Solving handover minimization as a generalized quadratic assignment problem

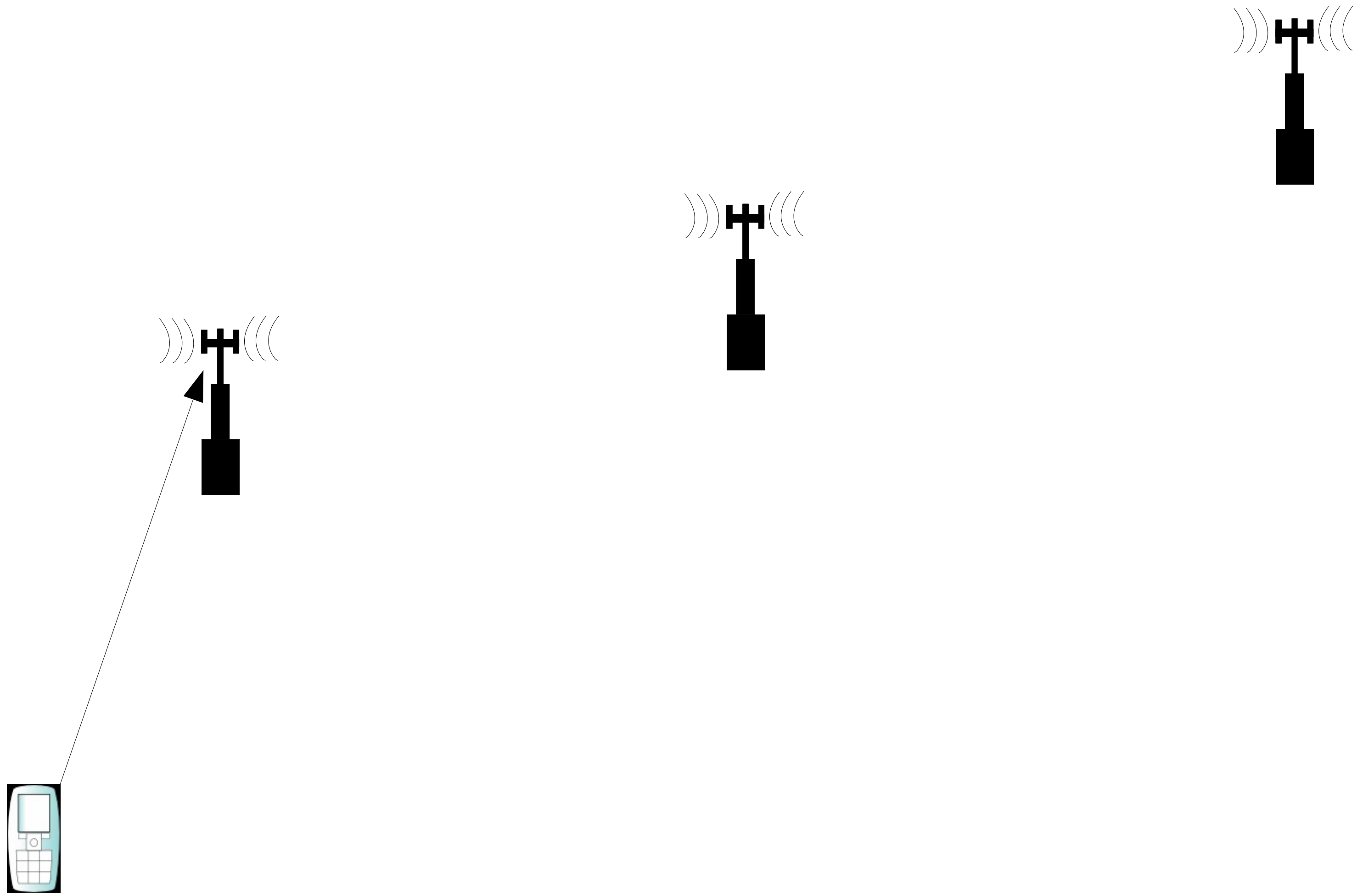
Ricardo M. A. Silva (UFPE)
Mauricio G. C. Resende (AT&T Research)

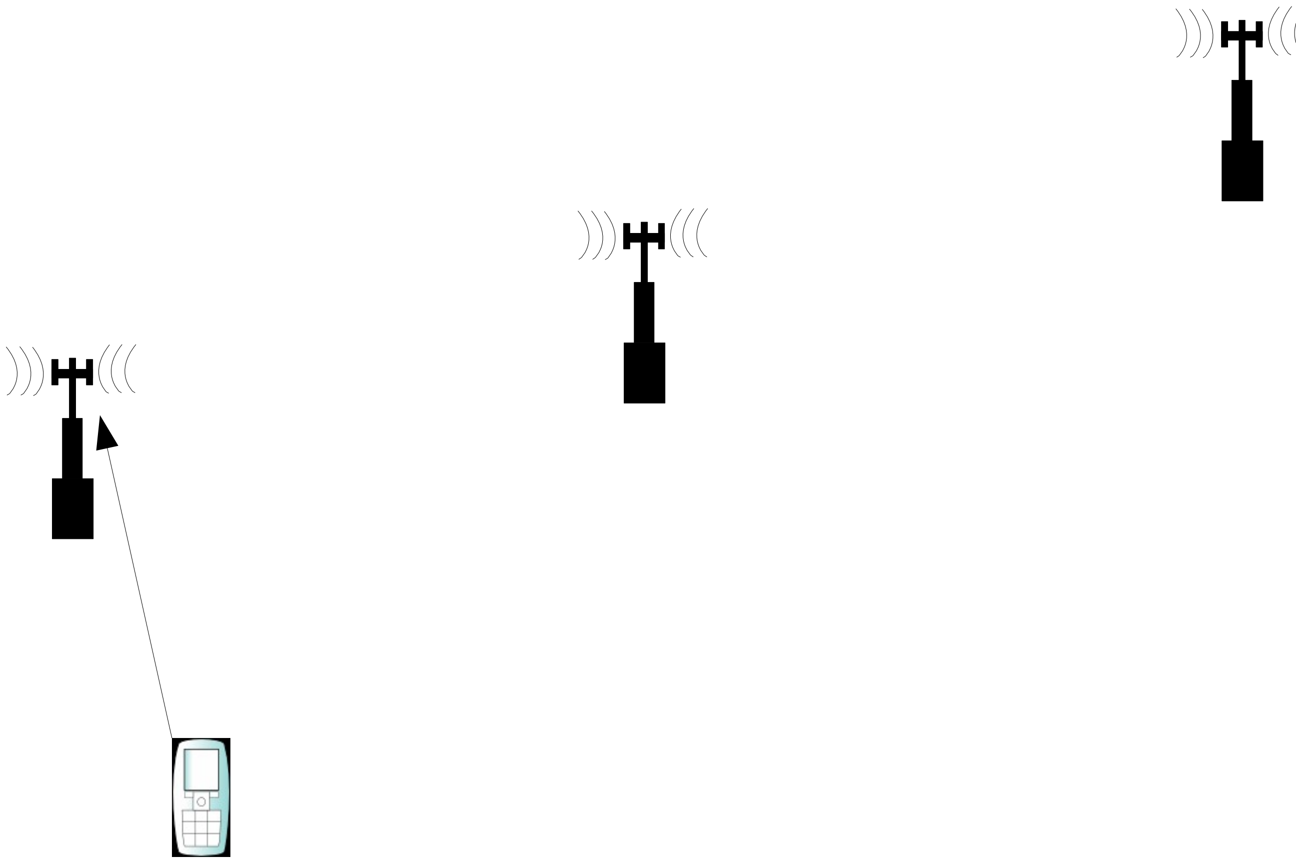
2012 INFORMS Telecommunications Conference
Boca Raton, Florida ☼ March 15-17, 2012

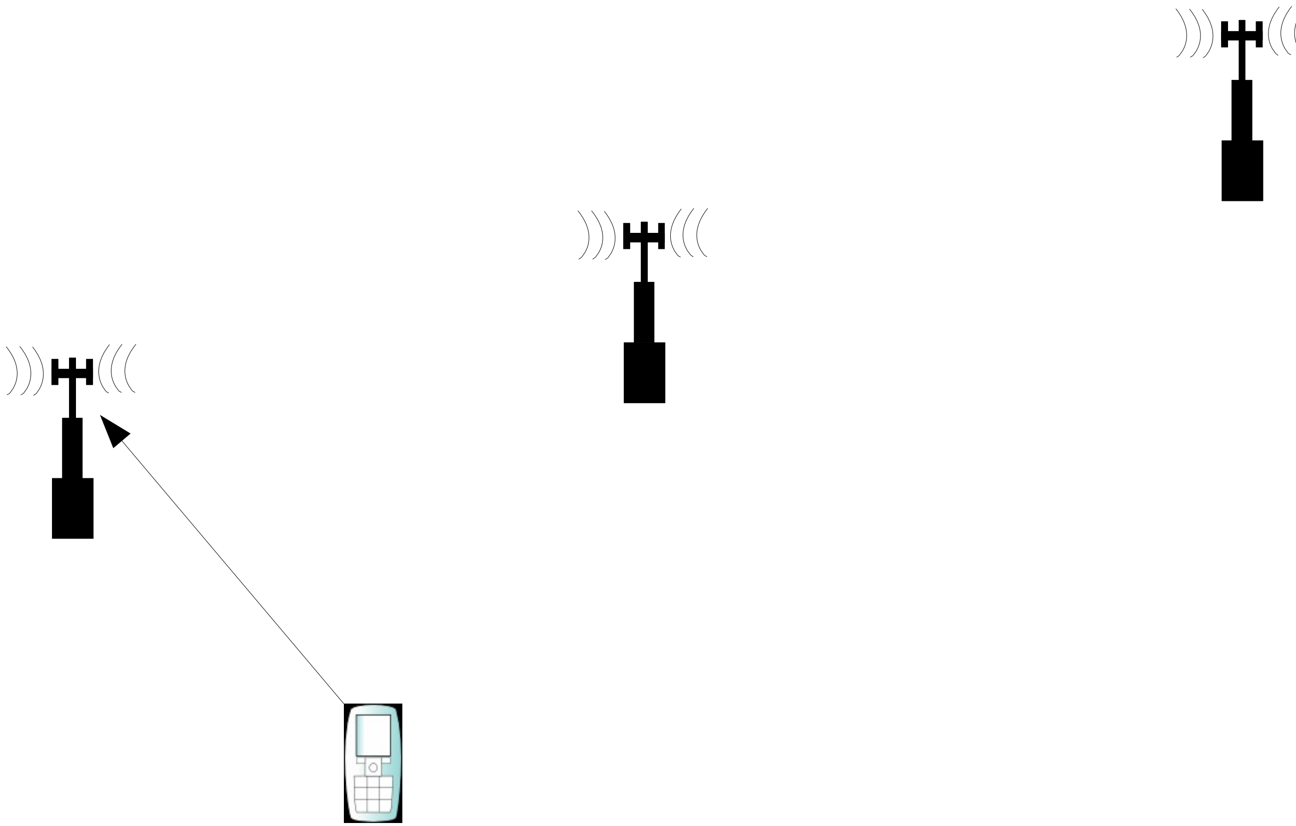
Outline

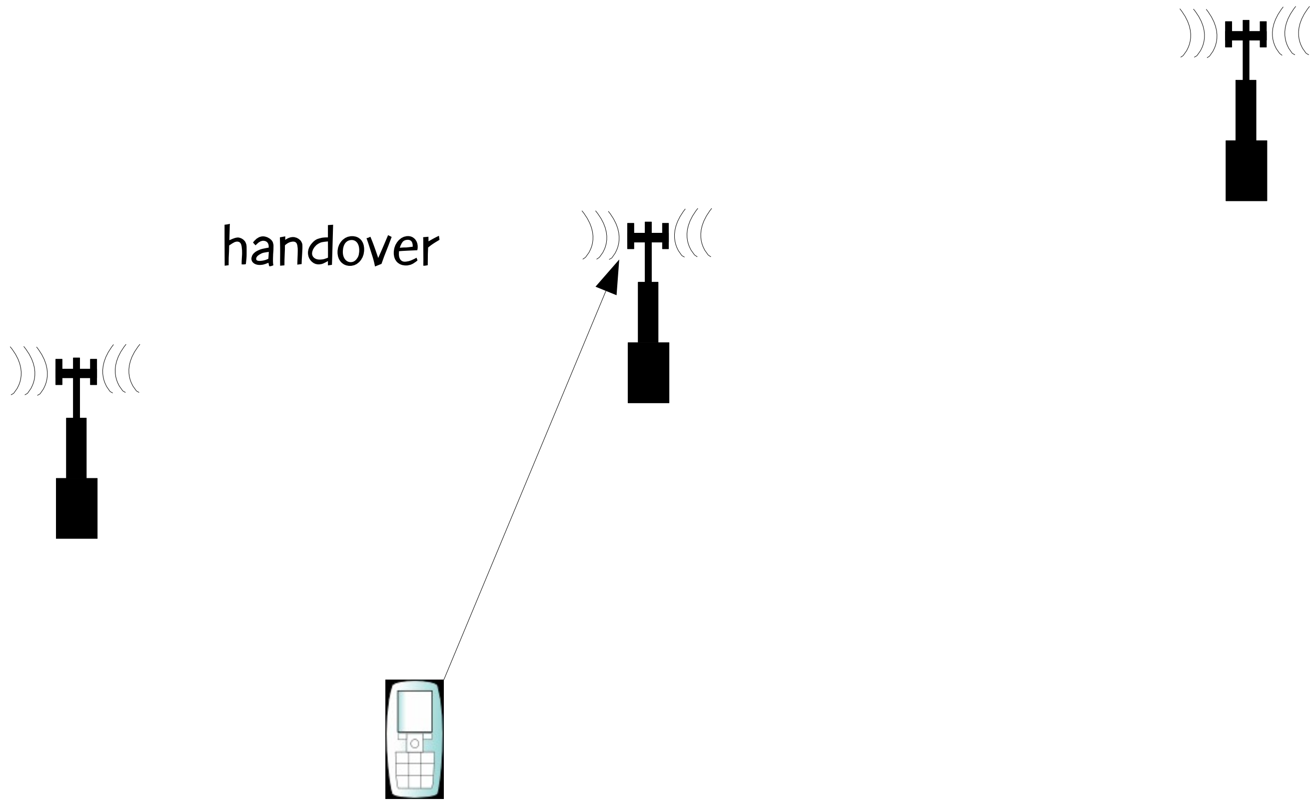
- Handover minimization
- Generalized quadratic assignment problem (GQAP)
- Handover minimization is a special case of GQAP
- Some experiments with synthetic networks
- Solution procedure: GRASP with path-relinking
- Concluding remarks

