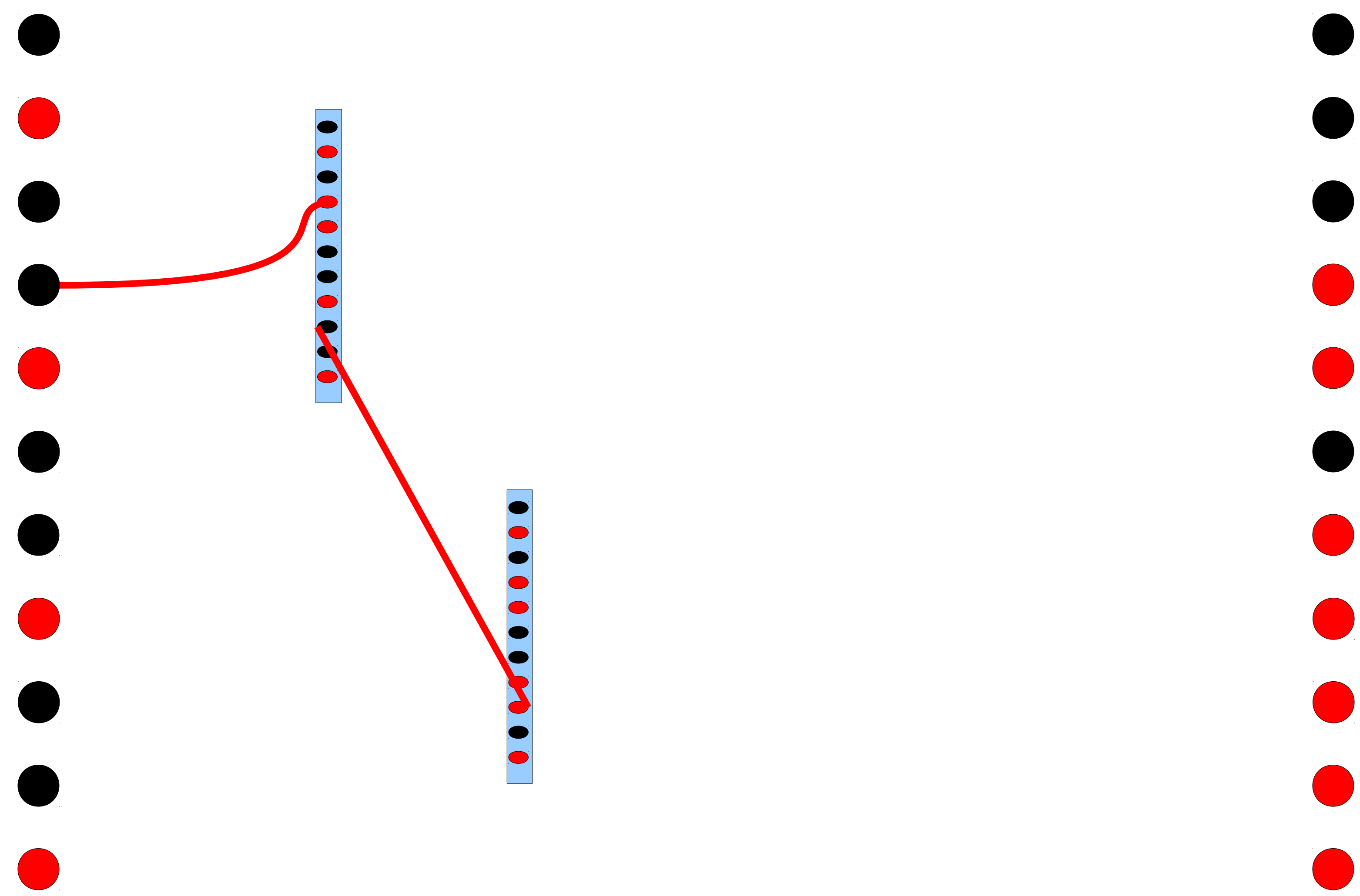
PR example

guiding solution

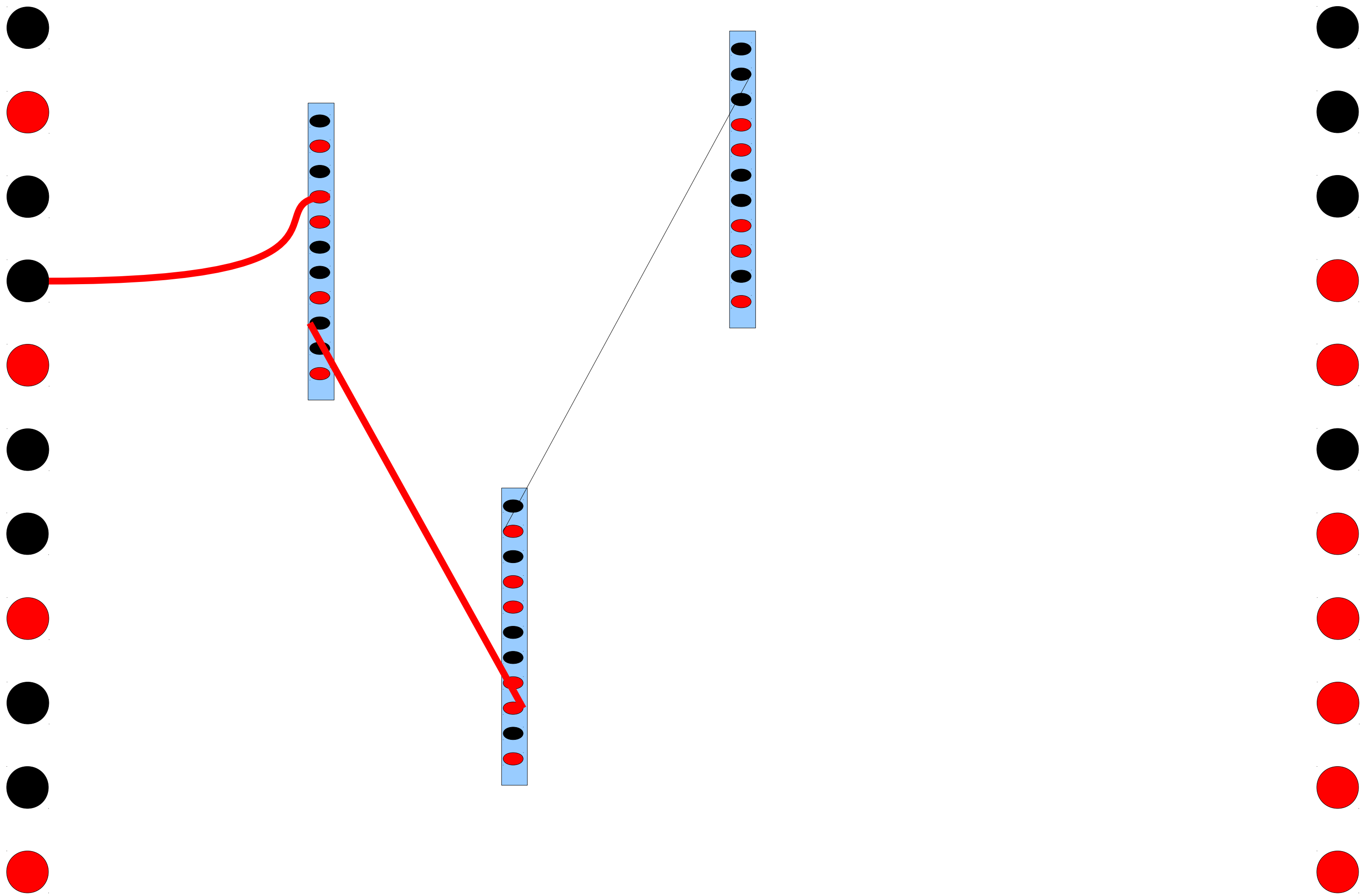


Example with two RNCs

starting solution PR example guiding solution



Example with two RNCs

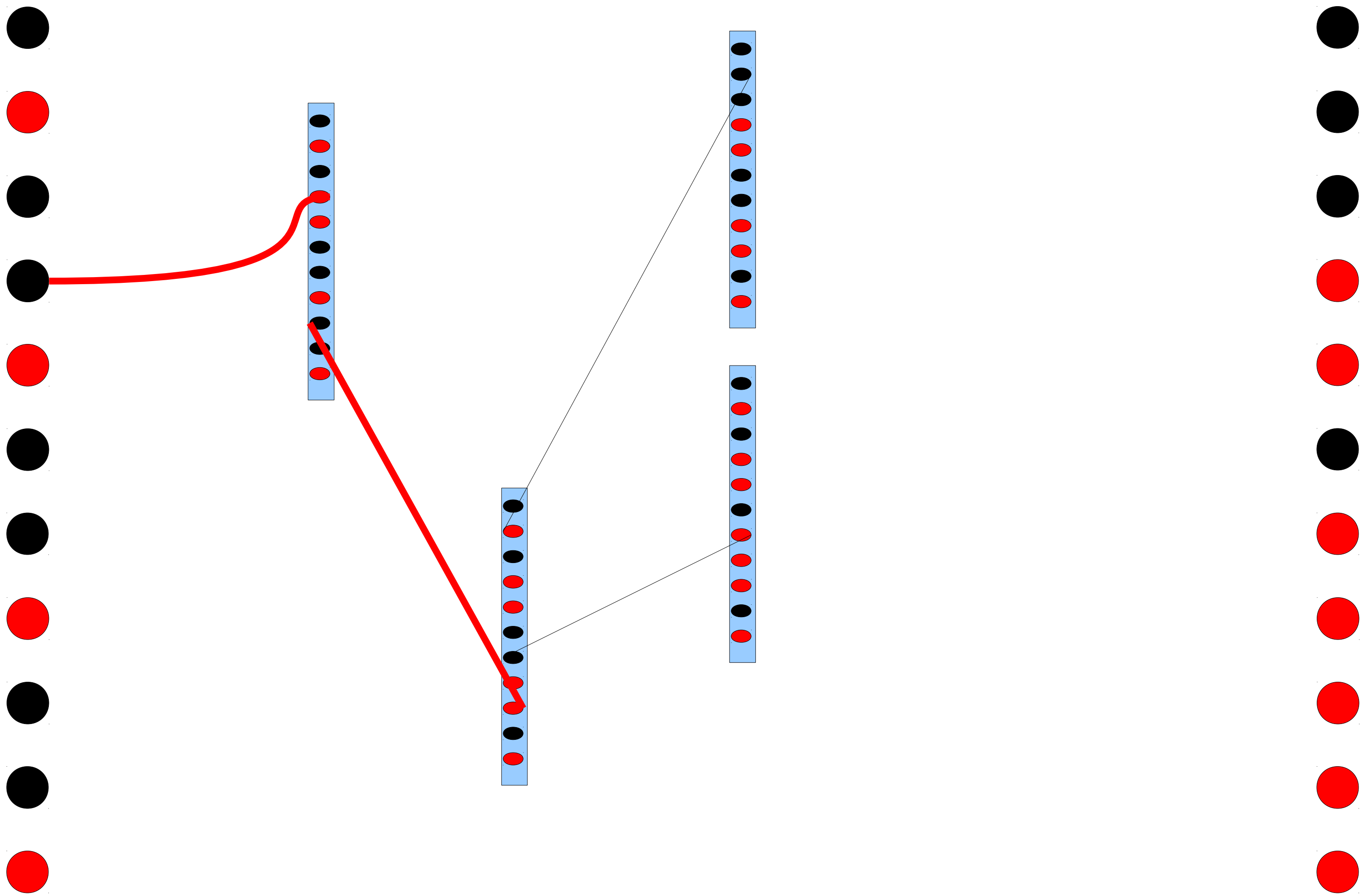


Example with two RNCs

starting solution

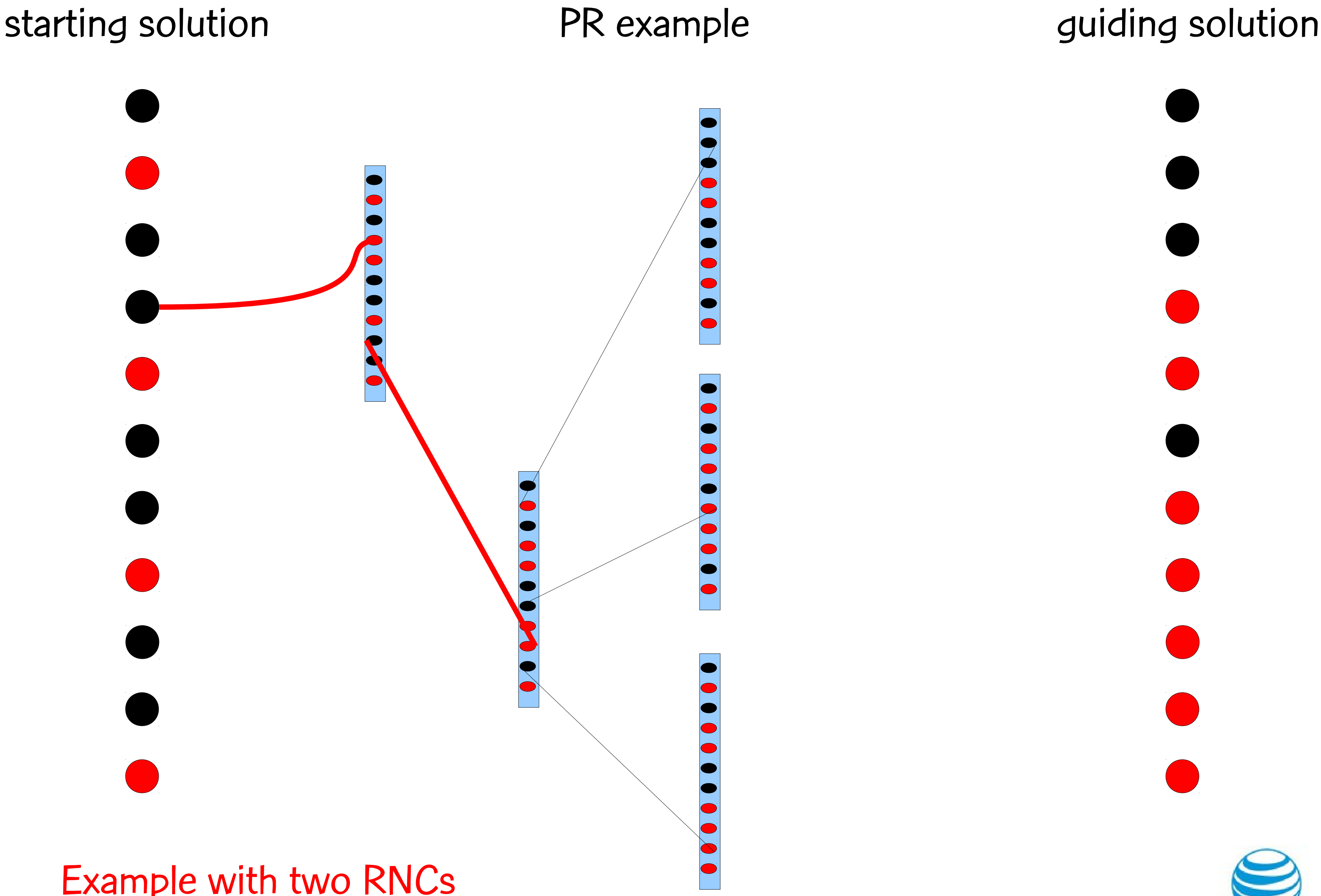
PR example

guiding solution



Example with two RNCs

PR example

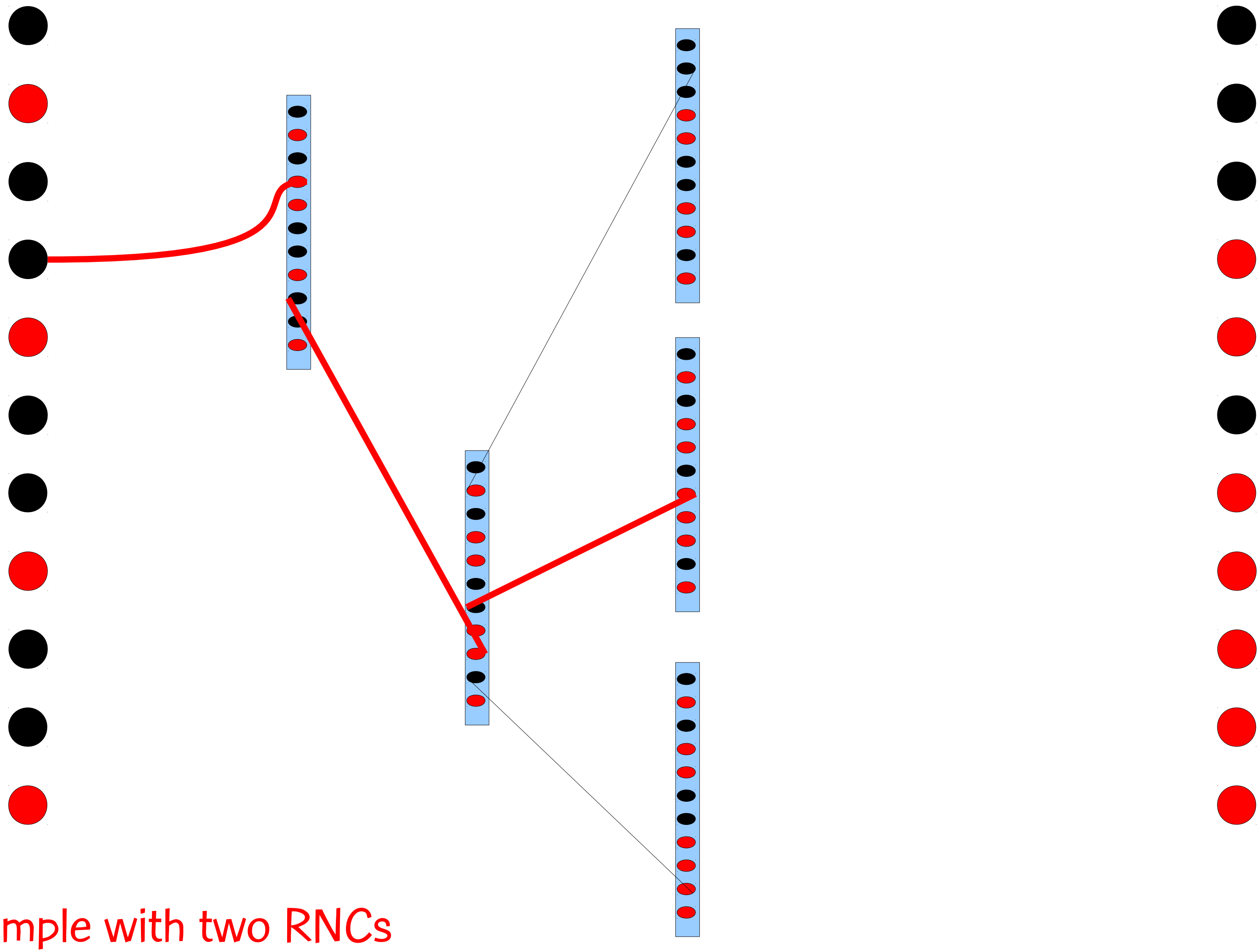


Example with two RNCs

starting solution

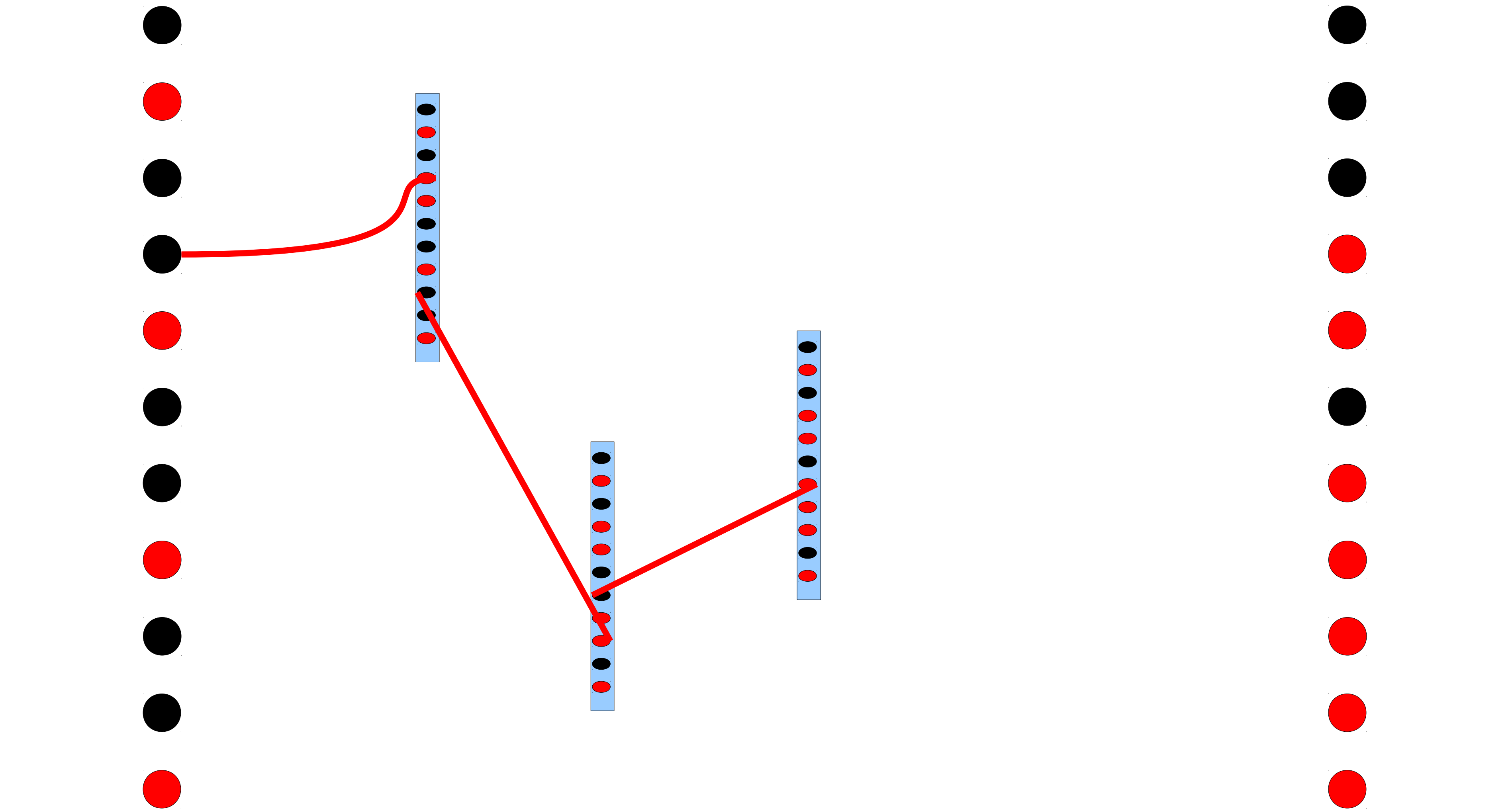
PR example

guiding solution



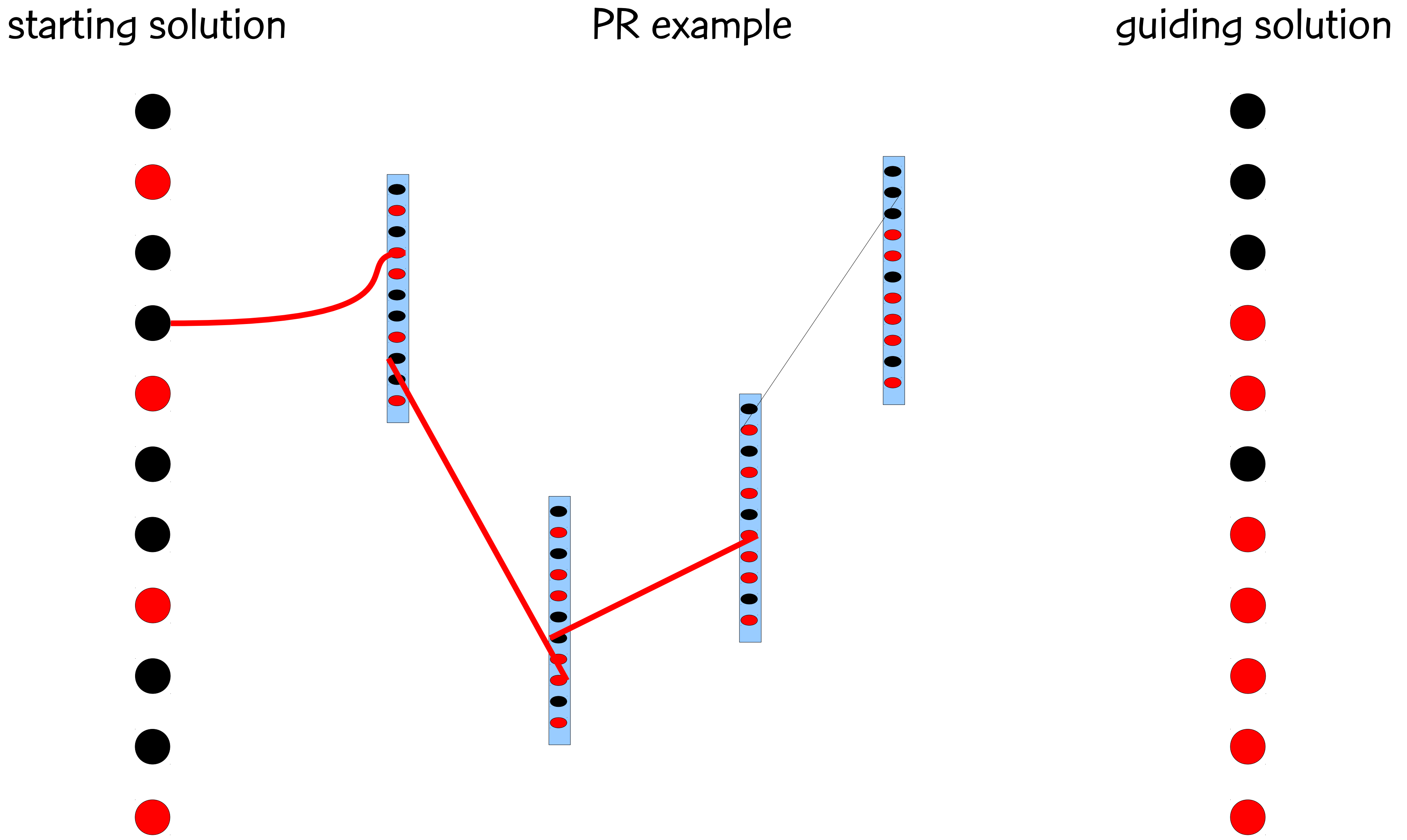
Example with two RNCs

PR example