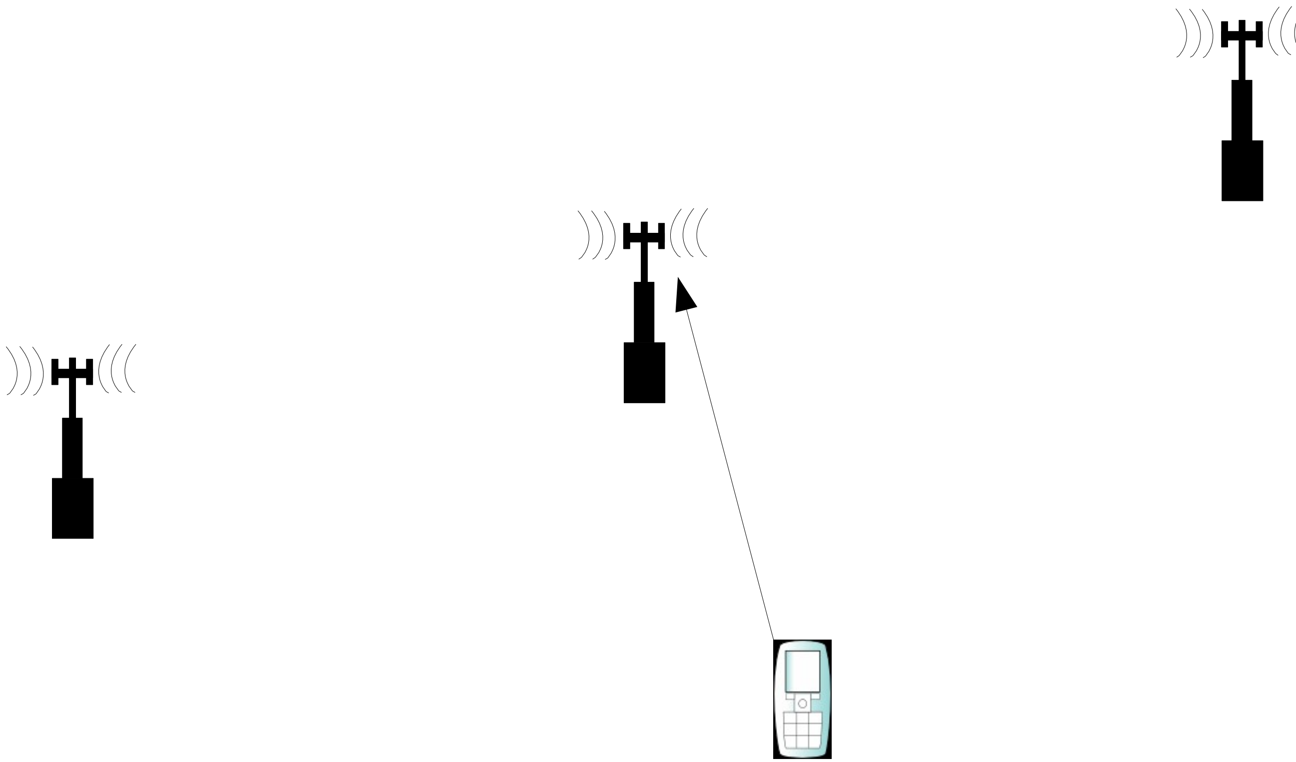
Handover minimization

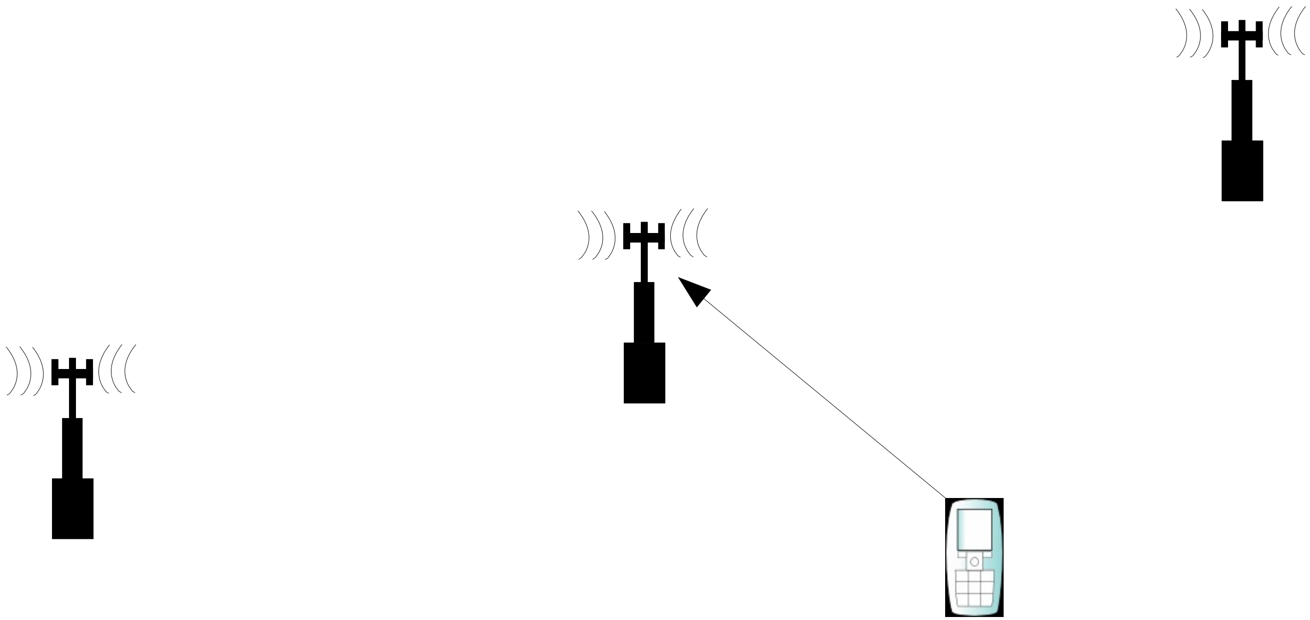




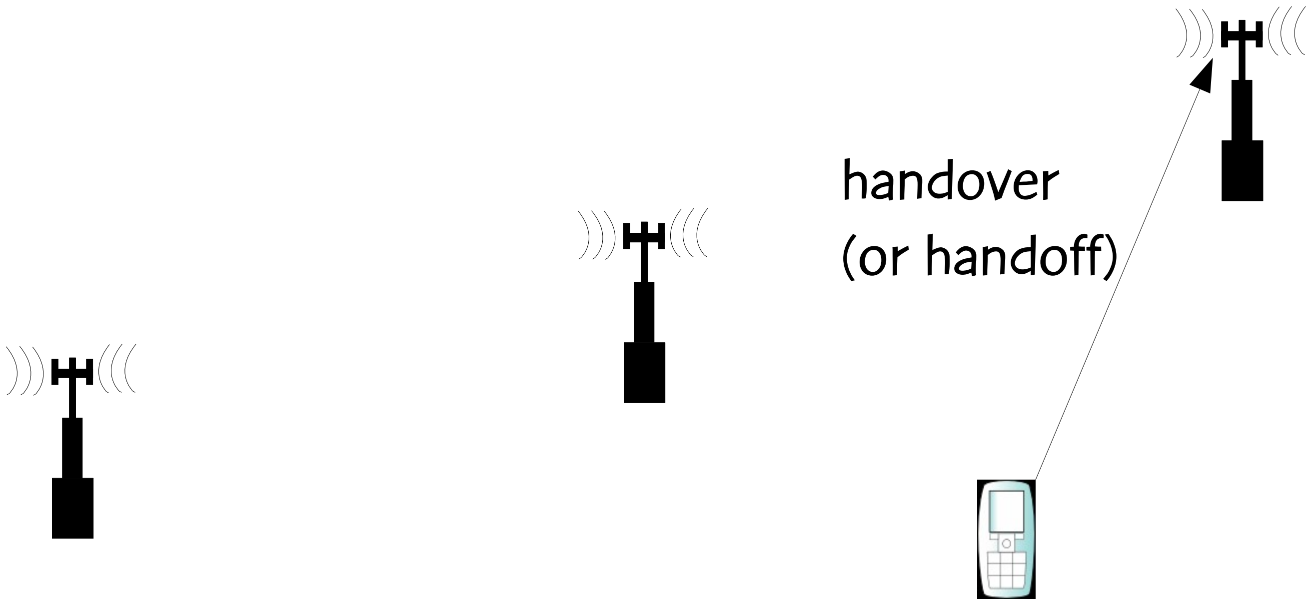


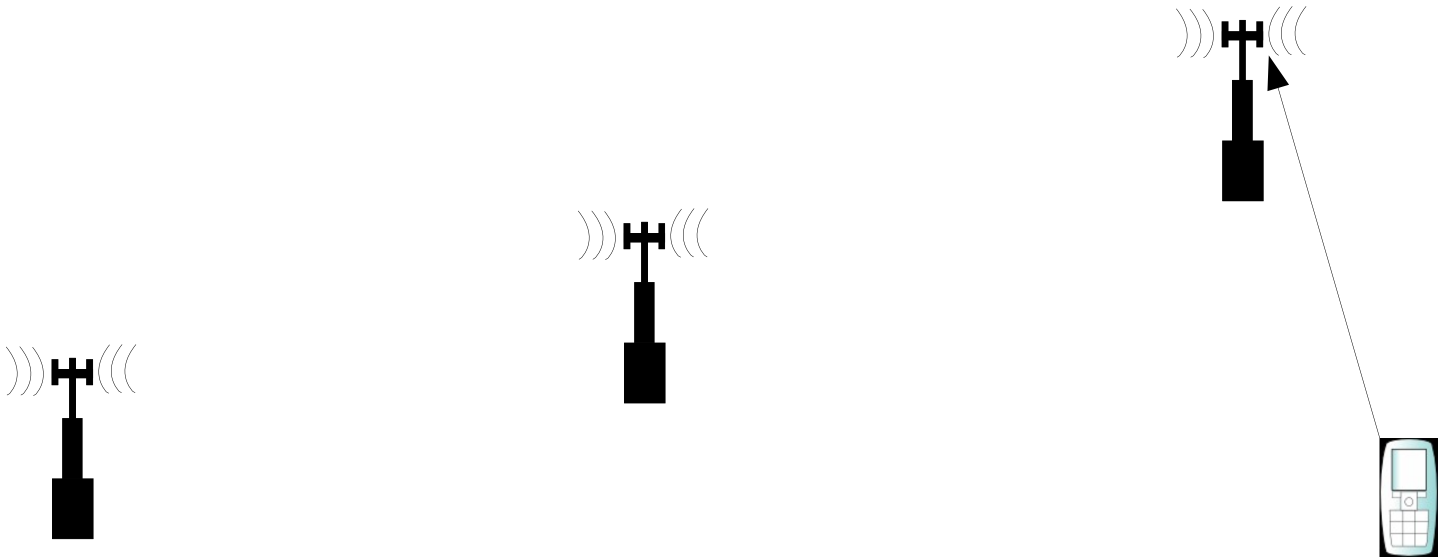

handover

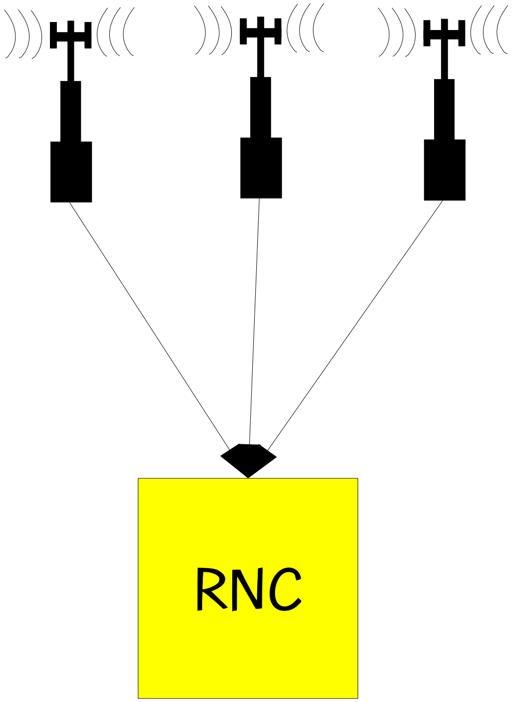




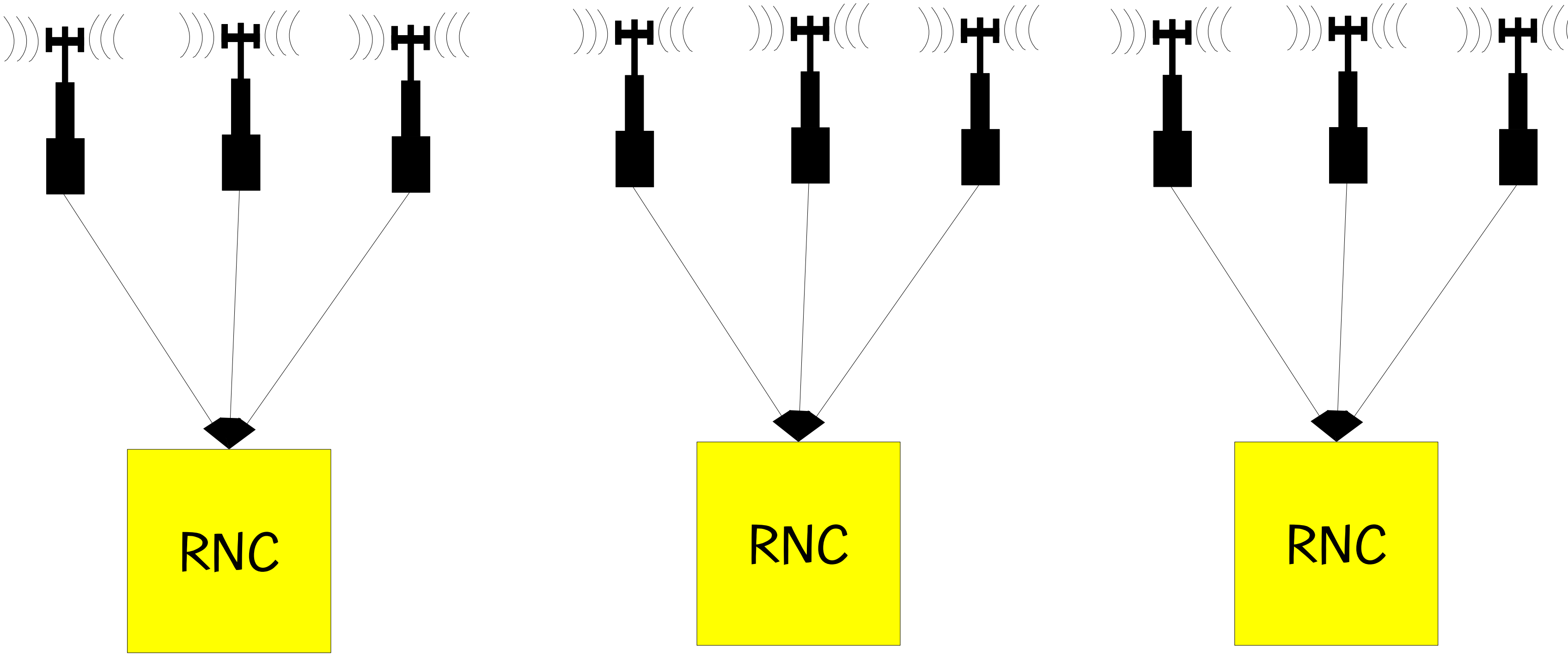
Handover minimization as a GQAP


handover
(or handoff)

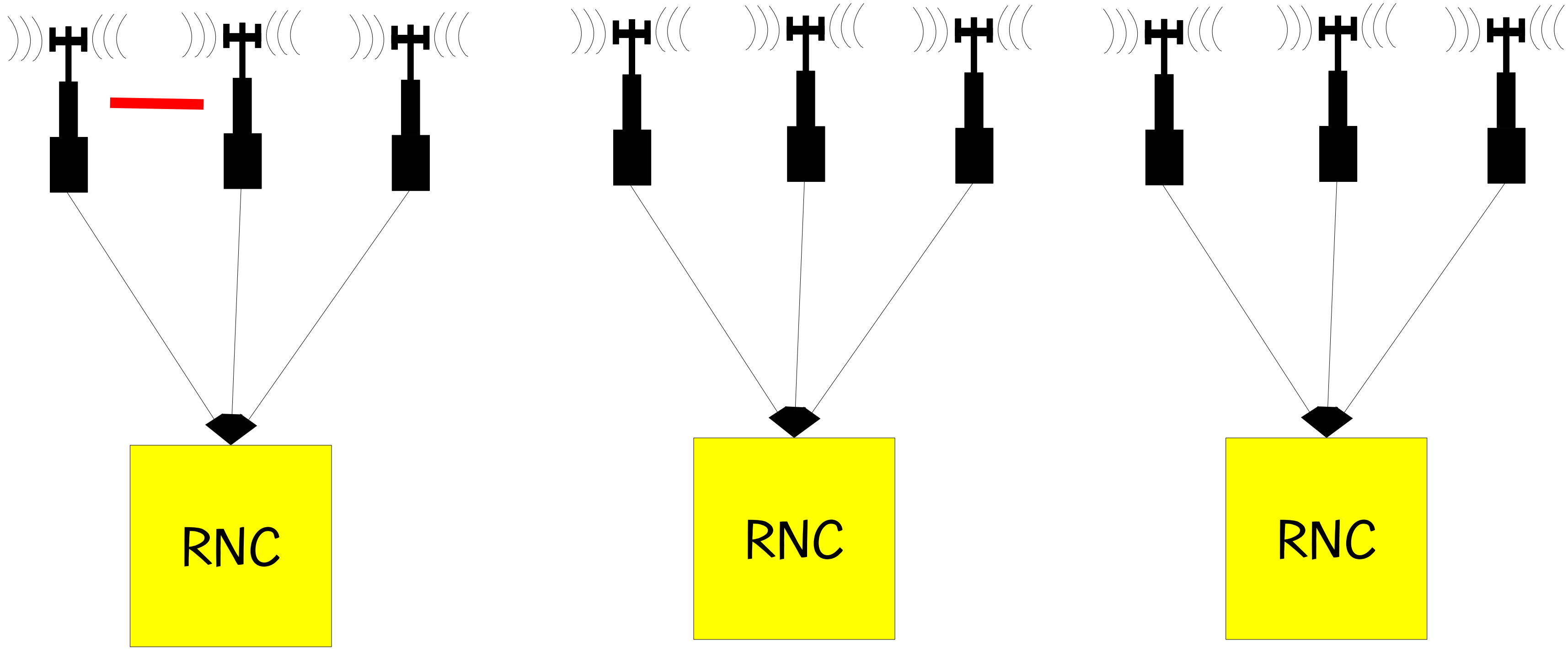




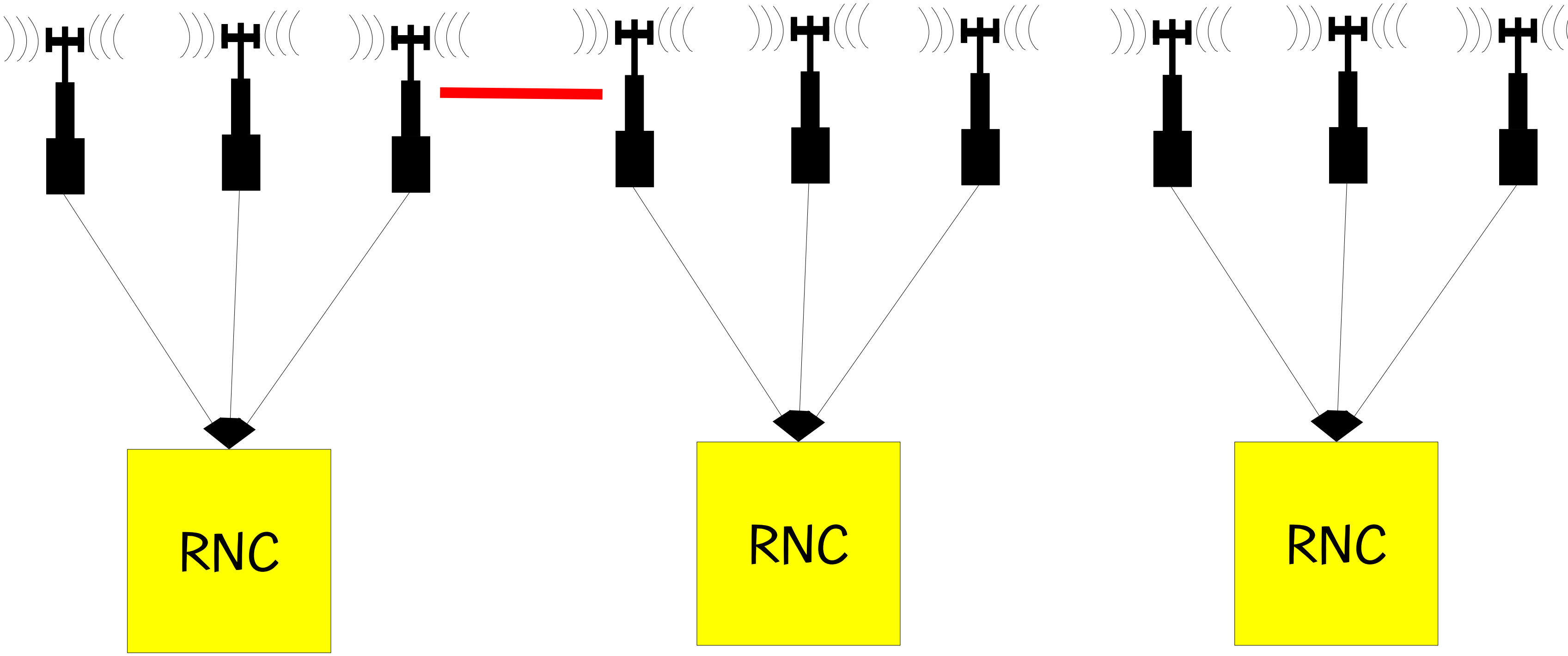
- Each cell tower has associated with it an amount of traffic.
- Each cell tower is connected to a Radio Network Controller (RNC).
- Each RNC can have one or more cell towers connected to it.
- Each RNC can handle a given amount of traffic … this limits the subsets of cell towers that can be connected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



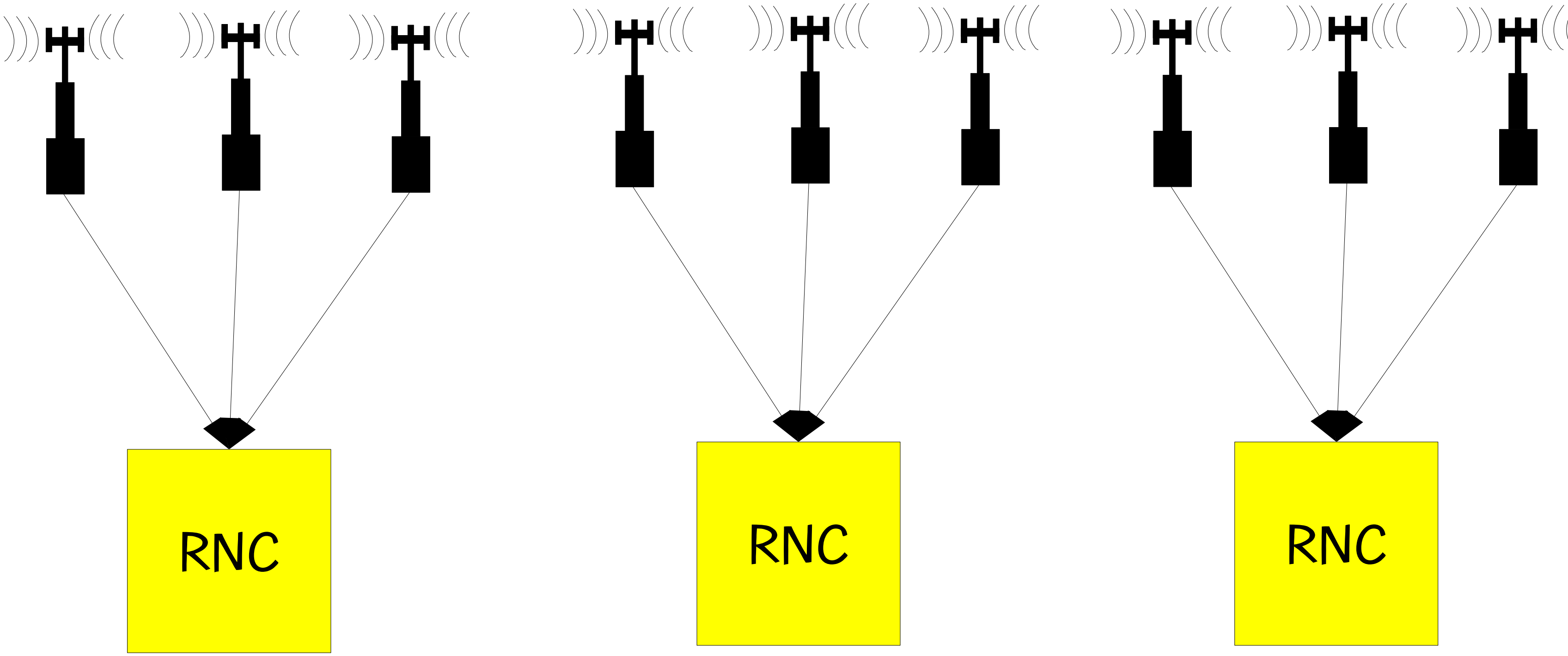
- Handovers can occur between towers



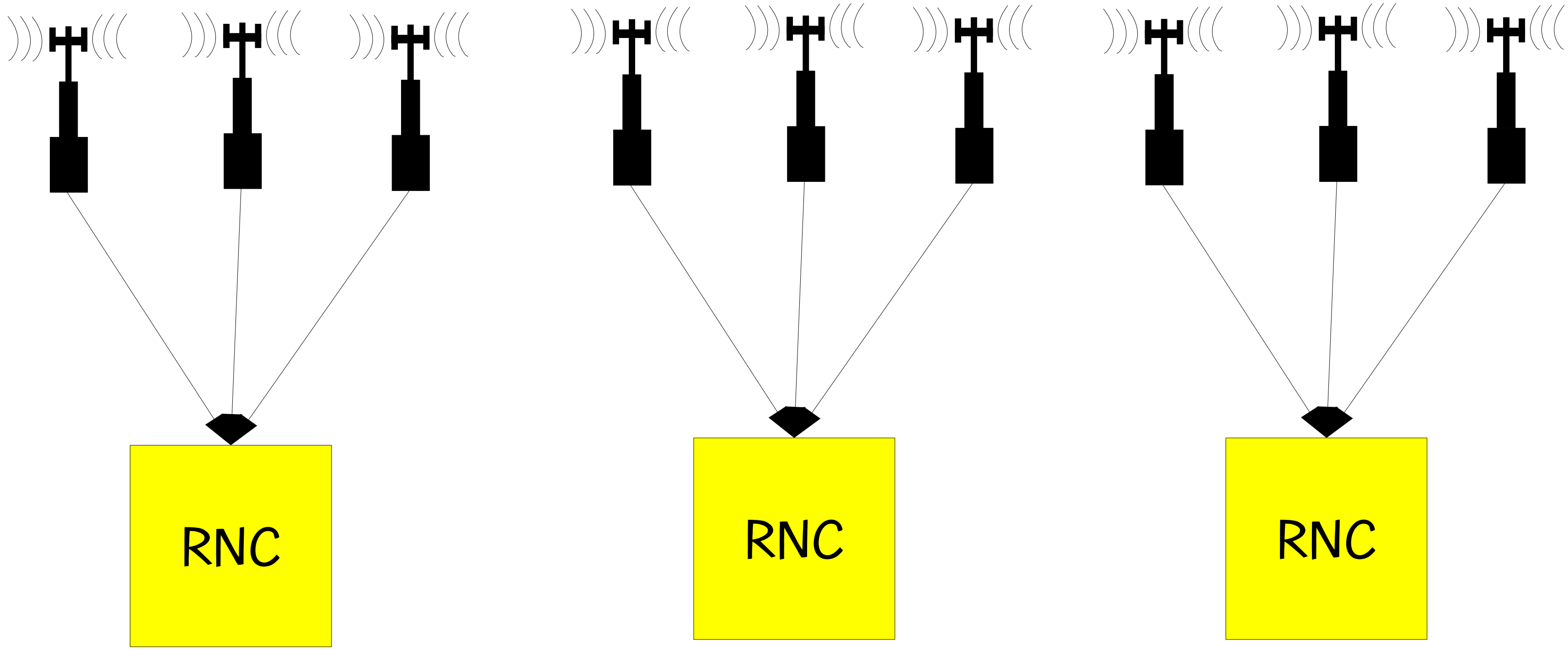
- Handovers can occur between towers
  - connected to the same RNC