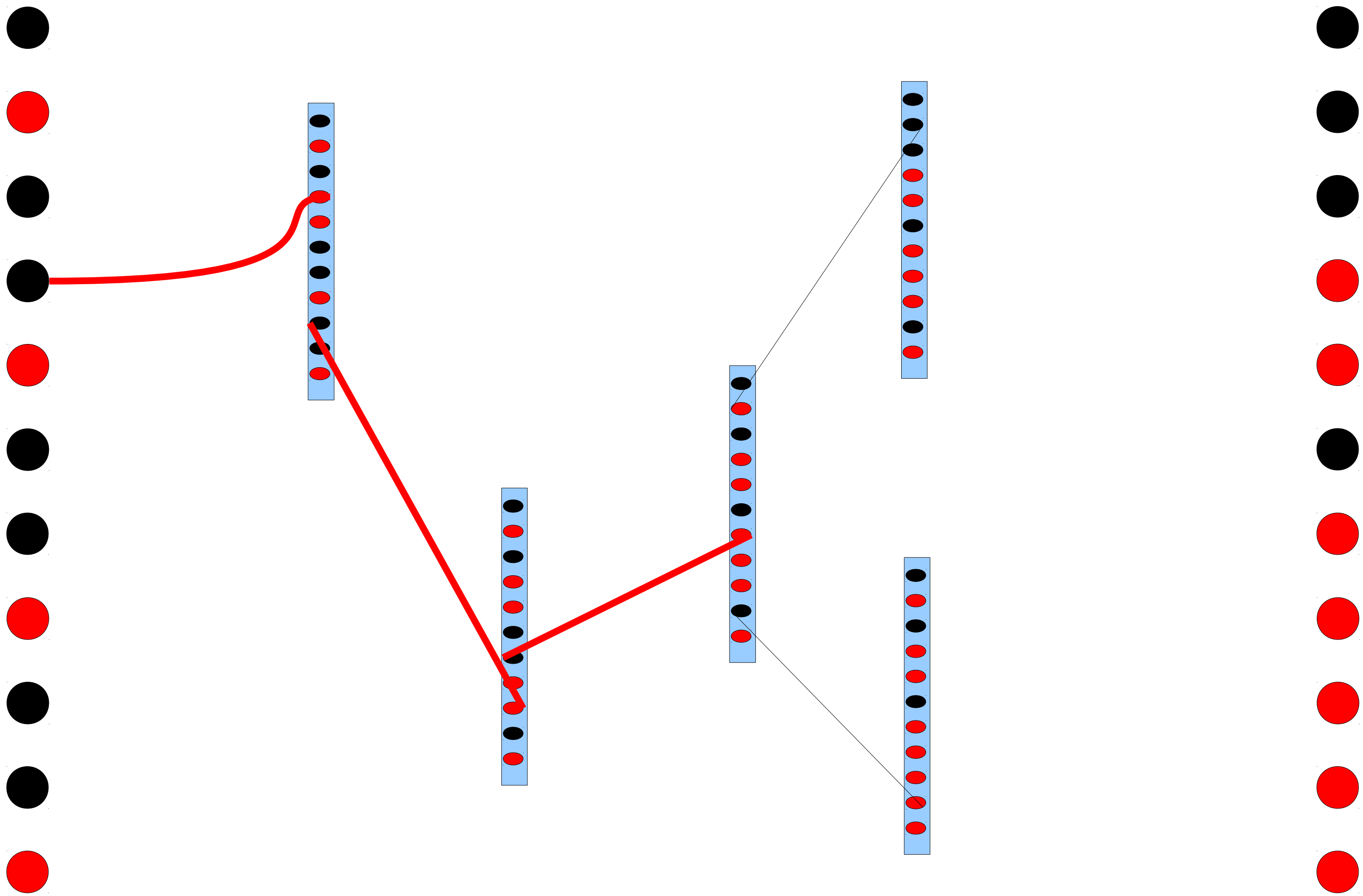


Example with two RNCs

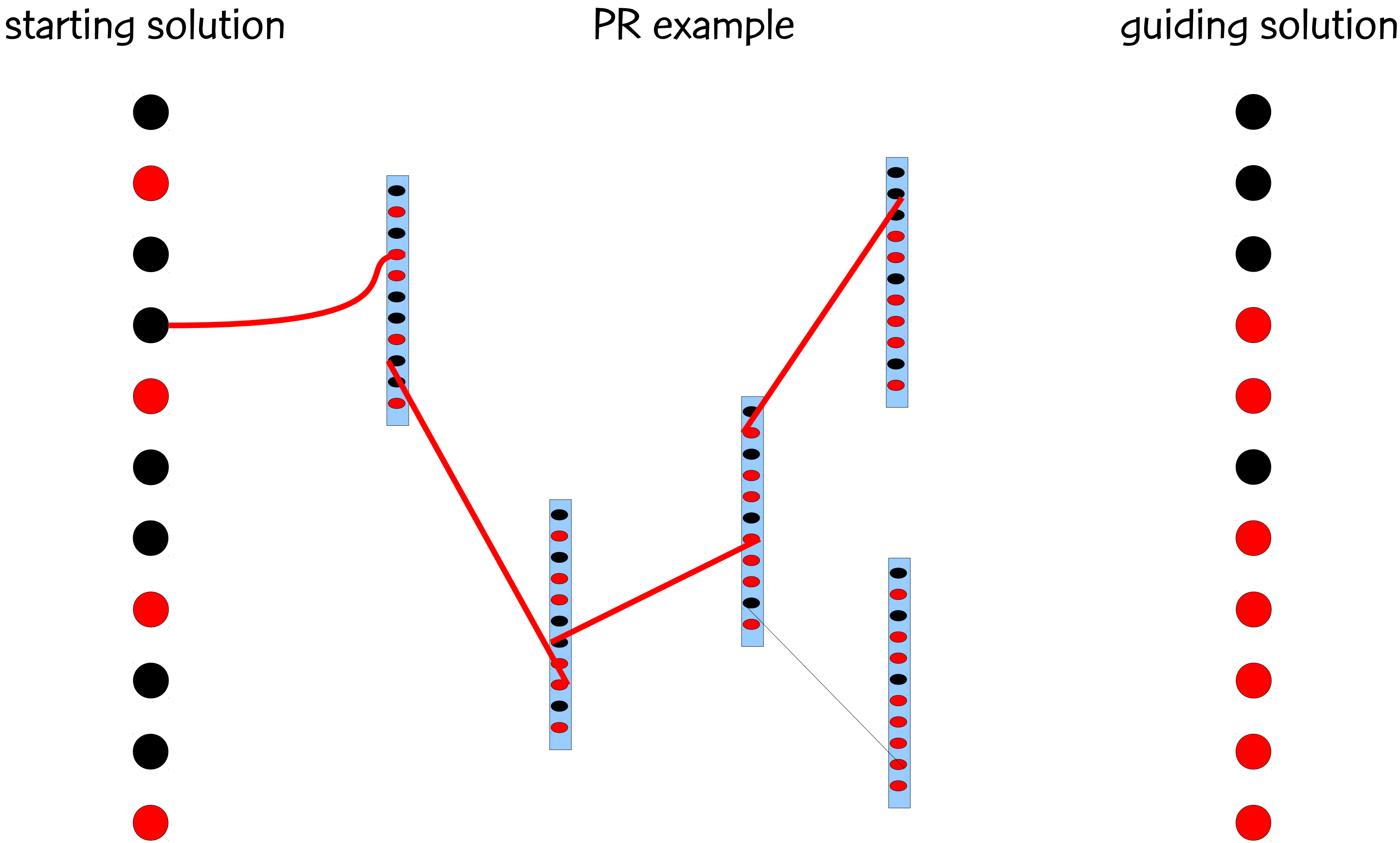
PR example



Example with two RNCs

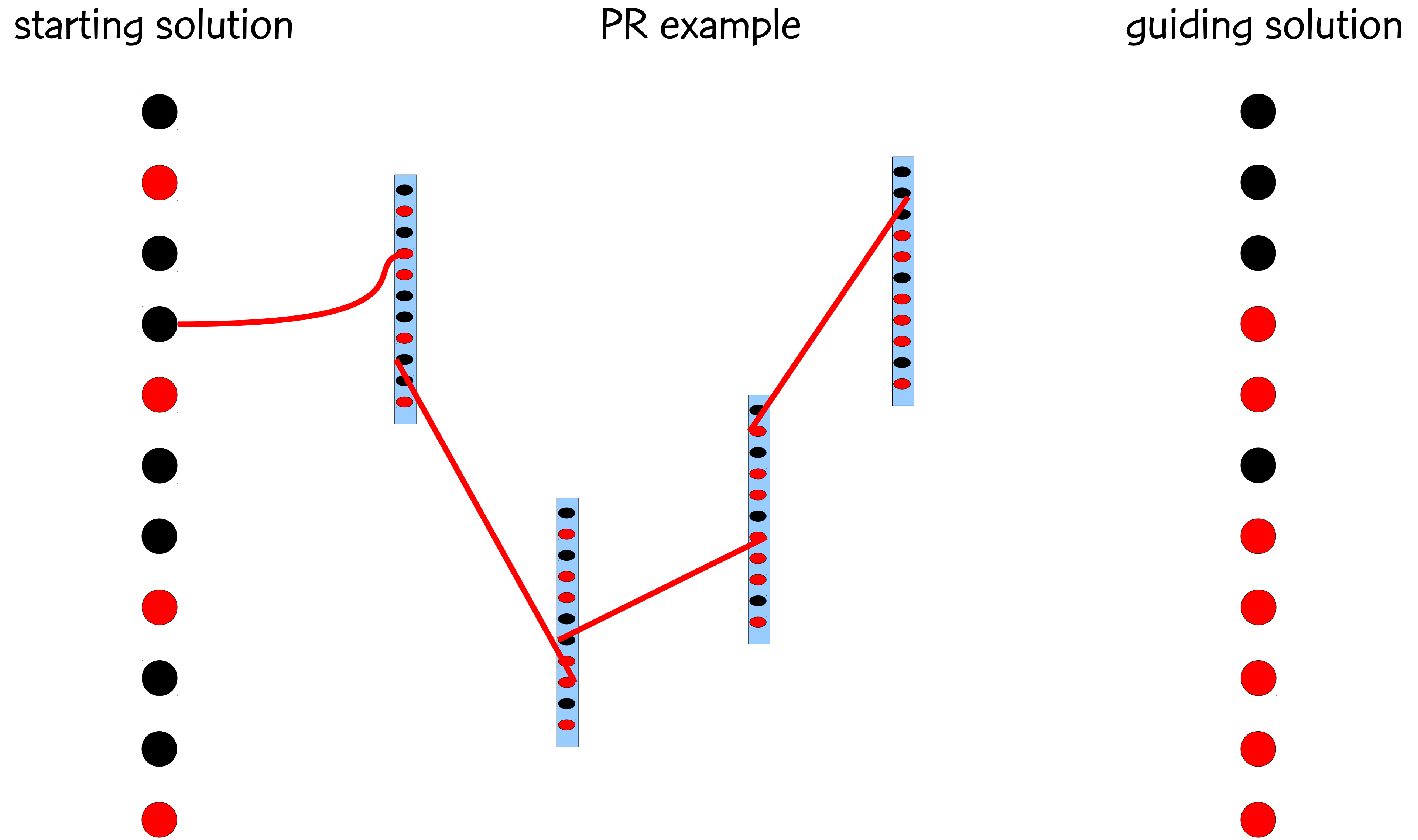


Example with two RNCs



Example with two RNCs

PR example

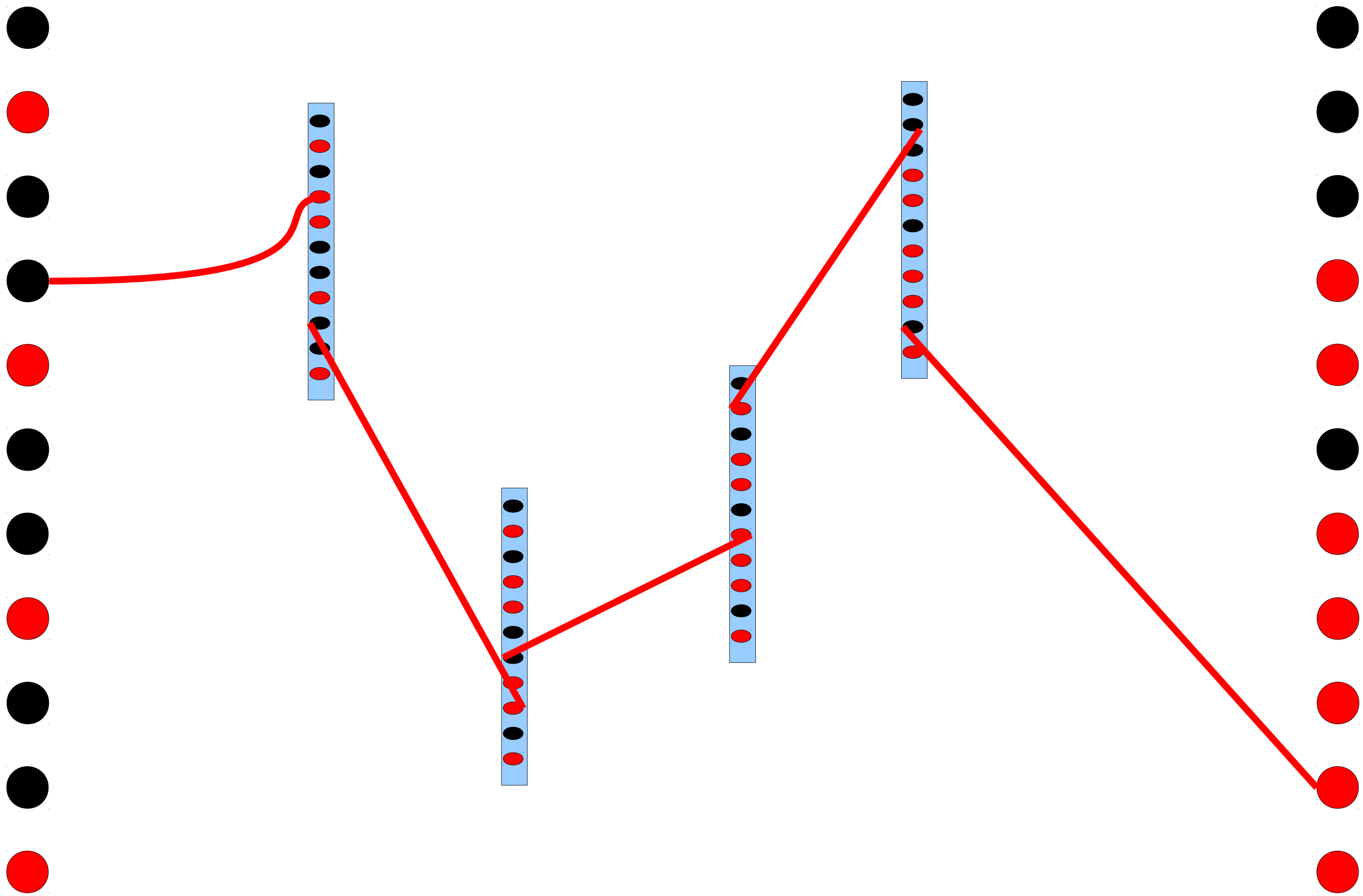


Example with two RNCs

starting solution

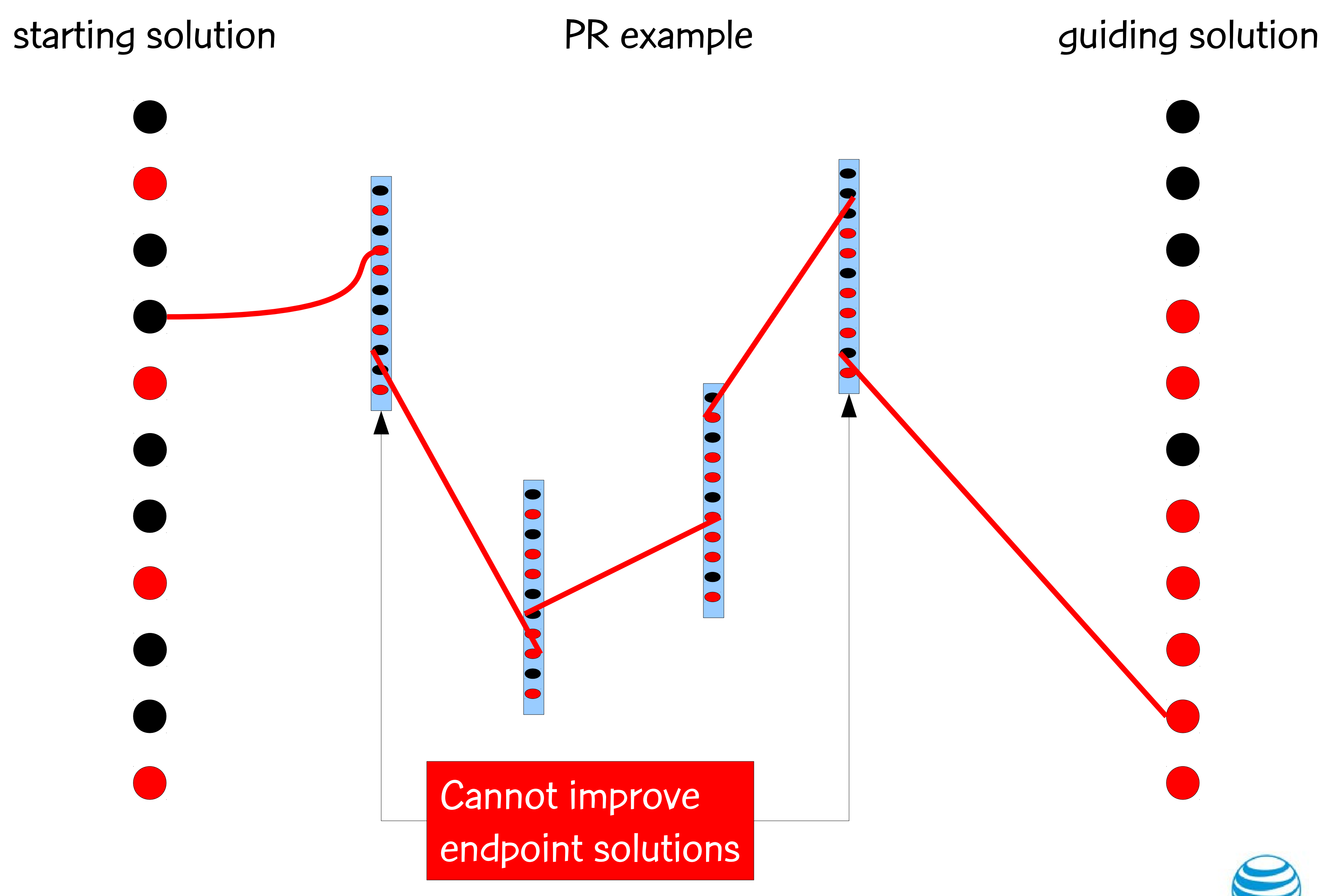
PR example

guiding solution



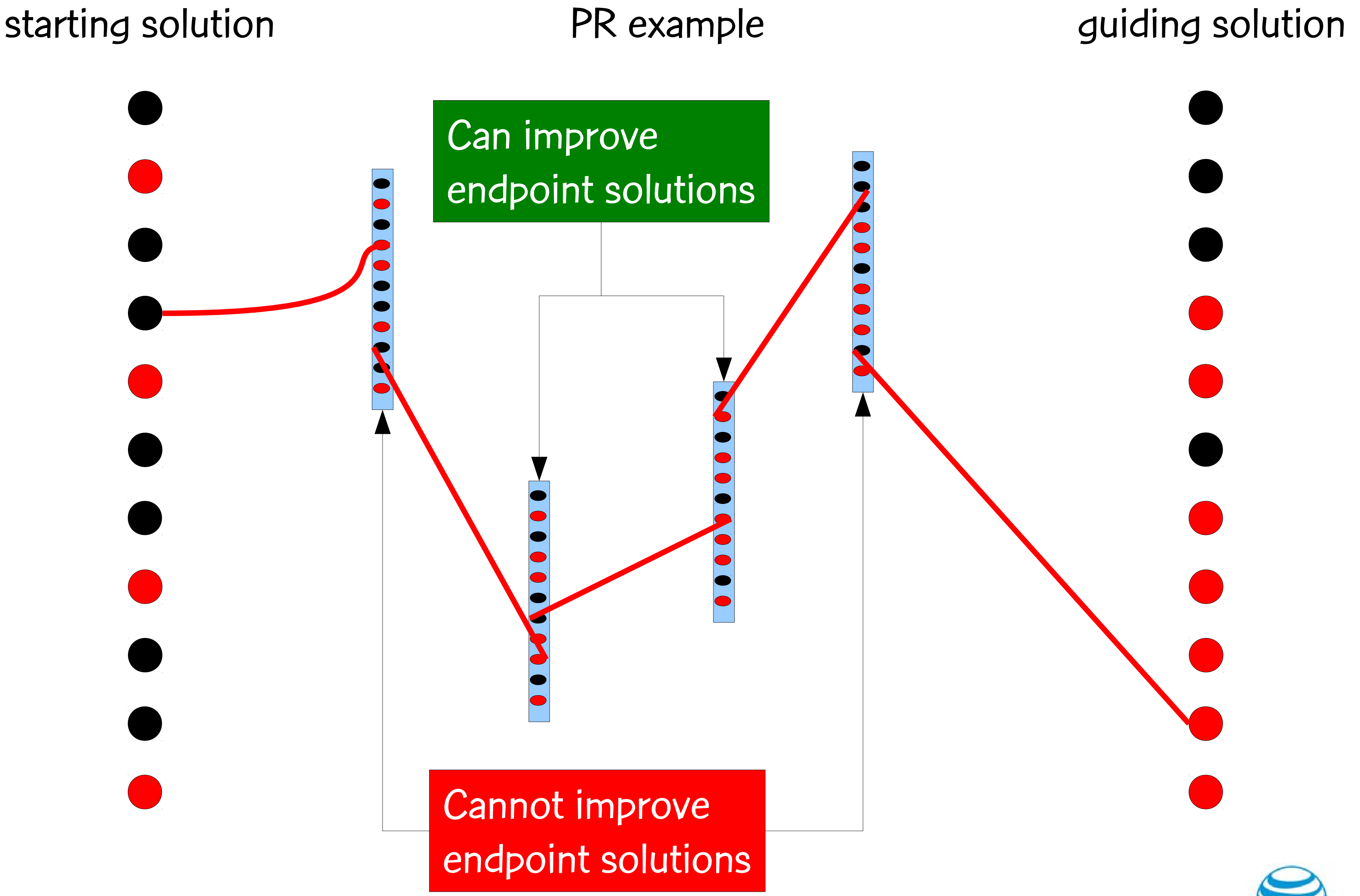
Example with two RNCs

PR example



starting solution
guiding solution
Cannot improve endpoint solutions

PR example


starting solution
guiding solution
Can improve endpoint solutions