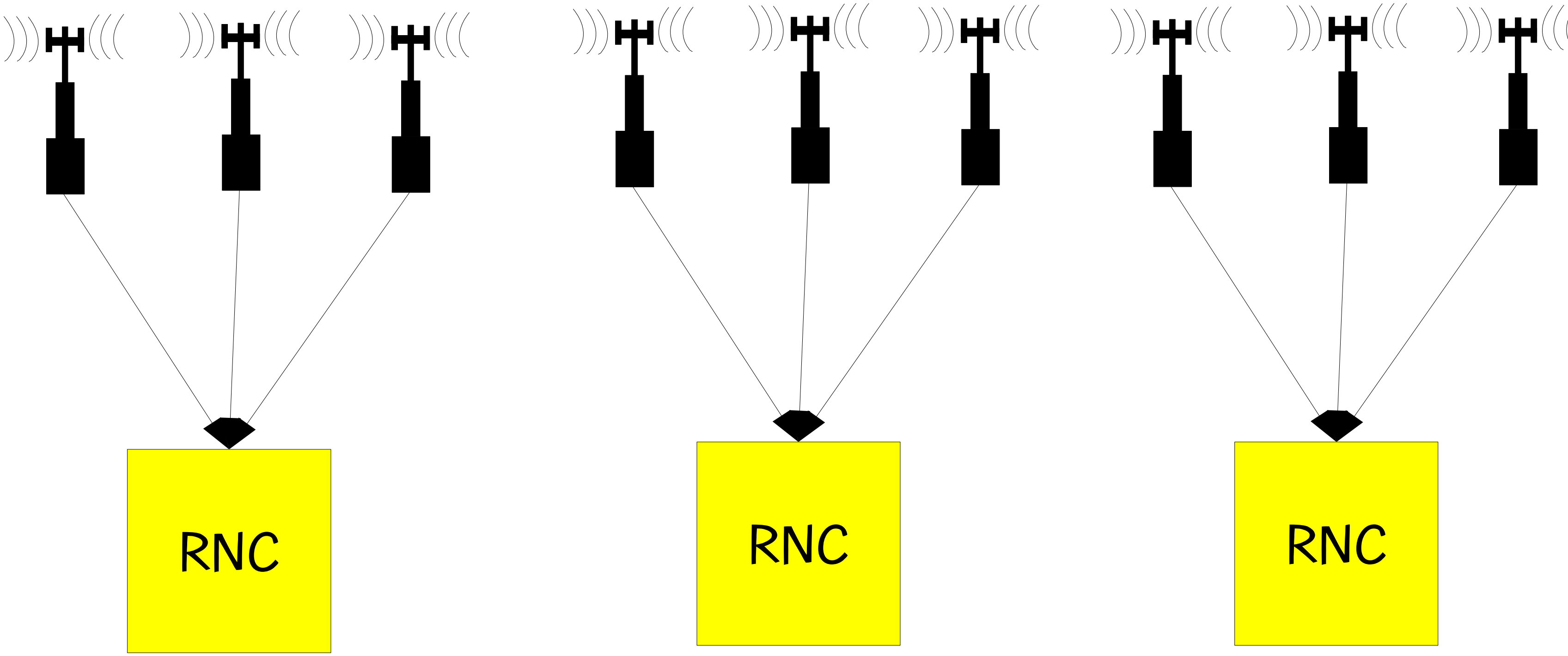
- Handovers can occur between towers
  - connected to the same RNC
  - connected to different RNCs



- 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!

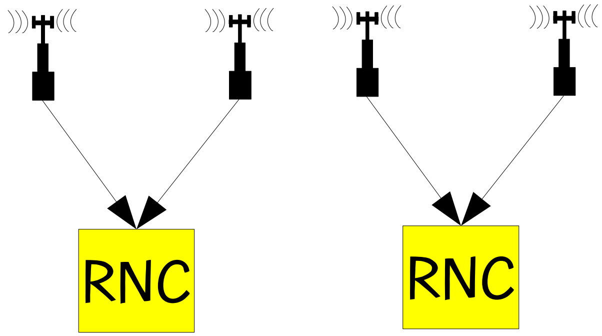


- If we minimize the number of handovers between towers connected to different RNCs we may be able to reduce the number of dropped calls.



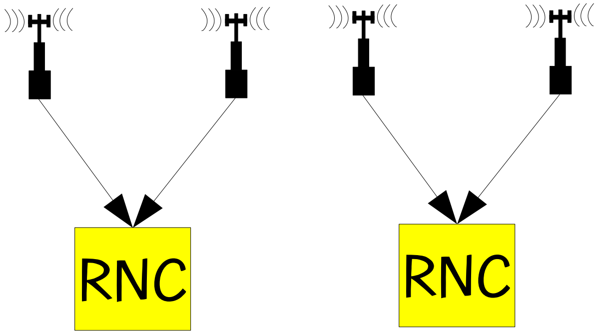
- HANDOVER MINIMIZATION: Assign towers to RNCs such that RNC capacity is not violated and number of handovers between towers assigned to different RNCs is minimized.

Example

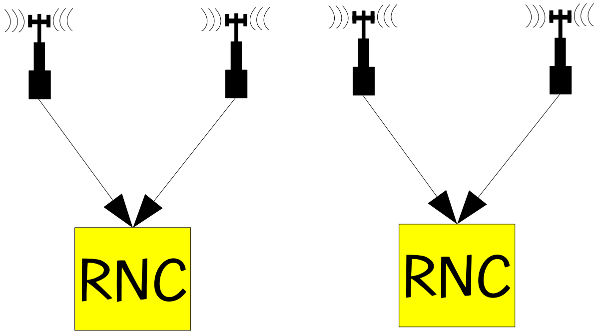


- 4 towers: t(1) = 25; t(2) = 15; t(3) = 35; t(4) = 25
- 2 RNCs: c(1) = 50; c(2) = 60
- Handover matrix:

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

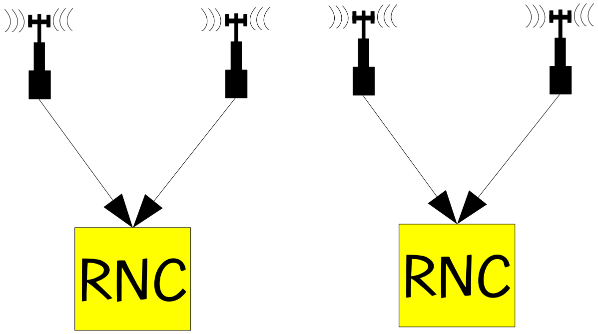


- 4 towers: t(1) = 25; t(2) = 15; t(3) = 35; t(4) = 25
- 2 RNCs: c(1) = 50; c(2) = 60
- Given this traffic profile and RNC capacities the feasible configurations are:
  - RNC(1): { 1, 2 }; RNC(2): { 3, 4 }
  - RNC(1): { 2, 3 }; RNC(2): { 1, 4 }
  - RNC(1): { 2, 4 }; RNC(2): { 1, 3 }
  - RNC(1): { 1,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



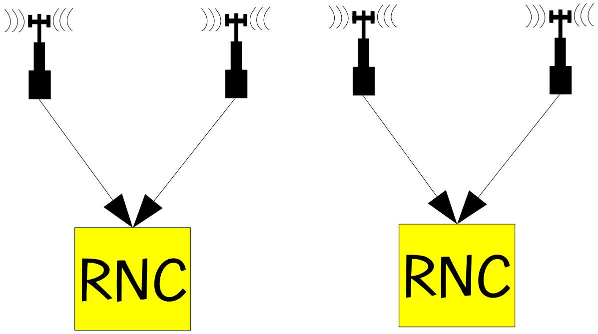
	1	2	3	4
1	0	100	10	0
2	100	0	200	50
3	10	200	0	500
4	0	50	500	0

- Total handover for each configuration:



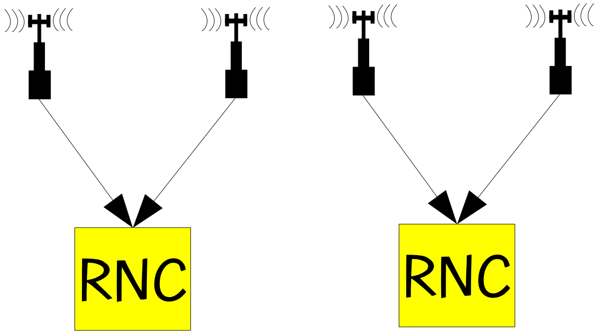
	1	2	3	4
1	0	100	10	0
2	100	0	200	50
3	10	200	0	500
4	0	50	500	0

- 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



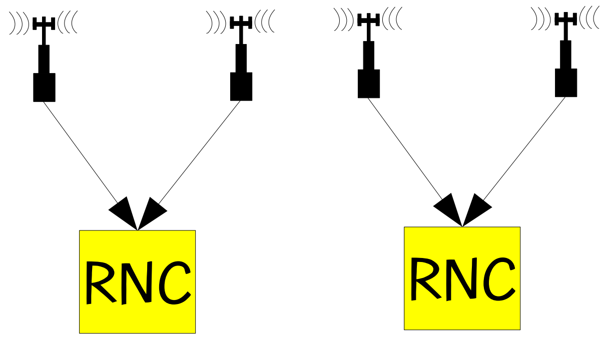
	1	2	3	4
1	0	100	10	0
2	100	0	200	50
3	10	200	0	500
4	0	50	500	0

- 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



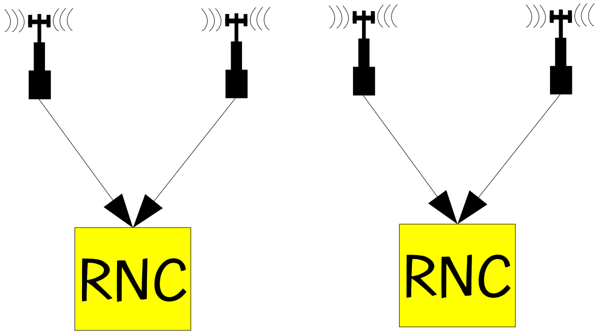
	1	2	3	4
1	0	100	10	0
2	100	0	200	50
3	10	200	0	500
4	0	50	500	0

- 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

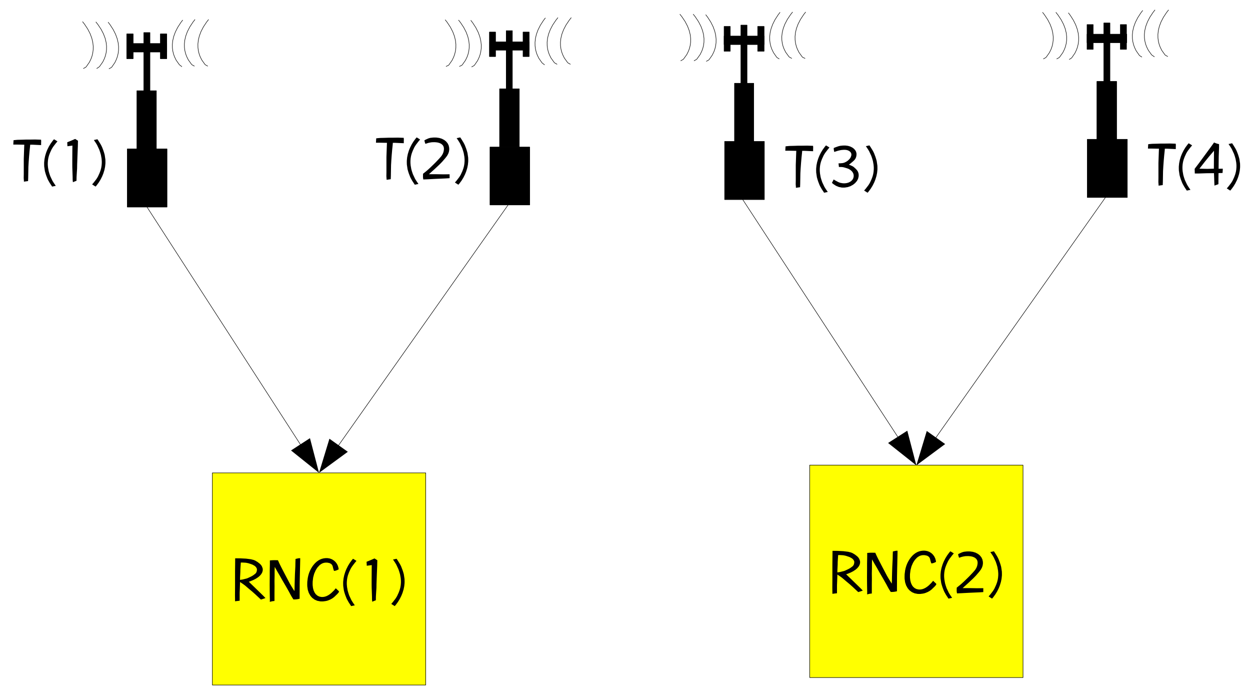


	1	2	3	4
1	0	100	10	0
2	100	0	200	50
3	10	200	0	500
4	0	50	500	0

- 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:

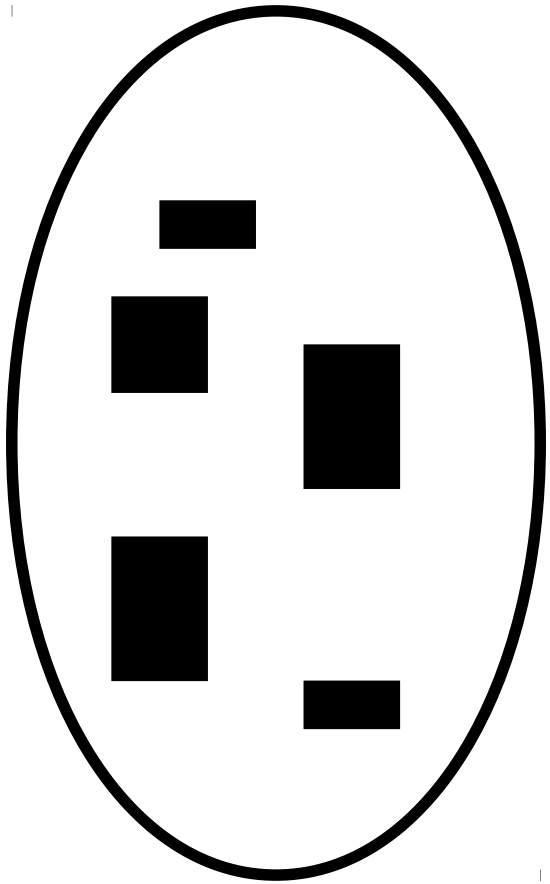


Generalized quadratic assignment problem

Generalized quadratic assignment

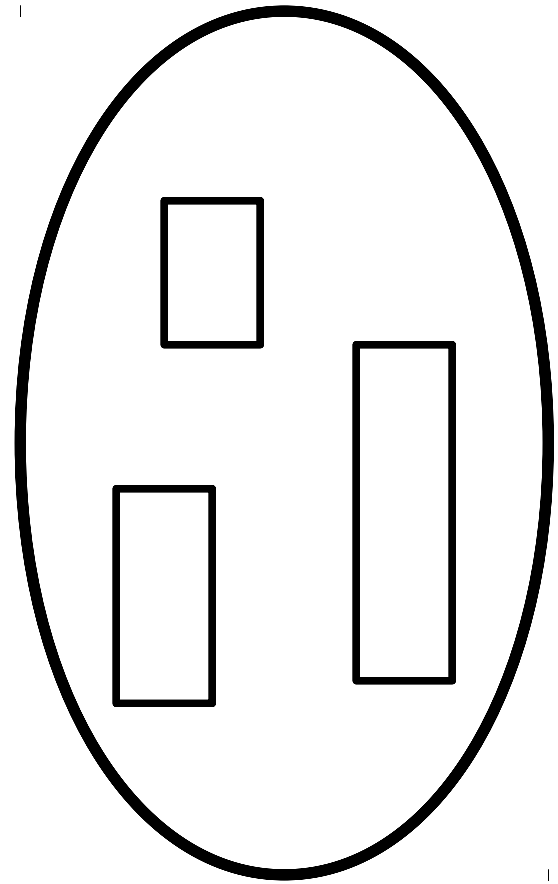
- Generalization of the quadratic assignment problem (QAP).
- Multiple facilities can be assigned to a single location as long as the capacity of the location allows.

N: set of n facilities



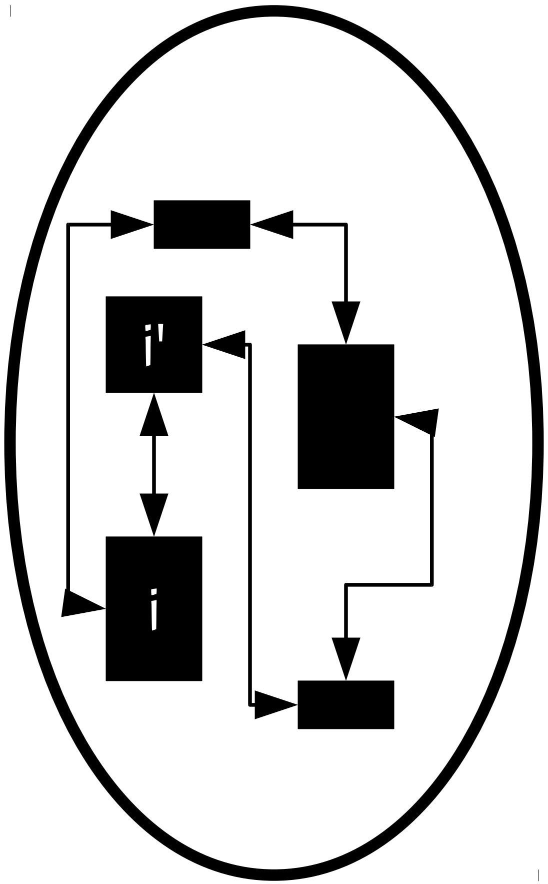
d_i : capacity demanded by facility $i \in N$