Cannot improve endpoint solutions

GRASP with path-relinking

Repeat GRASP with PR loop	1) Construct randomized greedy X 2) Y = local search to improve X 3) Path-relinking between Y and pool solution Z 4) Update pool

Evolutionary path-relinking (EvPR)

Evolutionary path-relinking

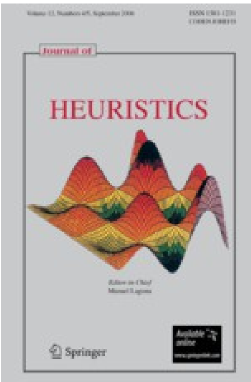
Evolutionary path-relinking "evolves" the pool, i.e. transforms it into a pool of diverse elements whose solution values are better than those of the original pool.

Evolutionary path-relinking can be used as

1) an intensification procedure at certain points of the solution process;

2) a post-optimization procedure at the end of the solution process.

Evolutionary path-relinking proposed in



M.G.C. Resende and R.F. Werneck, A hybrid heuristic for the p-median problem, J. of Heuristics, vol. 10, pp. 59-88, 2004.

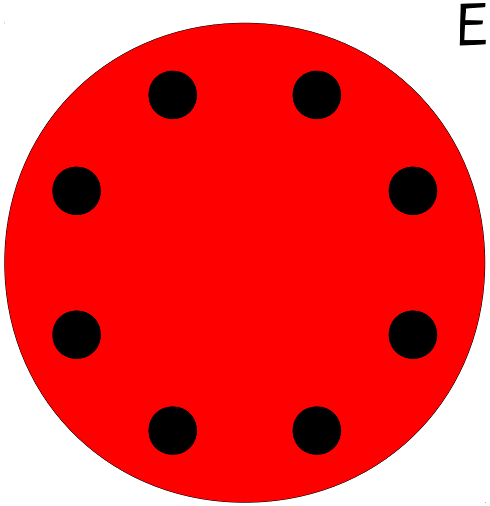
http://www.research.att.com/~mgcr/doc/hhpmedian.pdf

M.G.C. Resende, R. Martí, M. Gallego, and A. Duarte, GRASP and path relinking for the max-min diversity problem, Computers & Operations Research, vol. 37, pp. 498-508, 2010.

http://www.research.att.com/~mgcr/doc/gpr-maxmindiv.pdf

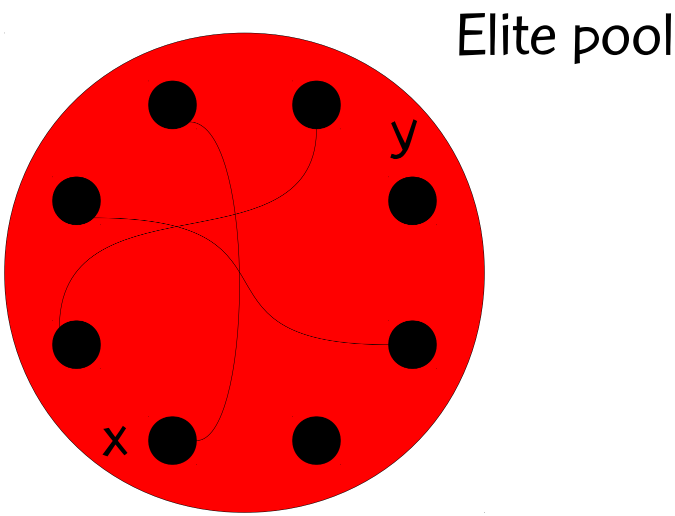
Evolutionary path-relinking (EvPR)

Elite pool



Start with current elite set.