M: set of m locations

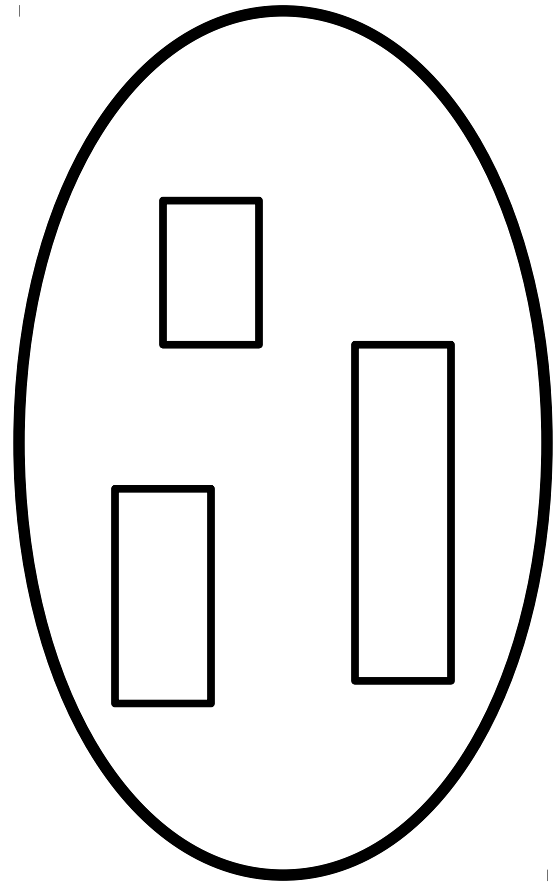


Q_j : capacity of location $j \in M$

N: set of n facilities

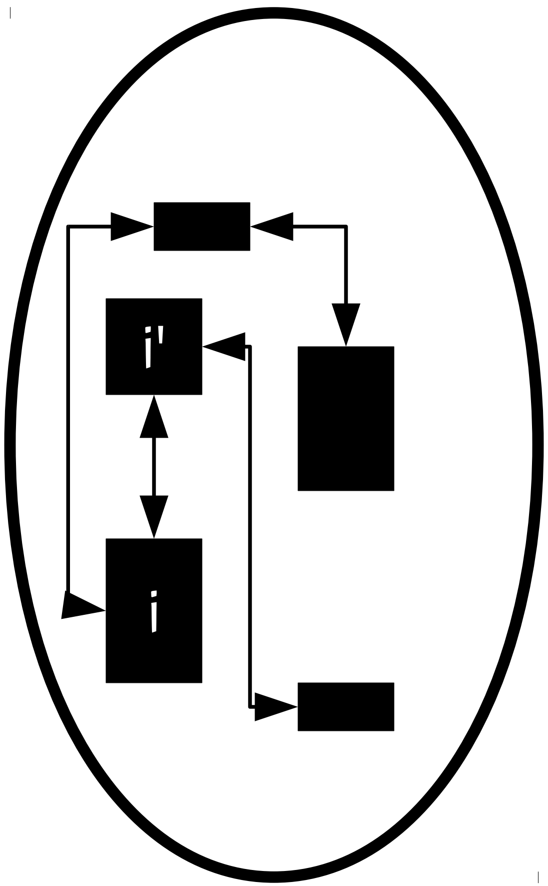


M: set of m locations

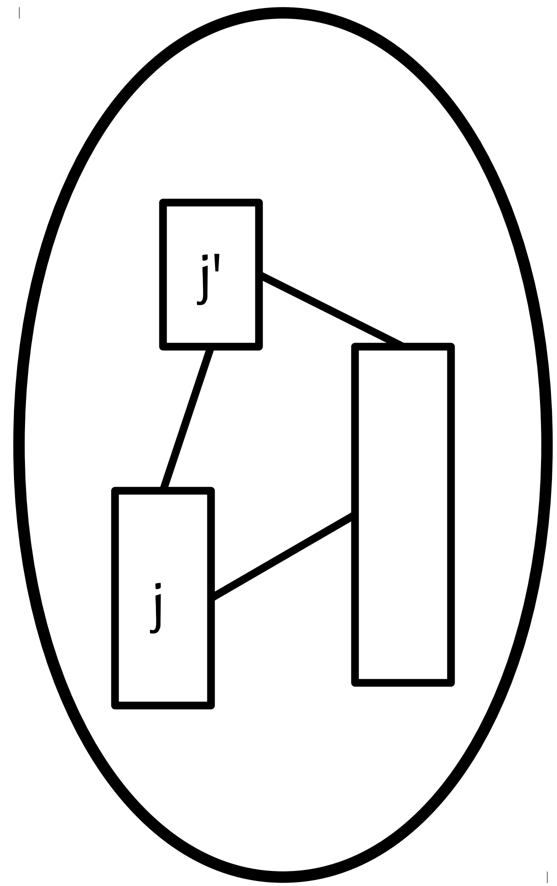


$A_{nxn} = (a_{ii'})$: flow between facilities

N: set of n facilities



M: set of m locations

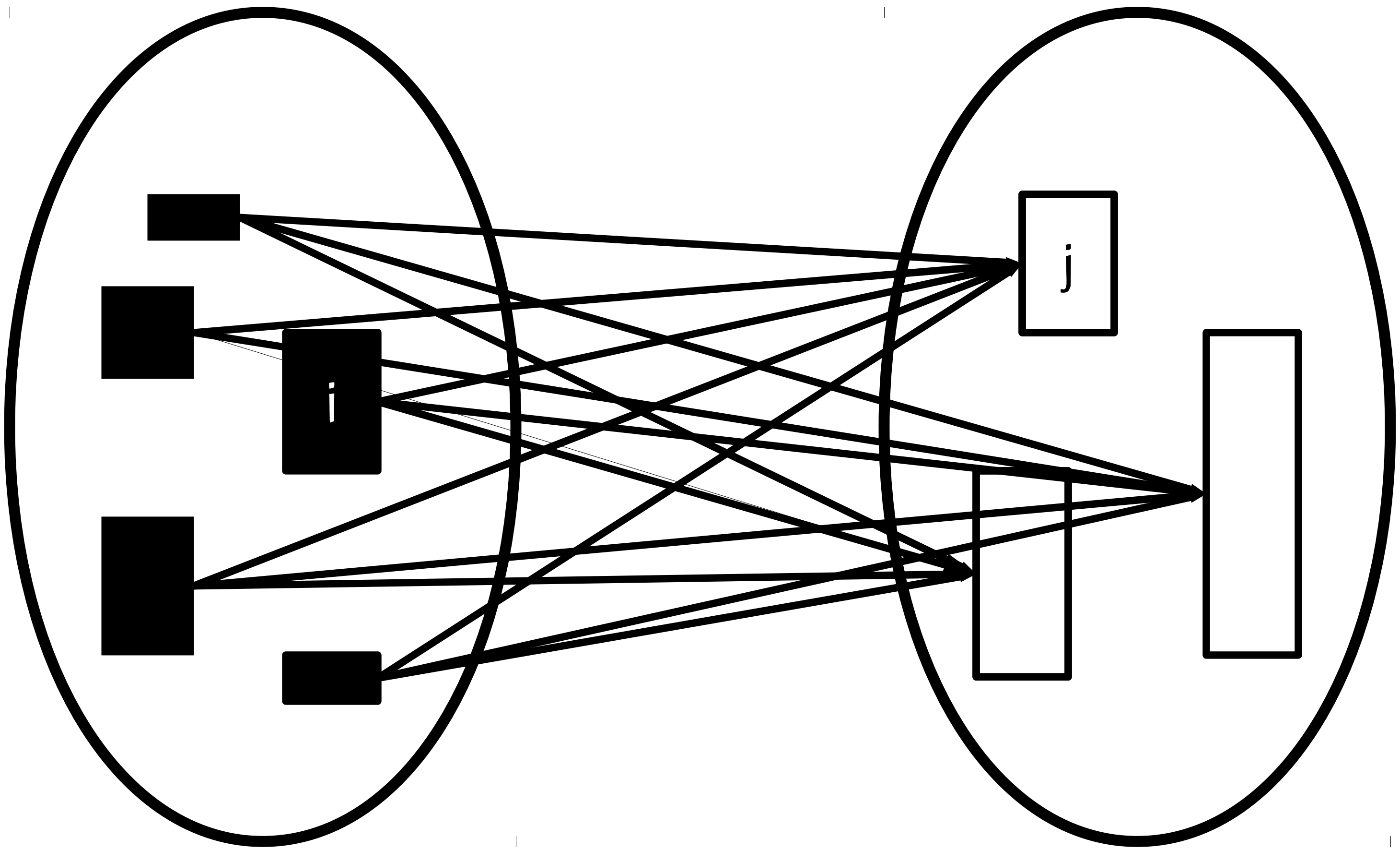


$A_{nxn} = (a_{ii'})$: flow between facilities

$B_{mxm} = (b_{jj'})$: distance between locations

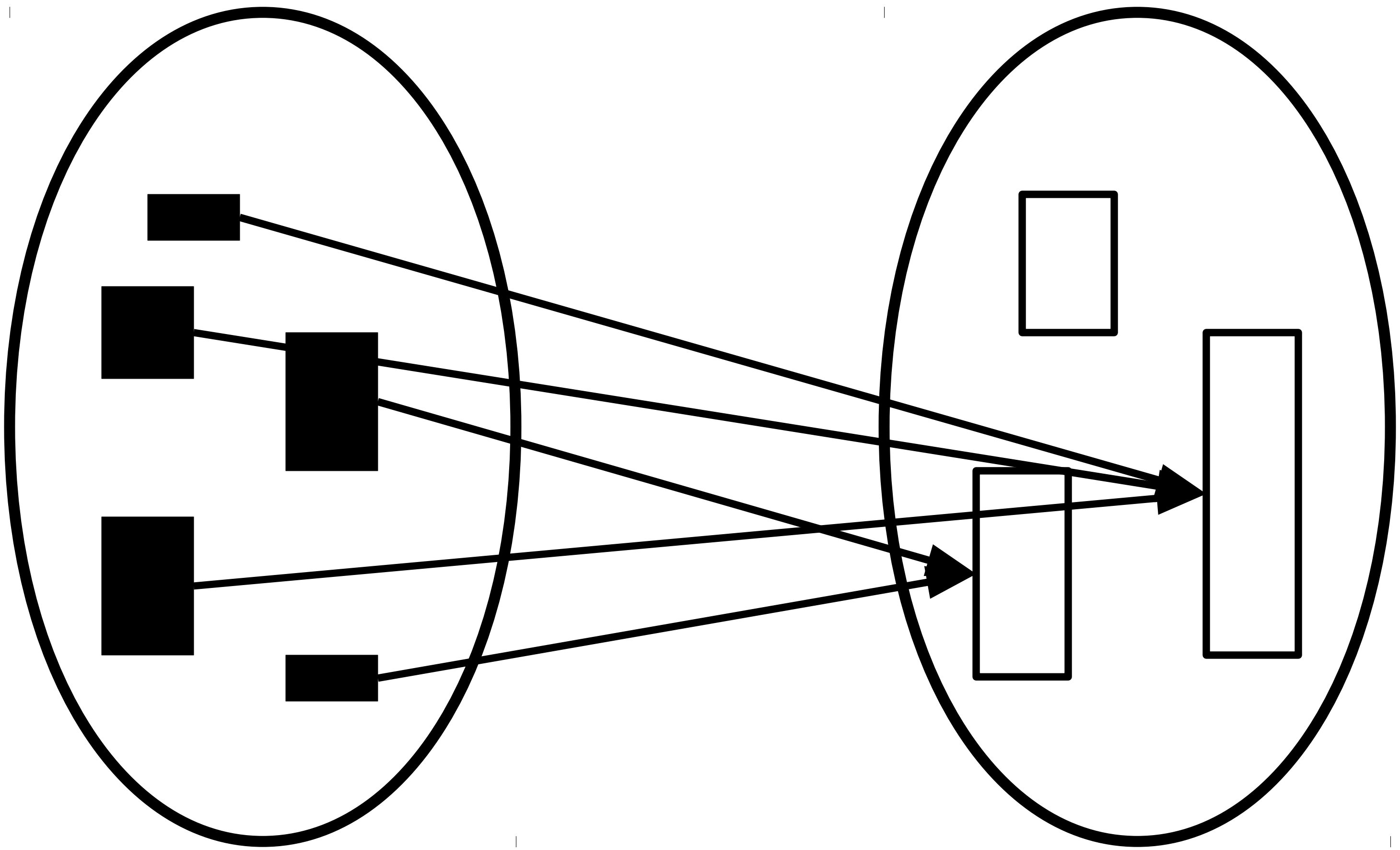
N: set of n facilties

M: set of m locations



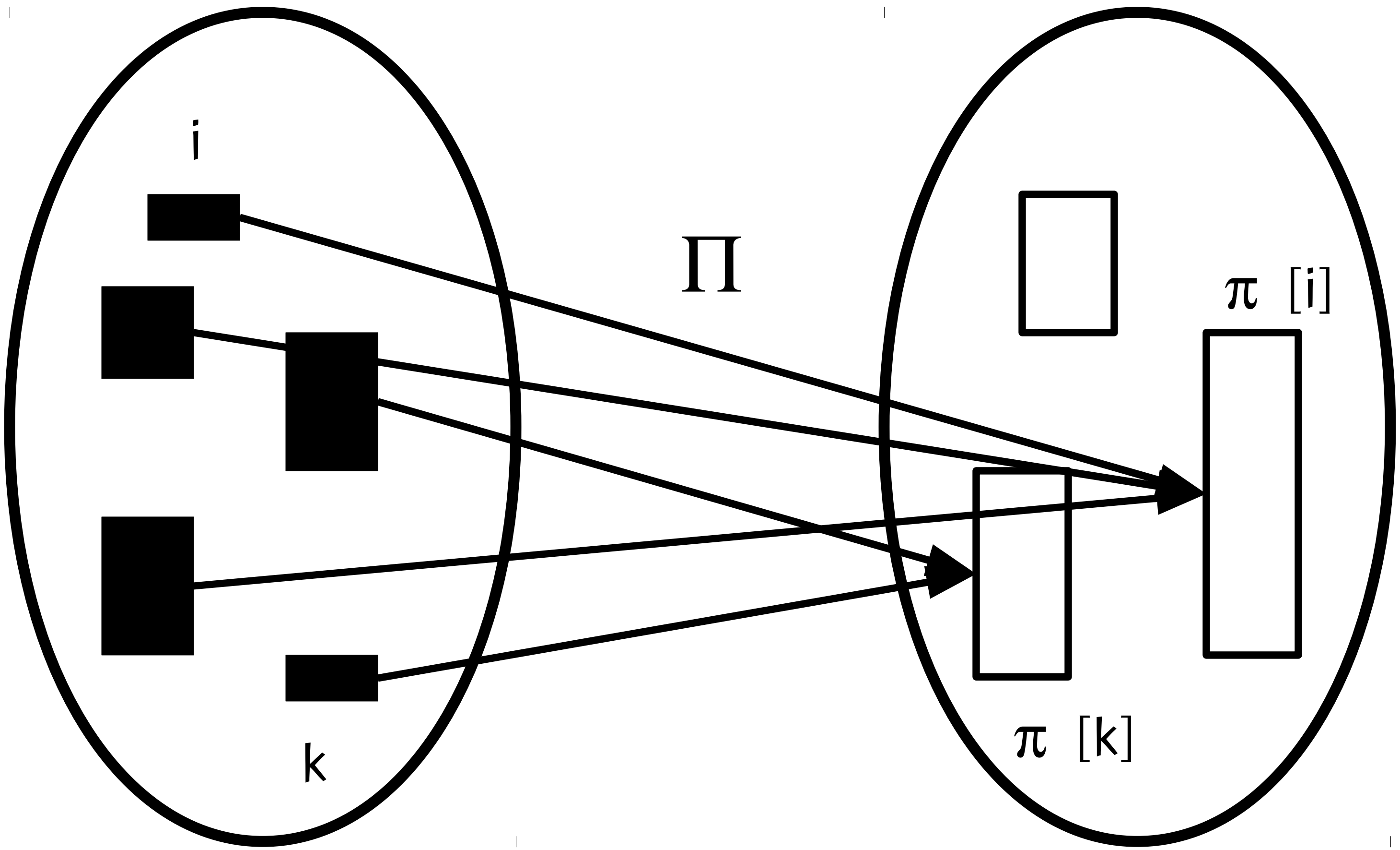
$C_{nxm}=(c_{ij})$: cost of assigning facility $i \in N$ to location $j \in M$

The generalized quadratic assignment problem



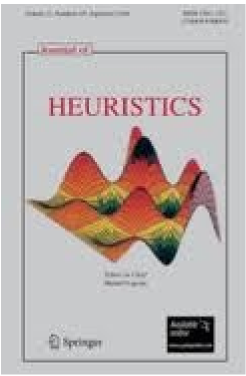
GQAP seeks a assignment, without violating the capacities of locations, that minimizes the sum of products of flows and distances in addition to a linear total cost of assignment.

The generalized quadratic assignment problem



$$\text{cost}[\Pi] = \text{sum}(i=1,n)\ \ c[i,\pi[i]] + \text{sum}(i=1,n)\ \text{sum}\ (i\neq k=1,n)\ \ F[i,k]*D[\pi[i],\pi[k]]$$

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

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

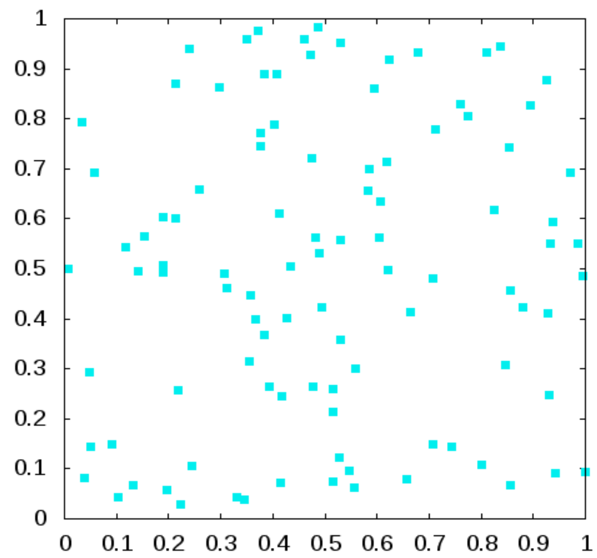
Experiments

Random instance generator

- Input T (number of towers), R (number of RNCs), r (max handover distance), and lower and upper bounds on traffic, handover, and capacity slack.

Random instance generator

- Input T (number of towers), R (number of RNCs), r (max handover distance), and lower and upper bounds on traffic, handover, and capacity slack.
- Generate T random points (towers) on the unit square.



Random instance generator

- Input T (number of towers), R (number of RNCs), r (max handover distance), and lower and upper bounds on traffic, handover, and capacity slack.
- Generate T random points (towers) on the unit square.
- For each tower i, traffic[i] = randunif(l(t), u(t))

Random instance generator

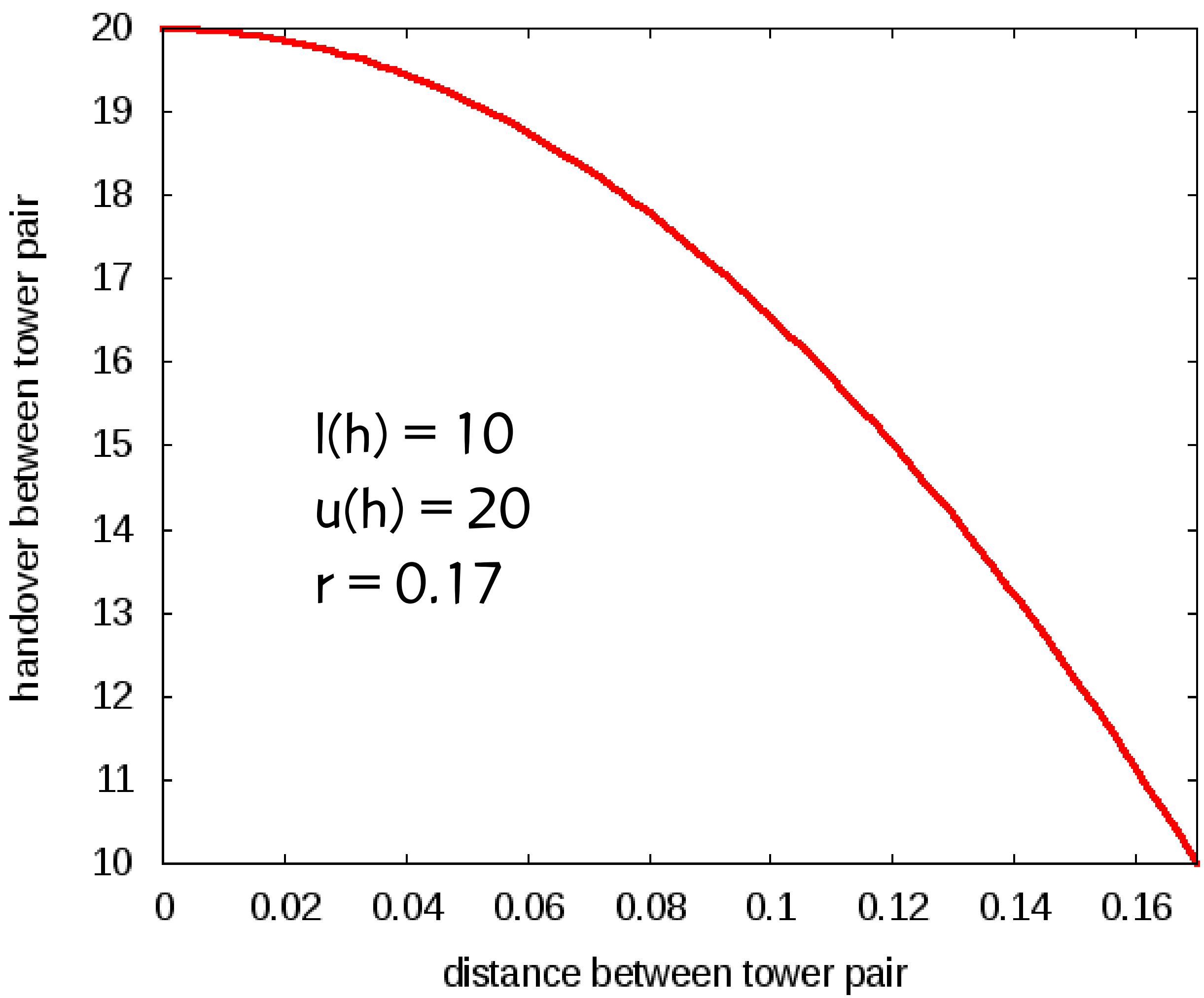
- Input T (number of towers), R (number of RNCs), r (max handover distance), and lower and upper bounds on traffic, handover, and capacity slack.
- Generate T random points (towers) on the unit square.
- For each tower i, traffic[i] = randunif(l(t), u(t))
- avg-traffic is sum of traffic[i]/T over all towers

Random instance generator

- Input T (number of towers), R (number of RNCs), r (max handover distance), and lower and upper bounds on traffic, handover, and capacity slack.
- Generate T random points (towers) on the unit square.
- For each tower i, traffic[i] = randunif(l(t), u(t))
- avg-traffic is sum of traffic[i]/T over all towers
- For each pair of towers {i, j}, if dist(i,j) < r, then $\text{handover}[i,j] = [l(h) - u(h)]/r^2 \times d^2 + u(h)$