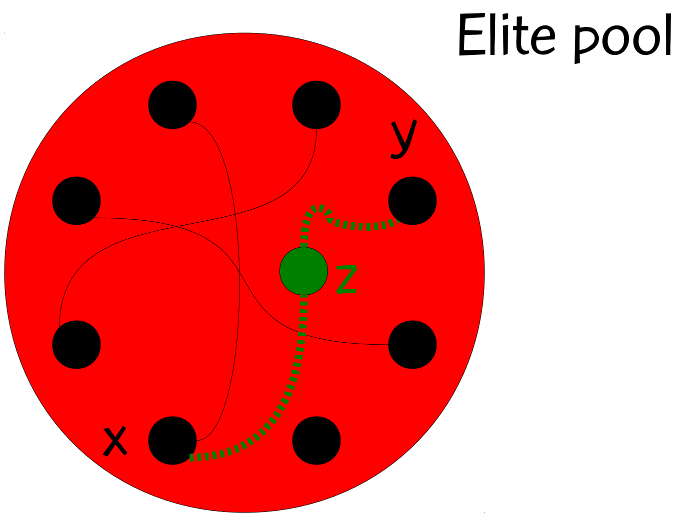
Evolutionary path-relinking (EvPR)



Start with current elite set.

While there is a pair {x,y} of pool solutions that has not yet been relinked:

Evolutionary path-relinking (EvPR)



Start with current elite set.

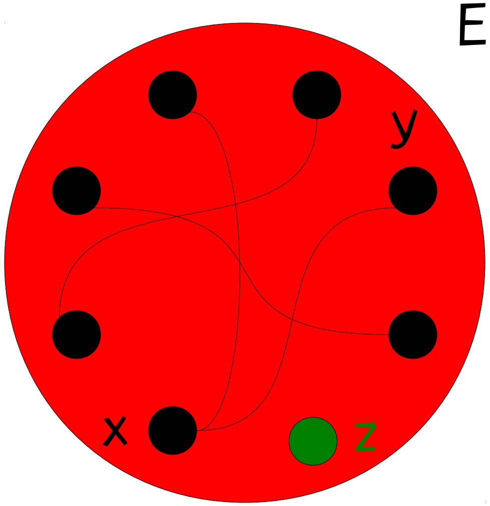
While there is a pair {x,y} of pool solutions that has not yet been relinked:

Relink the pair

$z = PR(x,y)$

Evolutionary path-relinking (EvPR)

Elite pool



Start with current elite set.

While there is a pair {x,y} of pool solutions that has not yet been relinked:

Relink the pair

$z = PR(x,y)$

and attempt to insert z into the pool, replacing some other pool solution.

GRASP with evolutionary path-relinking

As post-optimization

Repeat GRASP with PR loop

1) Construct greedy randomized
2) Local search
3) Path-relinking
4) Update pool

Evolutionary-PR

During GRASP + PR

Repeat outer loop

Repeat inner loop

1) Construct greedy randomized
2) Local search
3) Path-relinking
4) Update pool

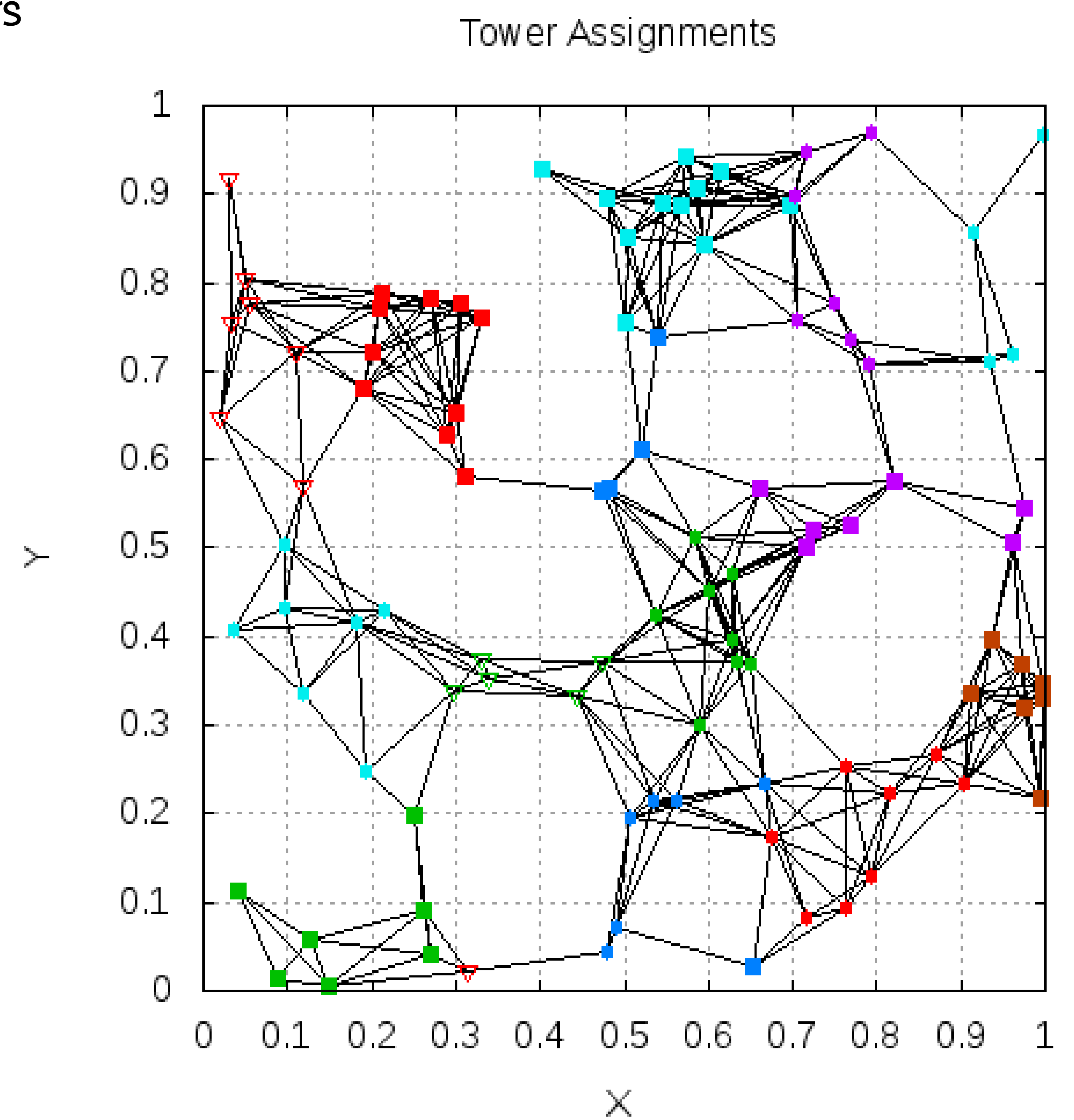
Evolutionary-PR

(Resende & Werneck, 2004, 2006)

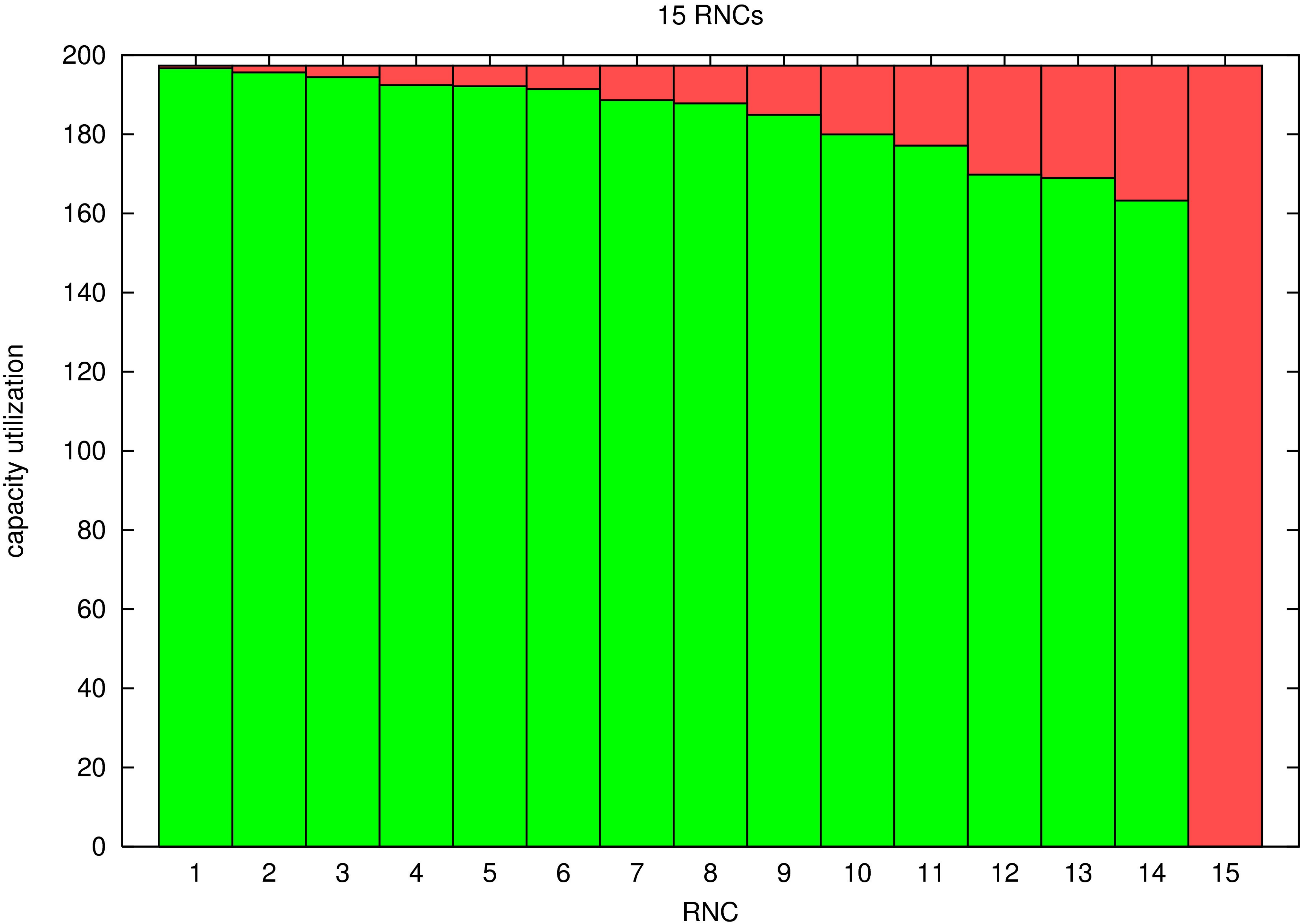
Experiments with GRASP with evPR for HMP

100 towers

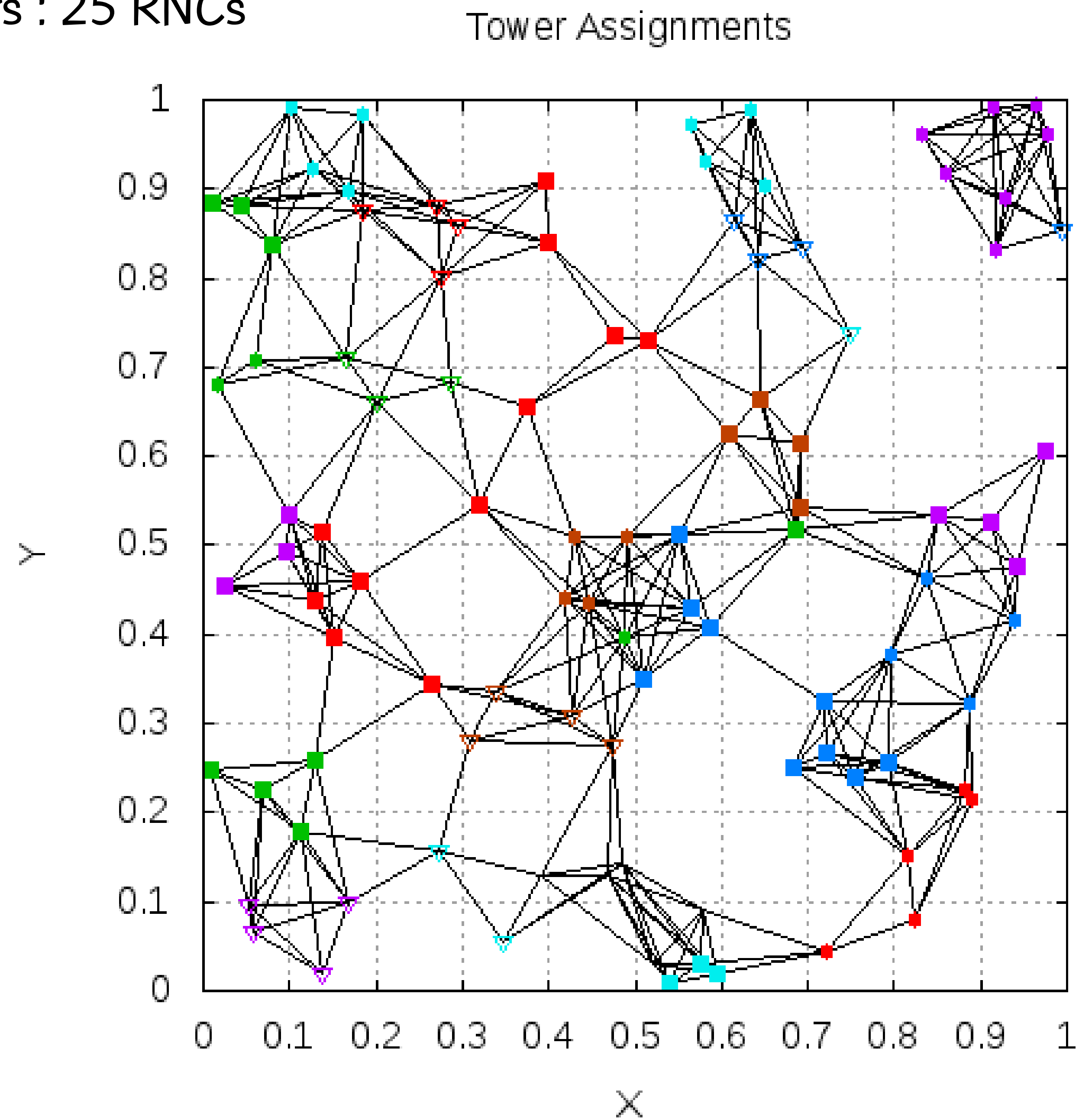
15 RNCs



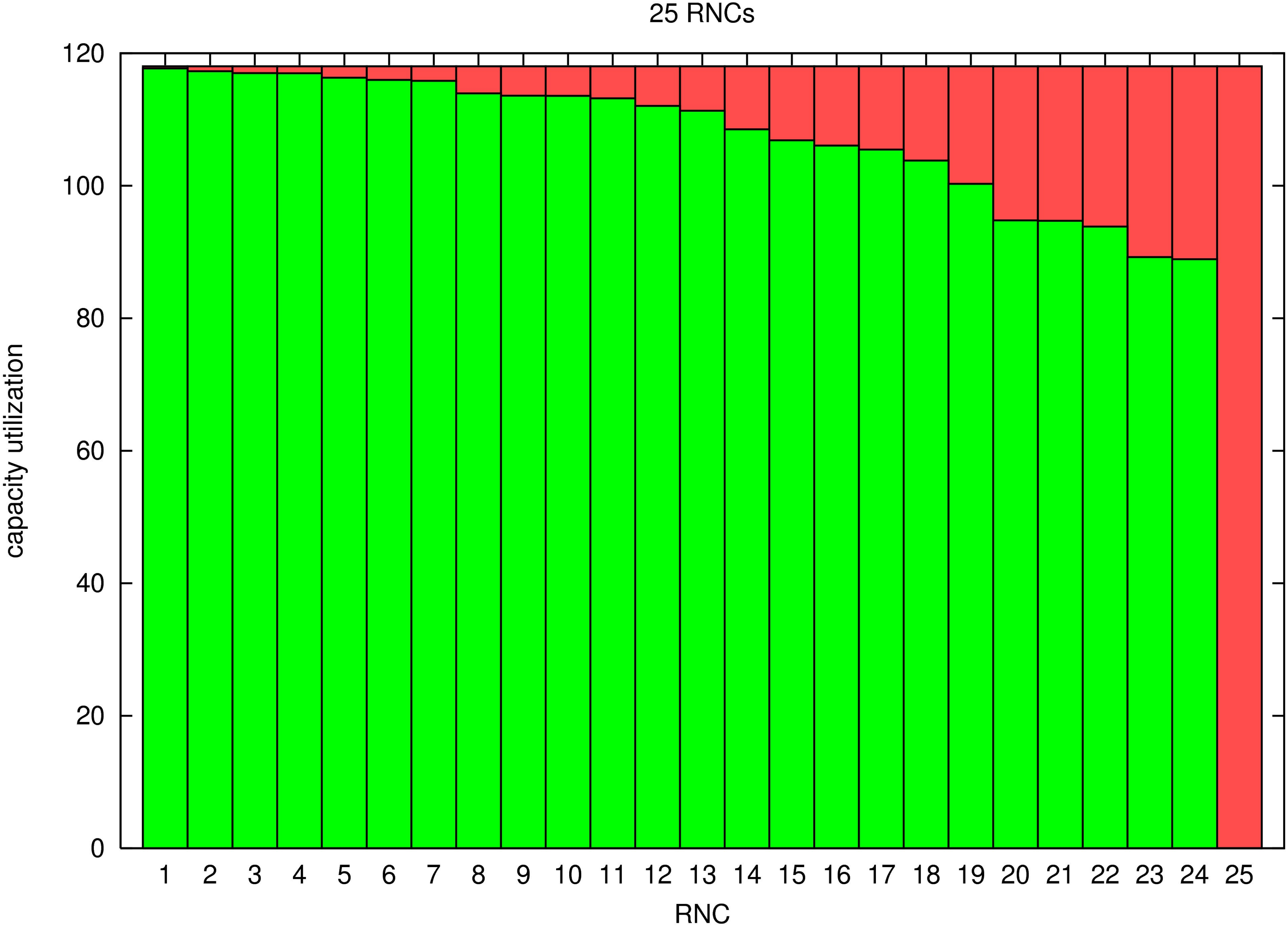
100 towers : 15 RNCs



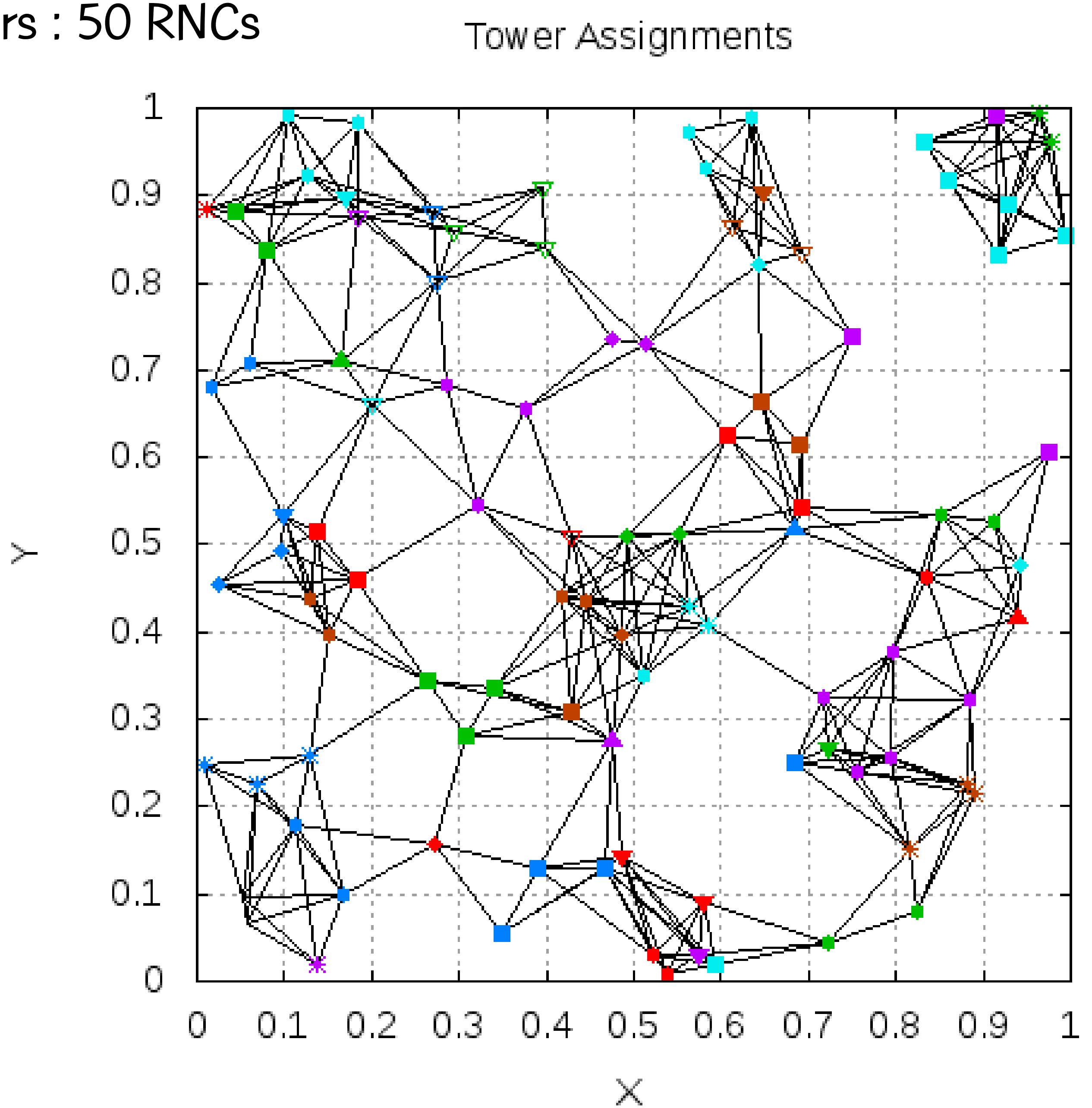
100 towers : 25 RNCs



100 towers : 25 RNCs



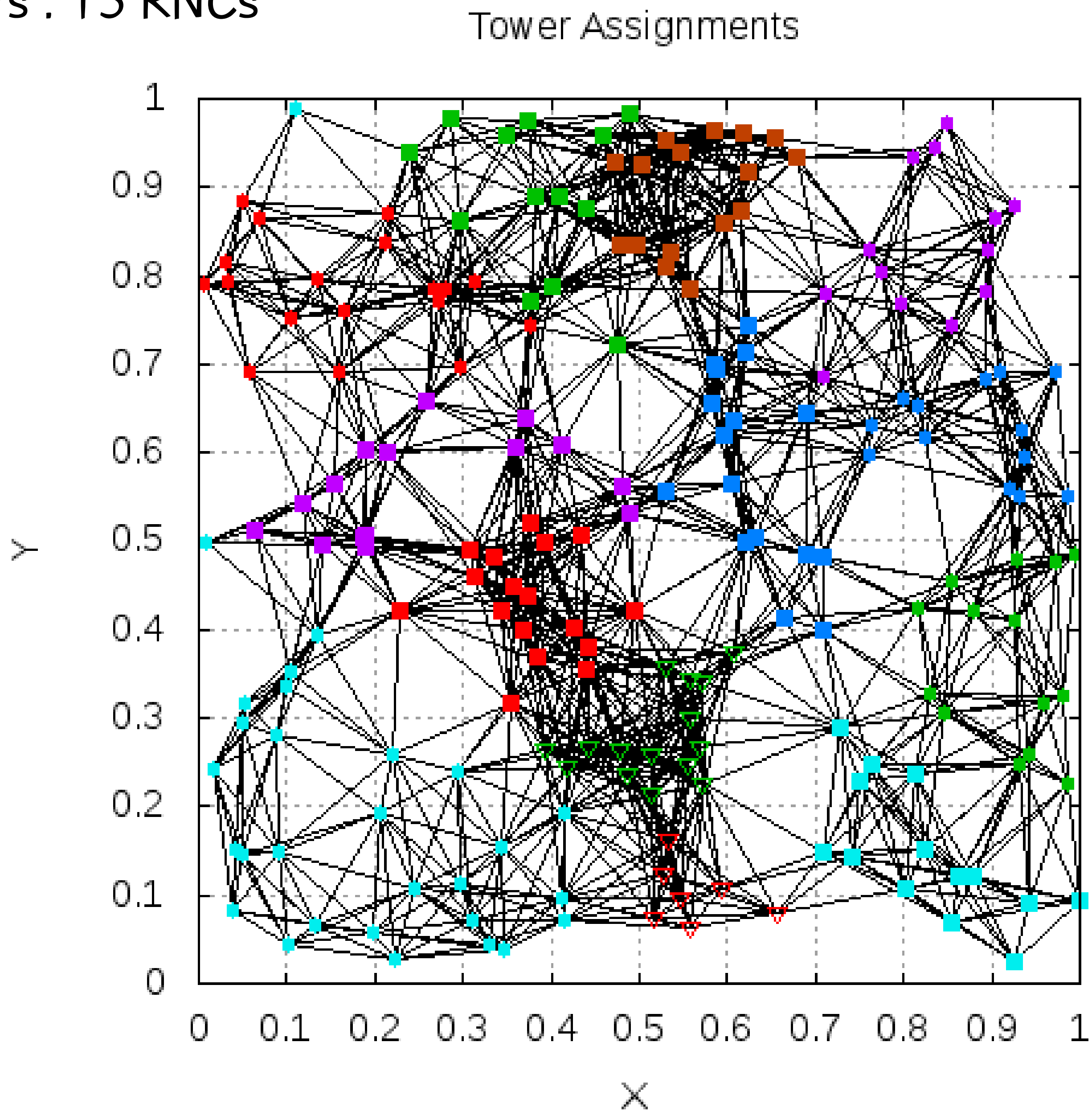
100 towers : 50 RNCs



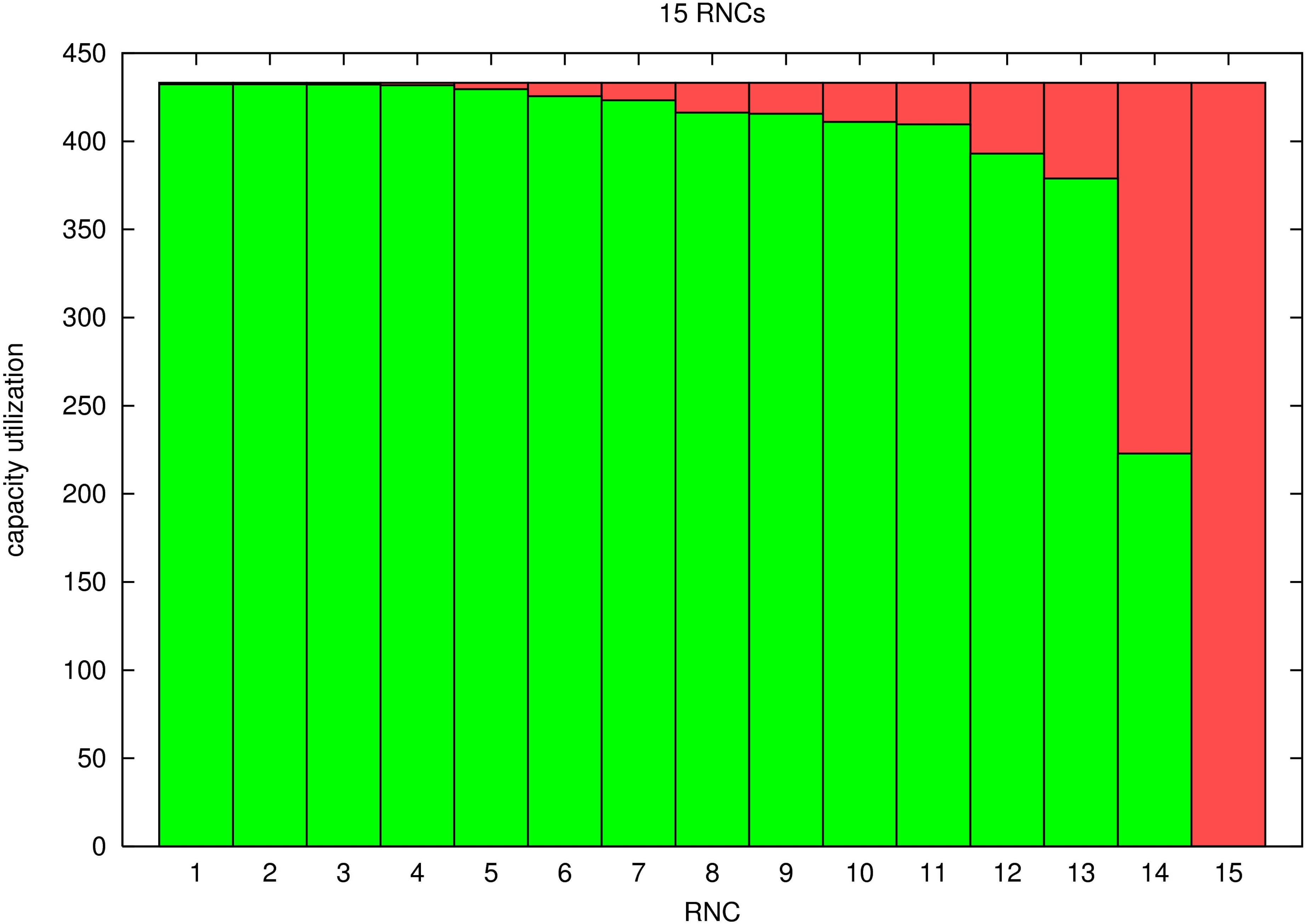
100 towers : 50 RNCs

Your world. Delivered.