For each pair of towers $\{i, j\}$, if $dist(i,j) < r$, then

$$handover[i,j] = [l(h) - u(h)]/r^2 \times d^2 + u(h)$$



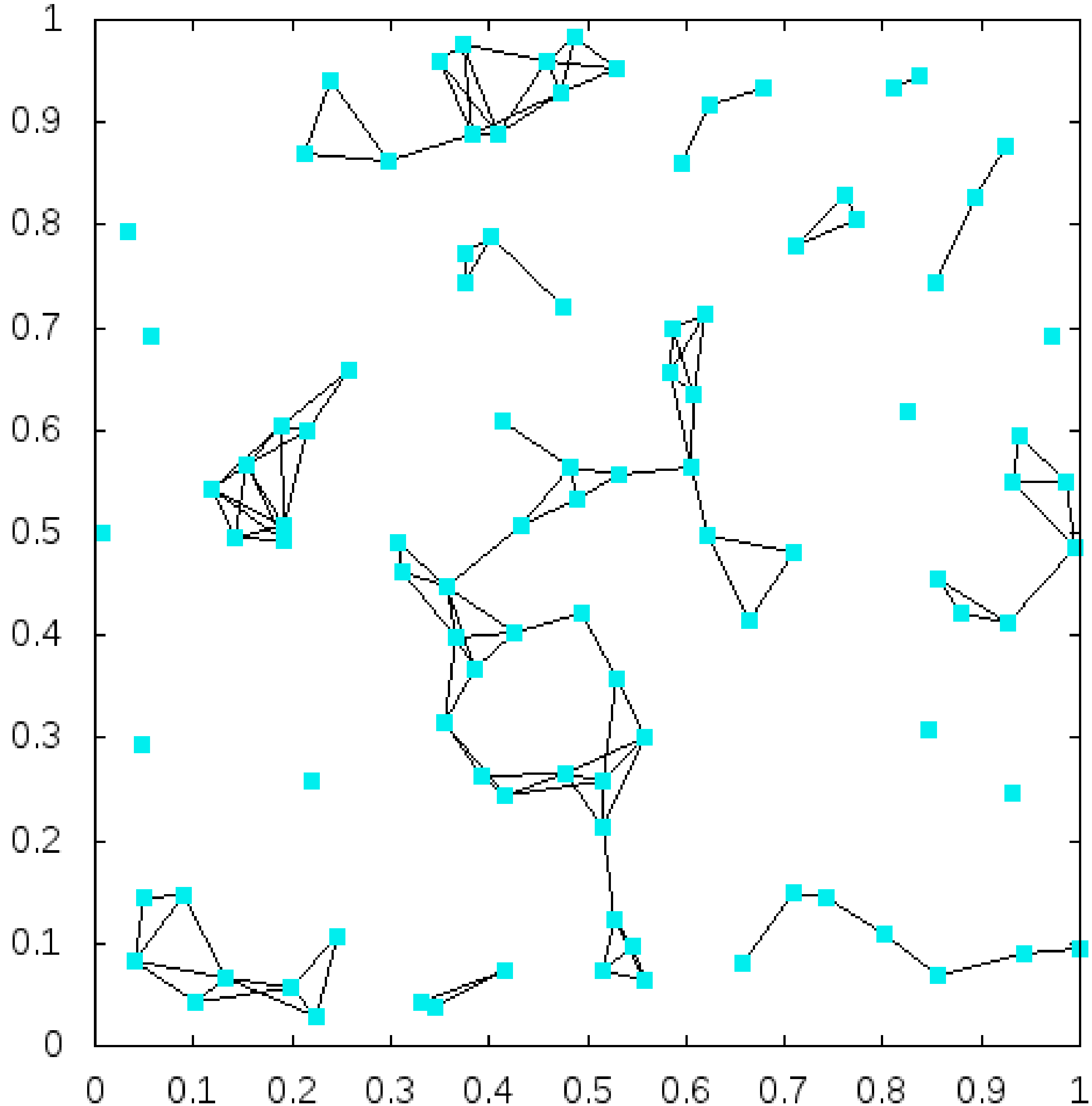
Random instance generator

- Input T (number of towers), R (number of RNCs), r (max handover distance), and lower and upper bounds on traffic, handover, and capacity slack.
- Generate T random points (towers) on the unit square.
- For each tower i, traffic[i] = randunif(l(t), u(t))
- avg-traffic is sum of traffic[i]/T over all towers
- For each pair of towers {i, j}, if dist(i,j) < r, then handover[i,j] = $[l(h) - u(h)]/r^2 \times d^2 + u(h)$
- For each RNC j, capacity[j] = randunif(l(c), u(c)) * avg-traffic, where u(c) > l(c) > 1.

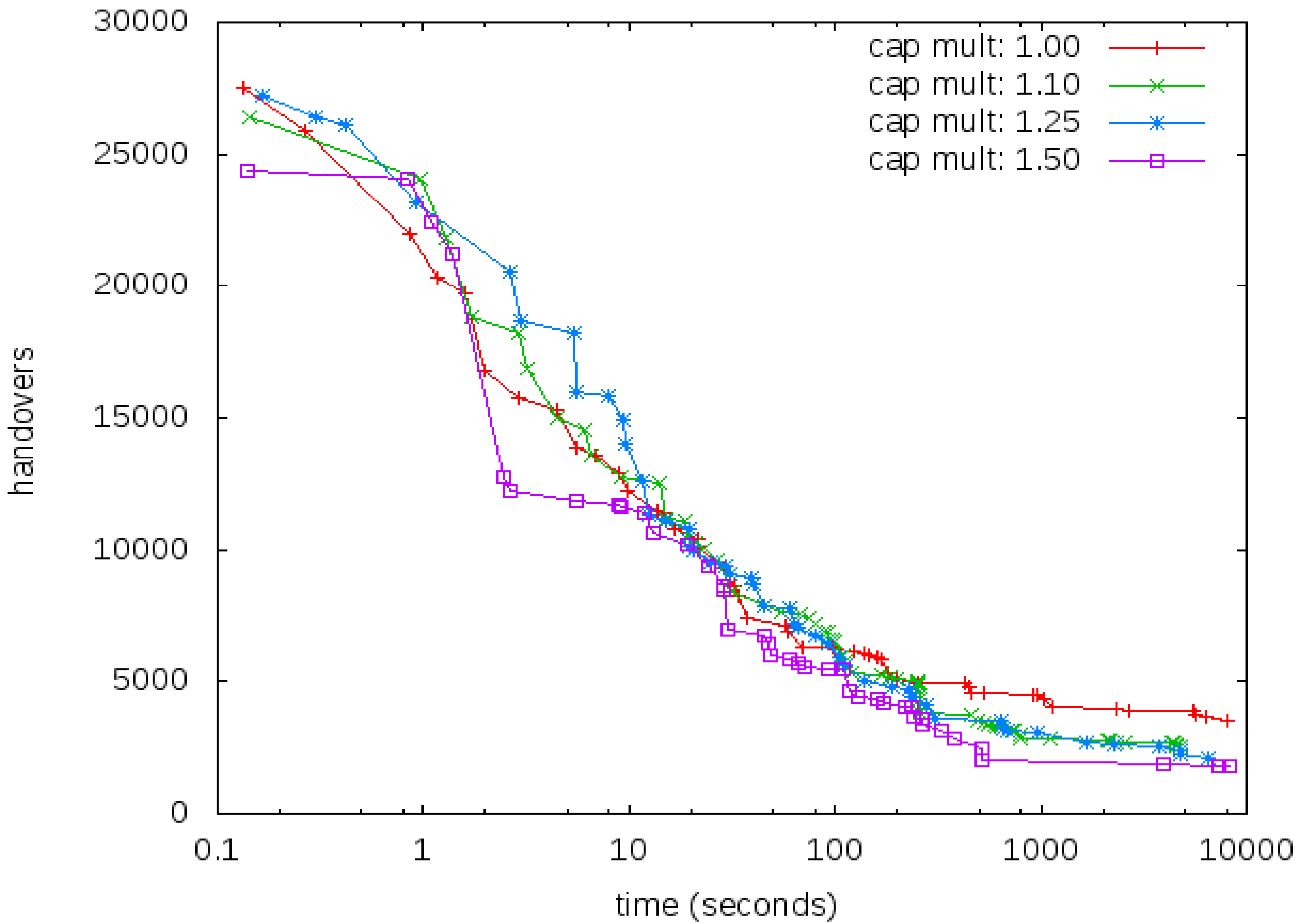
Three synthetic instances

- Number of towers: 100
- Number of RNCs: two instances with 15 and one with 15, 17, 19, 21, 23, 25, 27, and 29
- Tower traffic bounds: [5, 50]
- Handover bounds: [5, 200]
- RNC capacity slack bounds: [1.05, 1.15]
- Three values of max handover distance: 0.1, 0.17, and 0.25

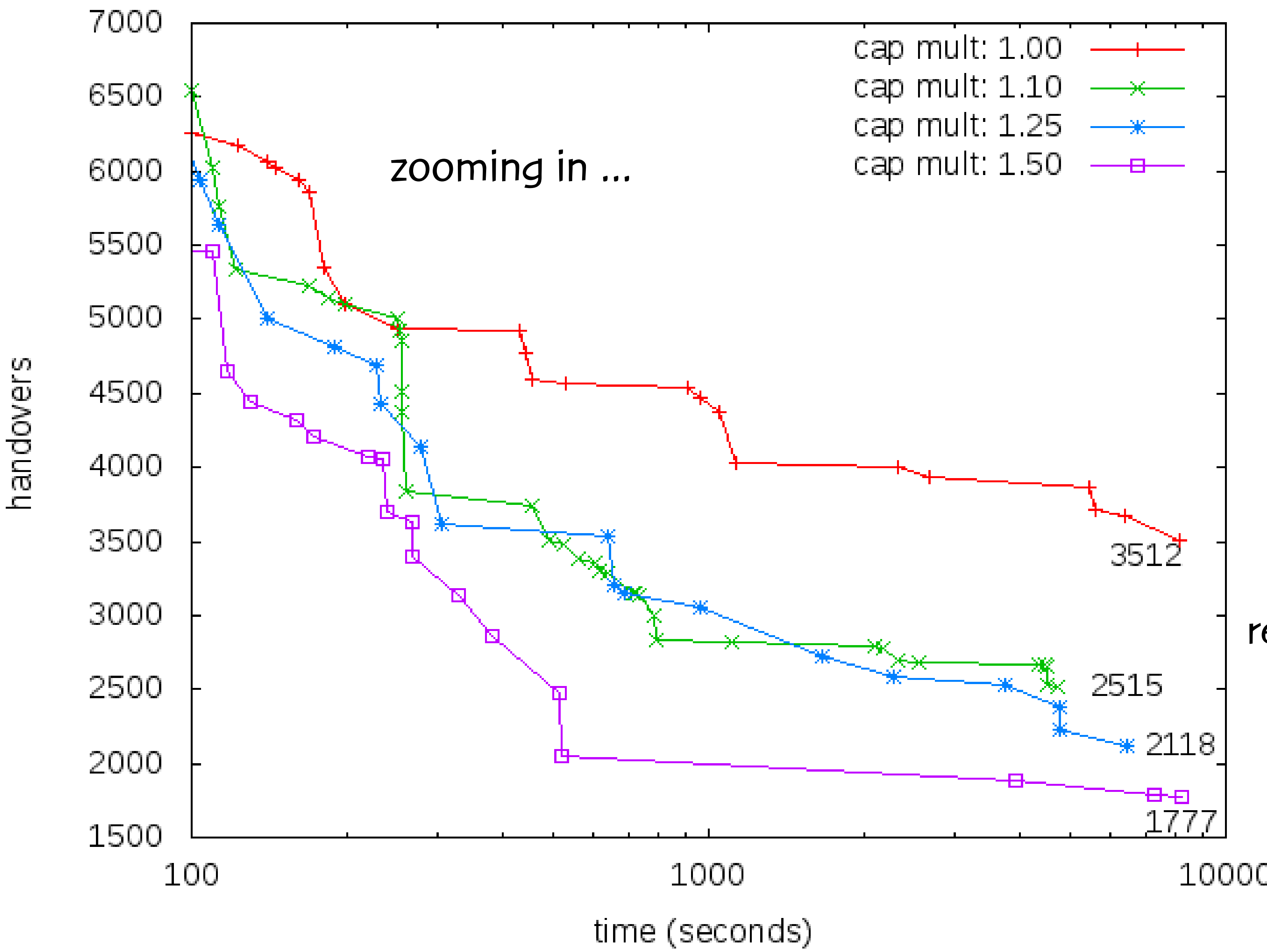
15 RNCs
100 towers
r = 0.1



at&t
Your world. Delivered.



cap mult: 1.00
cap mult: 1.10
cap mult: 1.25
cap mult: 1.50
handovers
time (seconds)
30000
25000
20000
15000
10000
5000
0
0.1
1
10
100
1000
10000