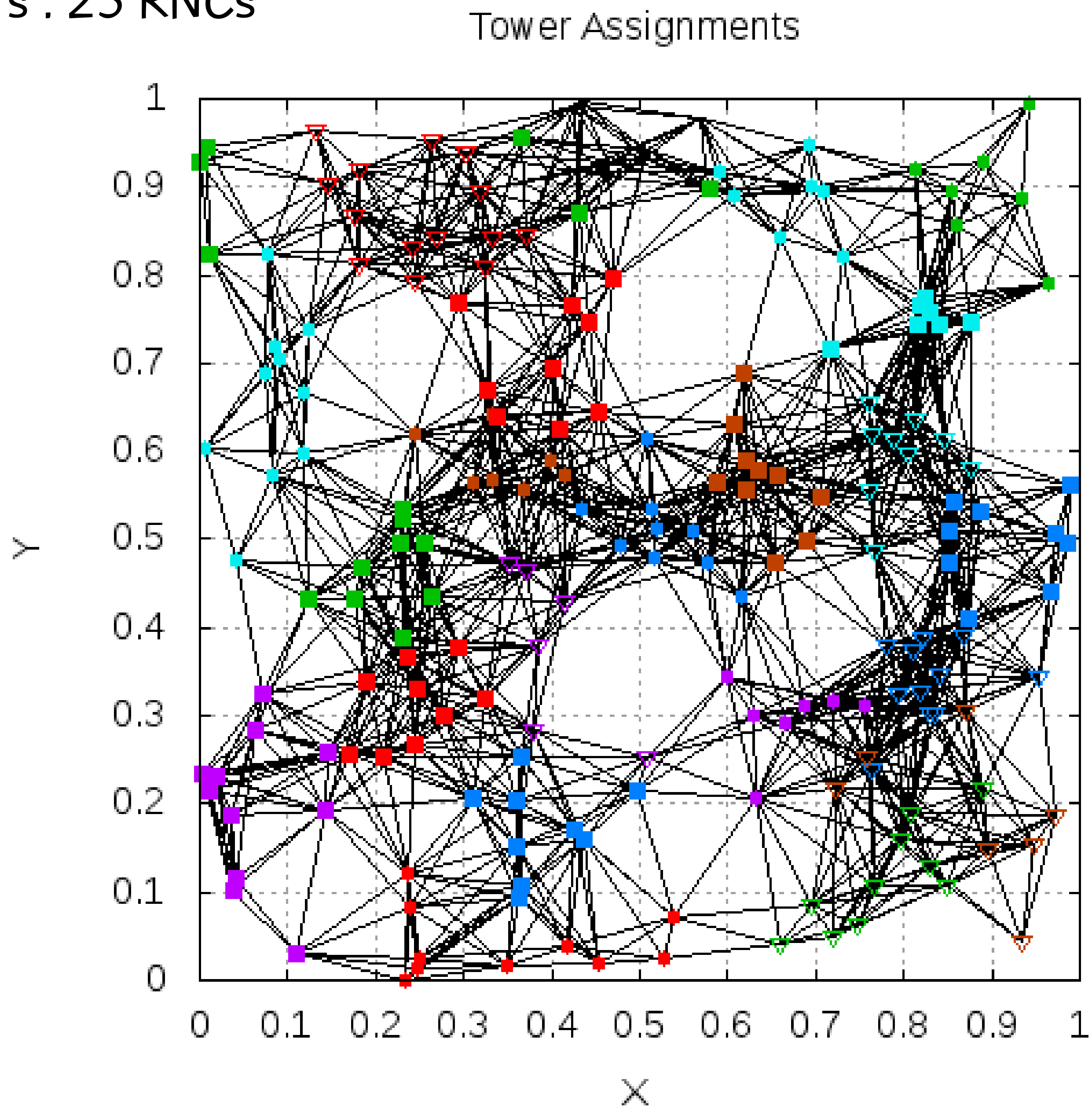
200 towers : 15 RNCs



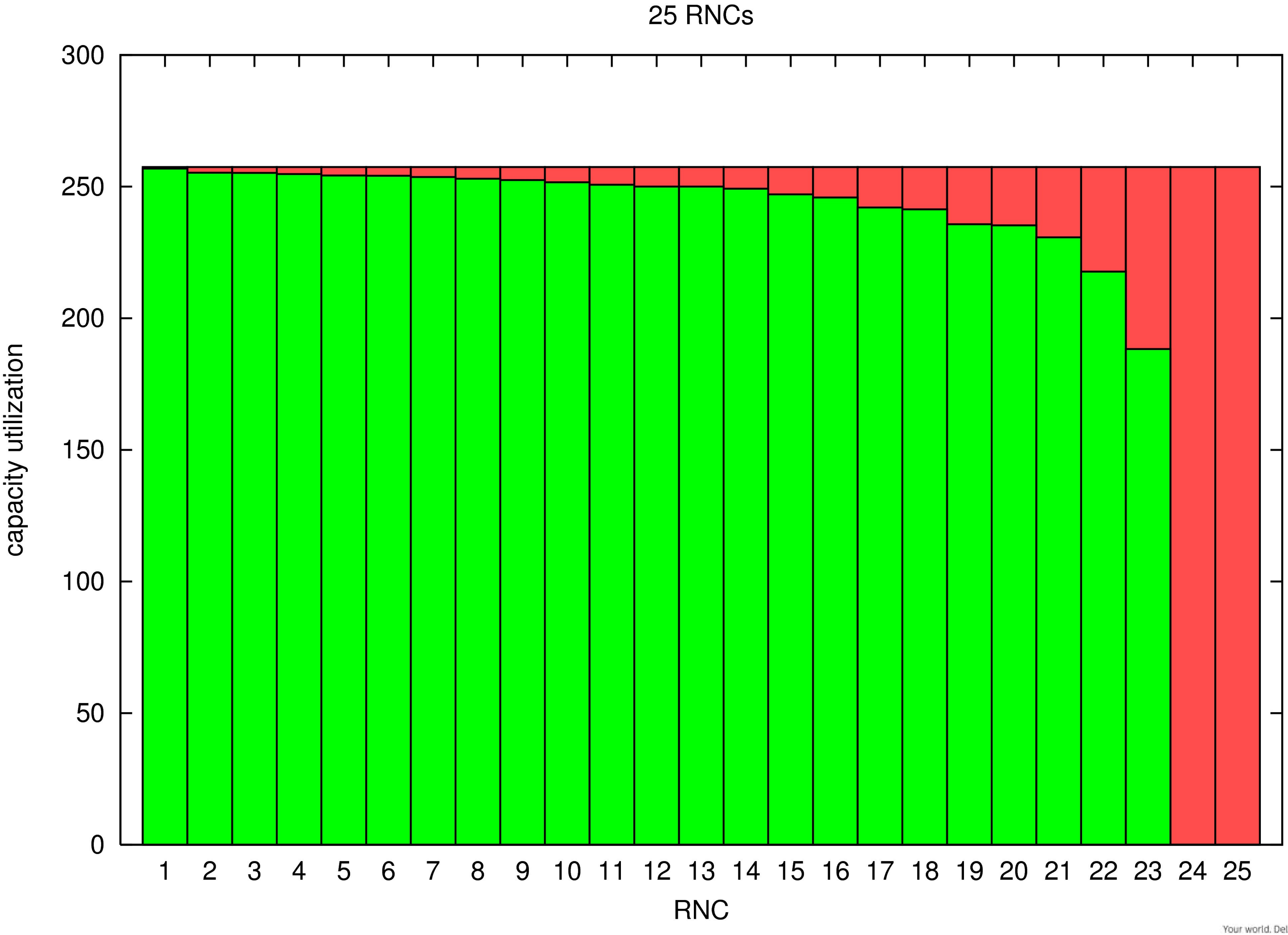
200 towers : 15 RNCs



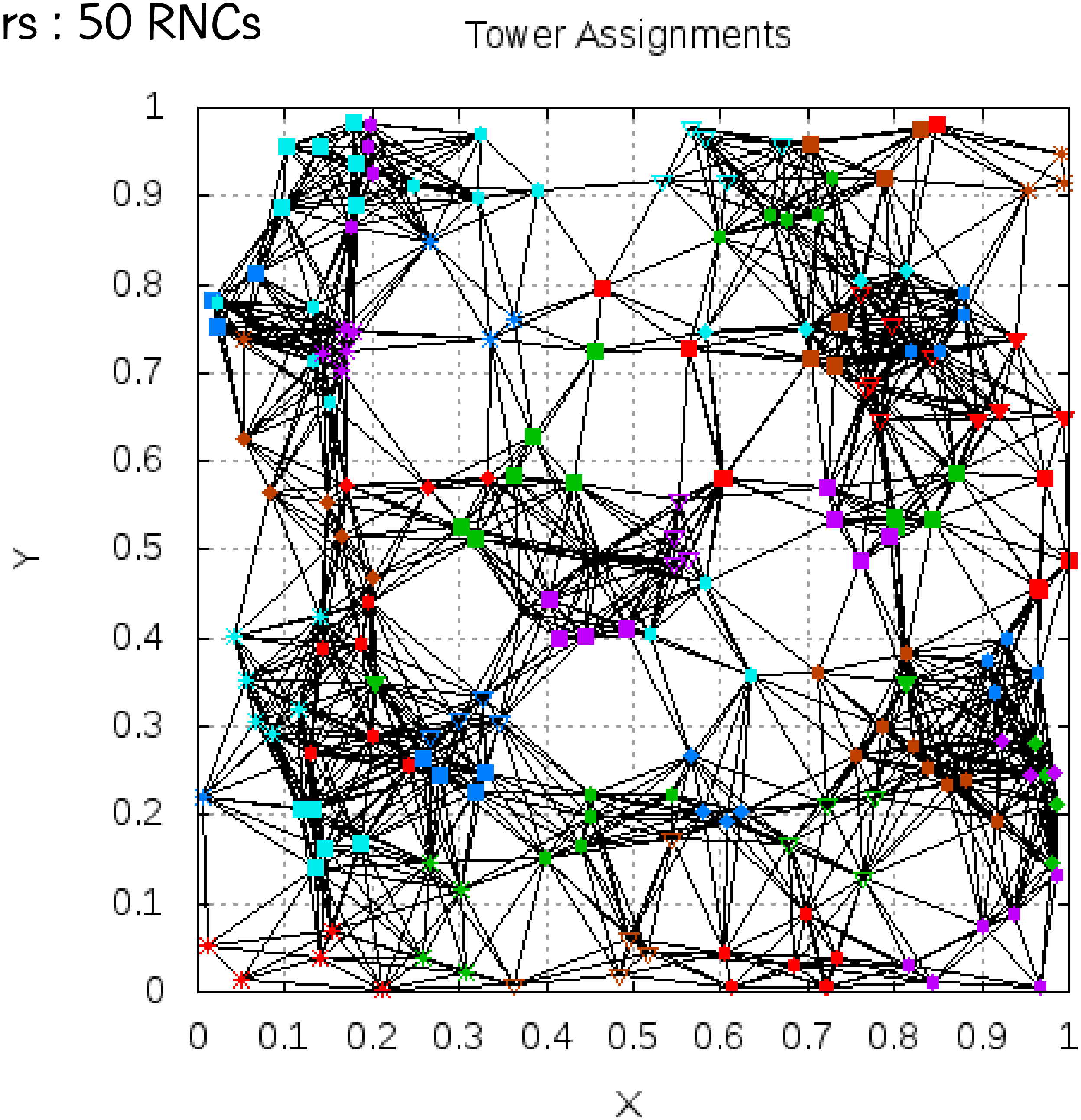
200 towers : 25 RNCs



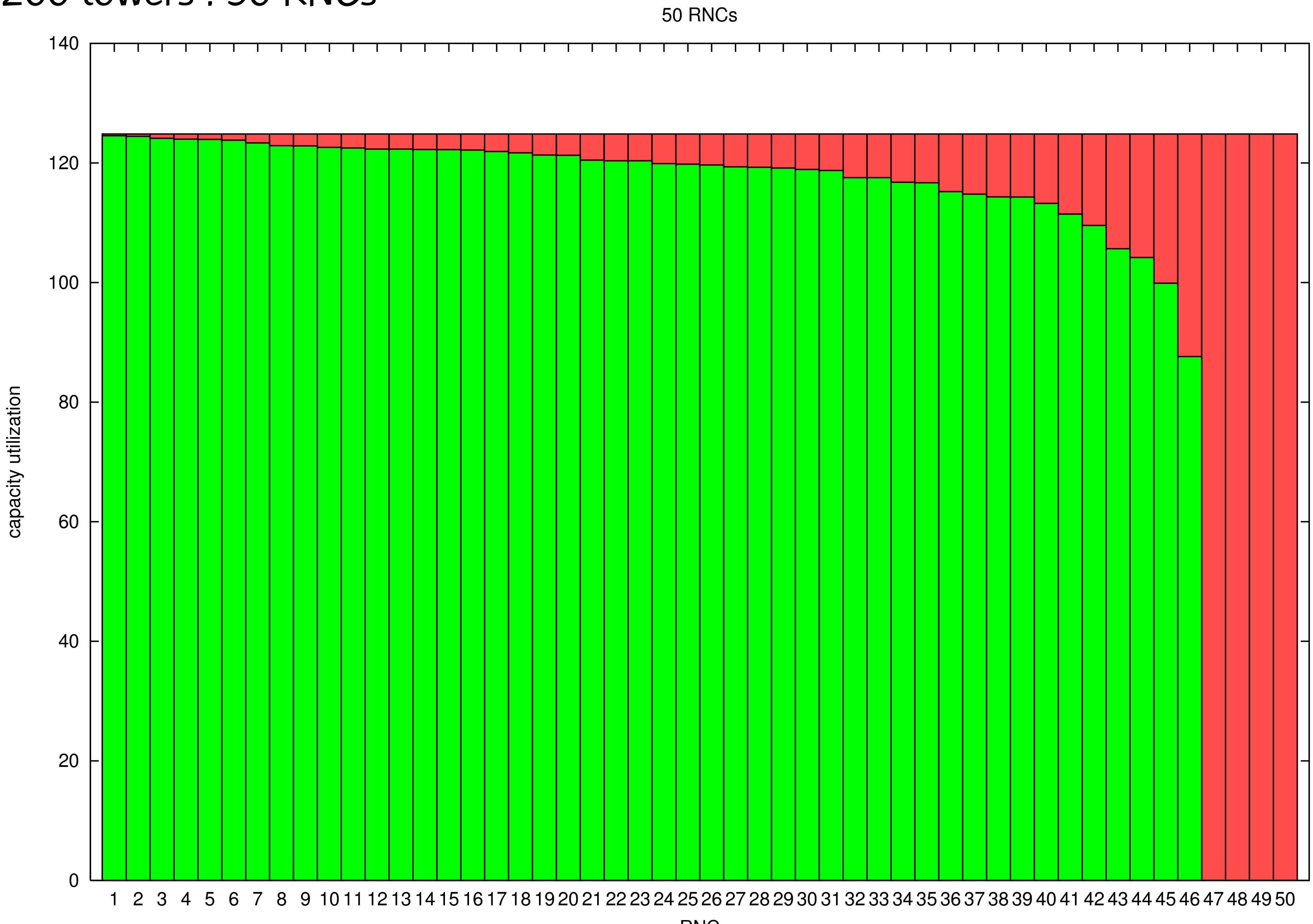
200 towers : 25 RNCs



200 towers : 50 RNCs



200 towers : 50 RNCs



Comparing three randomized heuristics

- GRASP with evolutionary path-relinking for HMP
- GRASP with path-relinking for GQAP
- Biased random-key genetic algorithm for HMP

Small instances: one hour run (2.67 GHz processor)

Instances up to 40 towers & 10 RNCs have known optimal solutions.

Instances		GevPR-HMP		GPR-GQAP		BRKGA		
Towers	RNCs	avg sol	avg time	avg sol	avg time	avg sol	avg time	avg BKS
20	5	381.6	0.0042	381.6	0.7680	381.6	0.2690	381.6
20	10	1900.5	0.0018	1900.5	6.6000	1900.5	0.3555	1900.5
30	5	458.0	0.0552	458.0	3.1300	458.0	0.4426	458.0
30	10	2316.8	0.4178	2316.8	3.9340	2316.8	11.043	2316.8
30	15	4566.5	0.1030	4566.5	4.2175	4566.5	12.125	4566.5
40	5	397.2	2.6066	397.202	22.166	397.2	1.0018	397.2
40	10	2933.6	10.303	2933.6	9.4000	2933.6	157.15	2933.6
40	15	5940.0	6.3414	5940.1	10.462	5940.2	985.01	5940.0

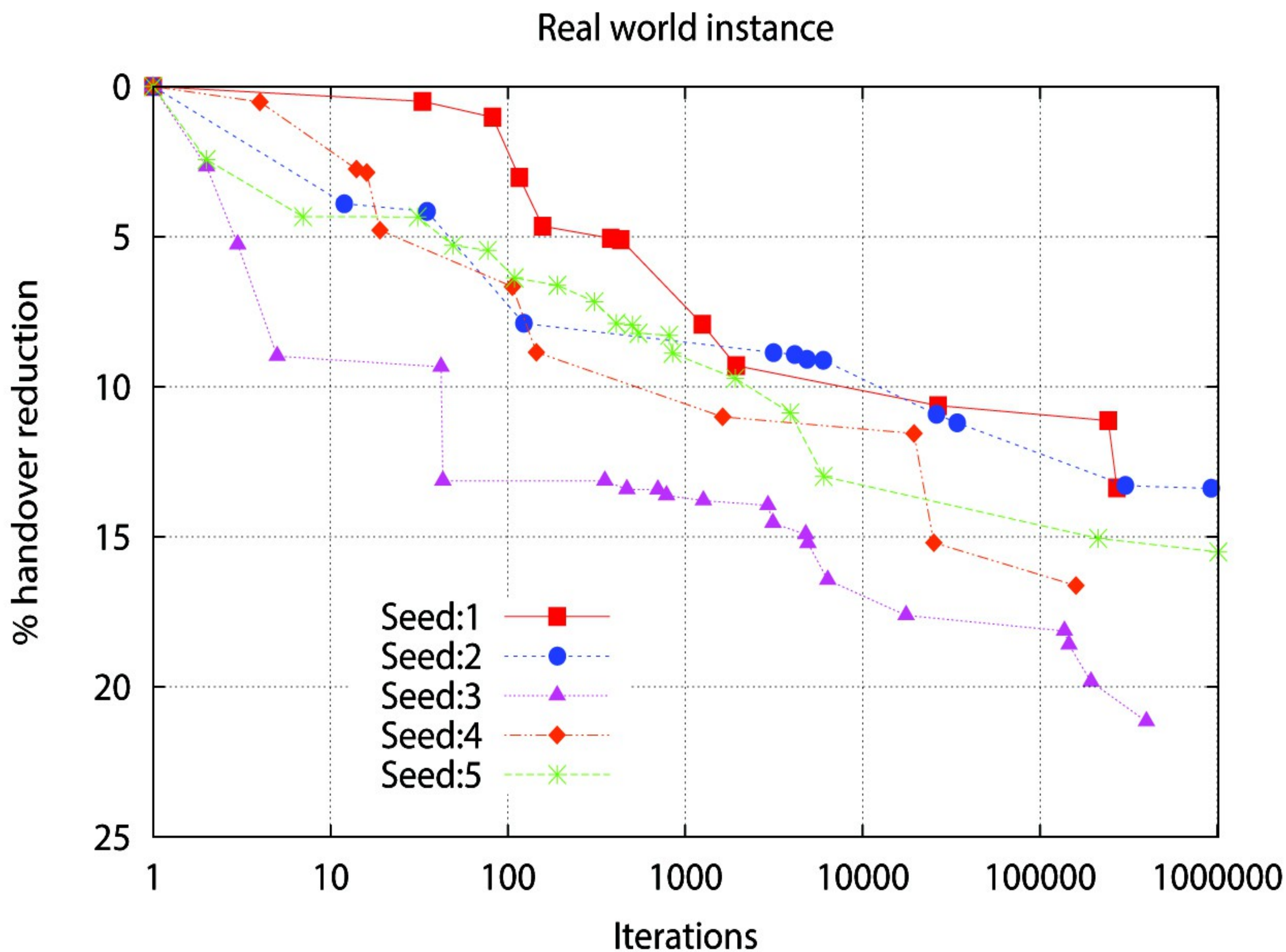
avg time to find best solution (in seconds)

Large instances: one day run (2.67 GHz processor)

Instances		GevPR-HMP		GPR-GQAP		BRKGA		
Towers	RNCs	avg sol	avg best	avg sol	avg best	avg sol	avg best	avg BKS
100	15	19814.4	19814.4	19817.1	19738.0	20255.0	19975.6	19738.0
100	25	36335.0	36221.2	36080.8	35988.4	36682.2	36354.0	35981.2
100	50	59407.3	59313.6	59441.8	59400.4	60264.2	60052.4	59304.4
200	15	85370.0	84984.4	88878.1	86759.6	86797.4	86093.6	84984.4
200	25	137991	137120	144542	142191	143106	141925	137120
200	50	221693	221048	224038	222818	223636	222874	220237
400	15	362337	359597	464227	445866	372774	369687	359121
400	25	547971	544243	678339	655570	561596	557724	543561
400	50	832416	829088	935032	927971	860906	857131	829088

For each size, there are 5 instances and for each there were 5 runs.

Progress of best solution for five independent runs of GevPR-HMP on a real instance with about 1000 towers and 30 RNCs.



Concluding remarks

- We described the handover minimization problem (HMP).
- Objective of handover minimization is to reduce number of dropped calls in a cellular network.
- The HMP is a special case of the generalized quadratic assignment problem (GQAP).
- We described experiments with three randomized heuristics for the HMP on synthetic instances of the problem and one real instance.

Concluding remarks

- We described the handover minimization problem (HMP).
- Objective of handover minimization is to reduce number of dropped calls in a cellular network.
- The HMP is a special case of the generalized quadratic assignment problem (GQAP).
- We described three randomized heuristics for the the HMP and applied them on synthetic instances of the problem and one real instance. GRASP with evolutionary PR turns out to be the best (w.r.t to solution quality x solution time) so far ...

Thanks!

Technical report: L.F. Morán-Mirabal, J.L. González-Velarde, MGCR, & R.M.A. Silva, "Randomized heuristics for handover minimization in mobiity networks" is available online at

http://www.research.att.com/~mgcr