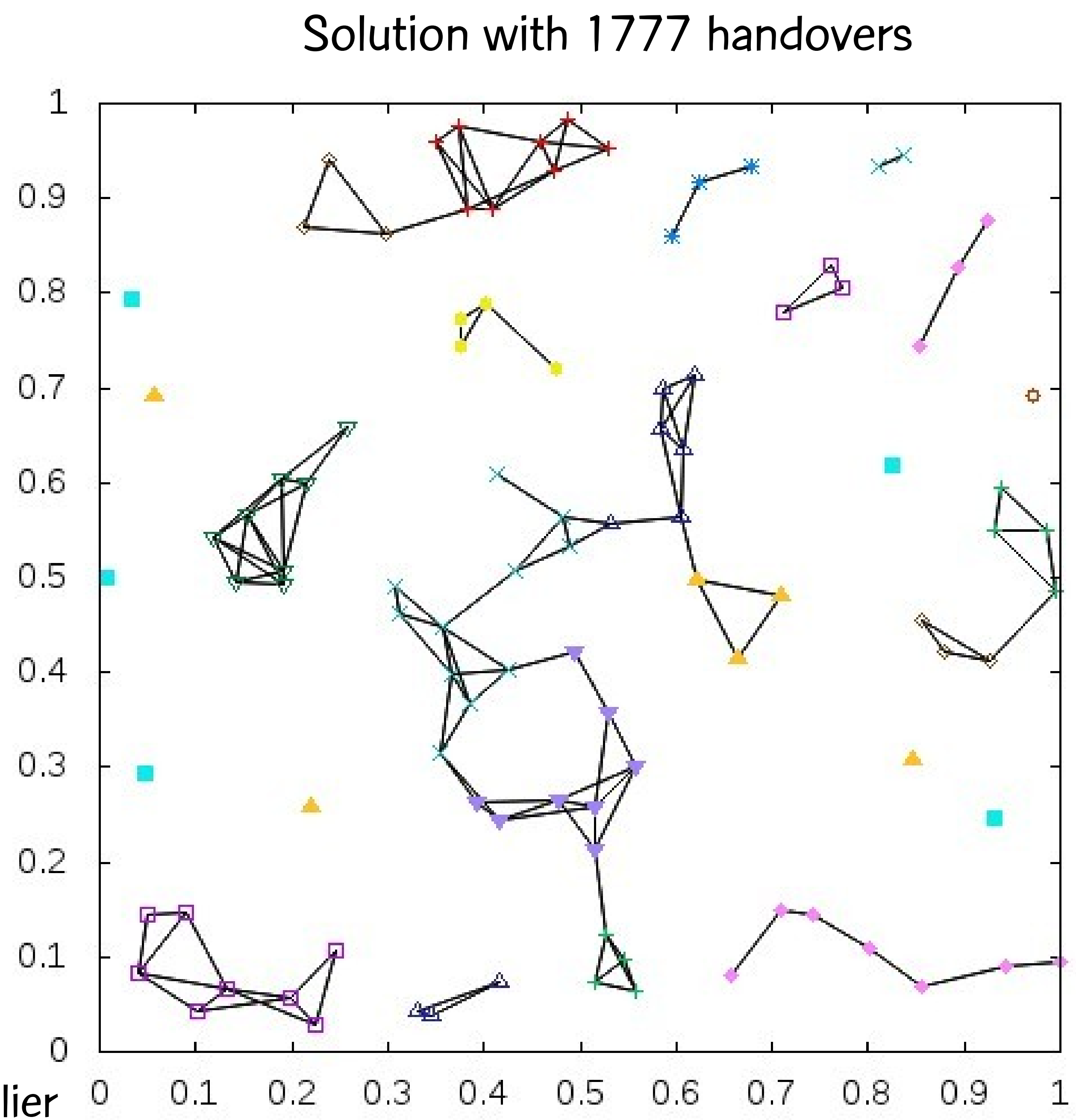
reductions

28%

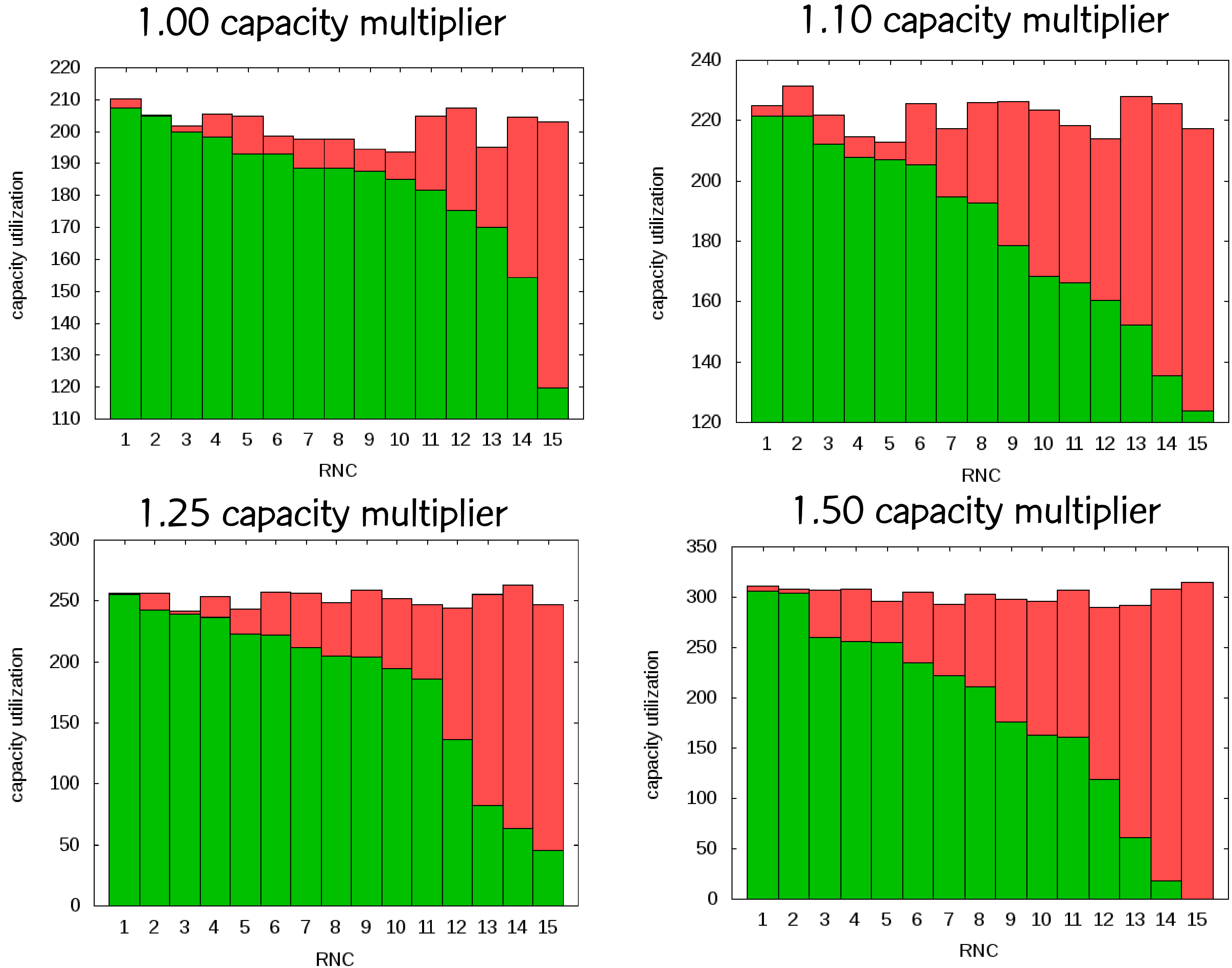
40%

49%

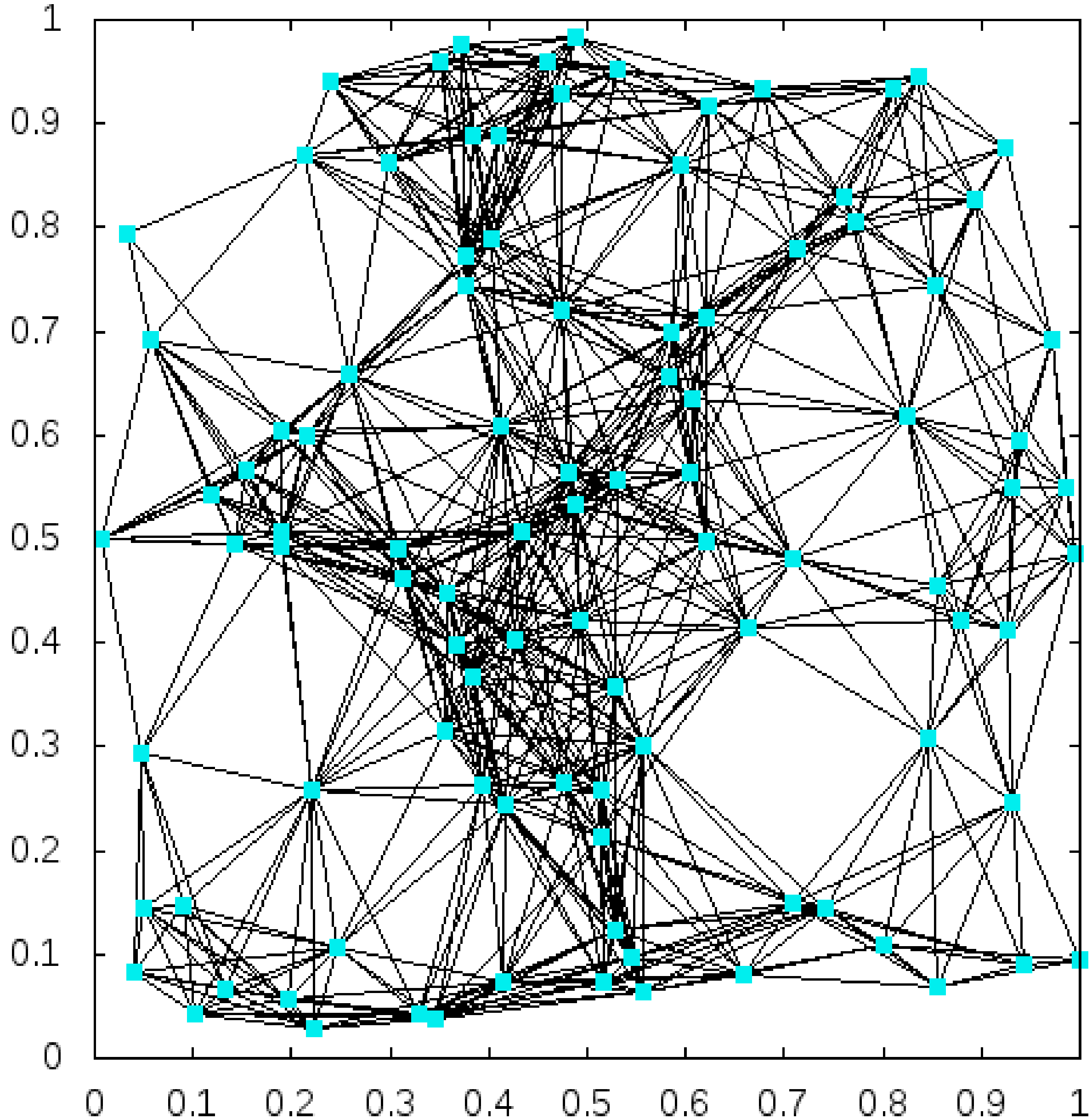
15 RNCs
100 towers
r = 0.1

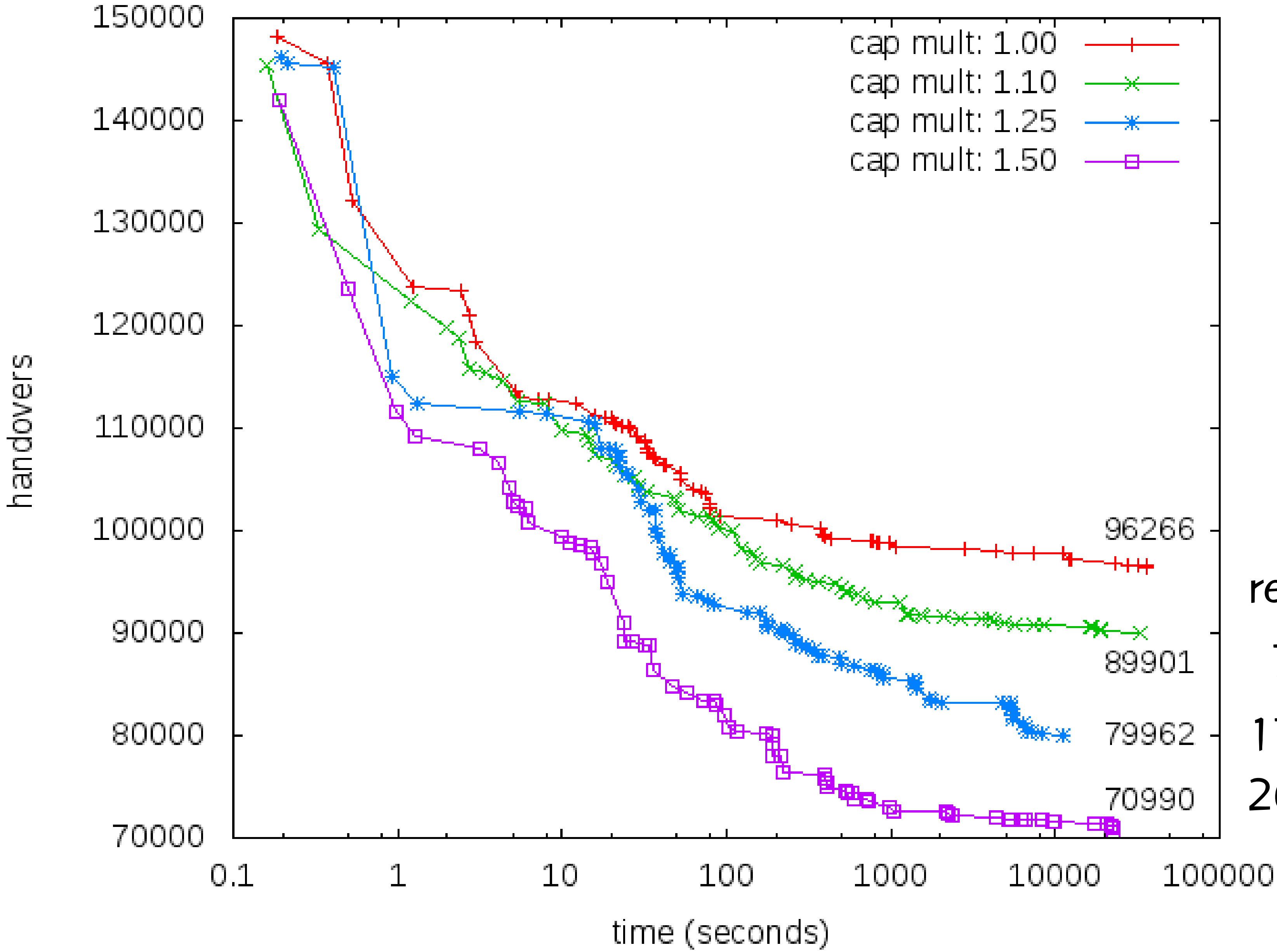


1.50 capacity multiplier


1.00 capacity multiplier
1.10 capacity multiplier
1.25 capacity multiplier
1.50 capacity multiplier
capacity utilization
RNC

15 RNCs
100 towers
r = 0.25



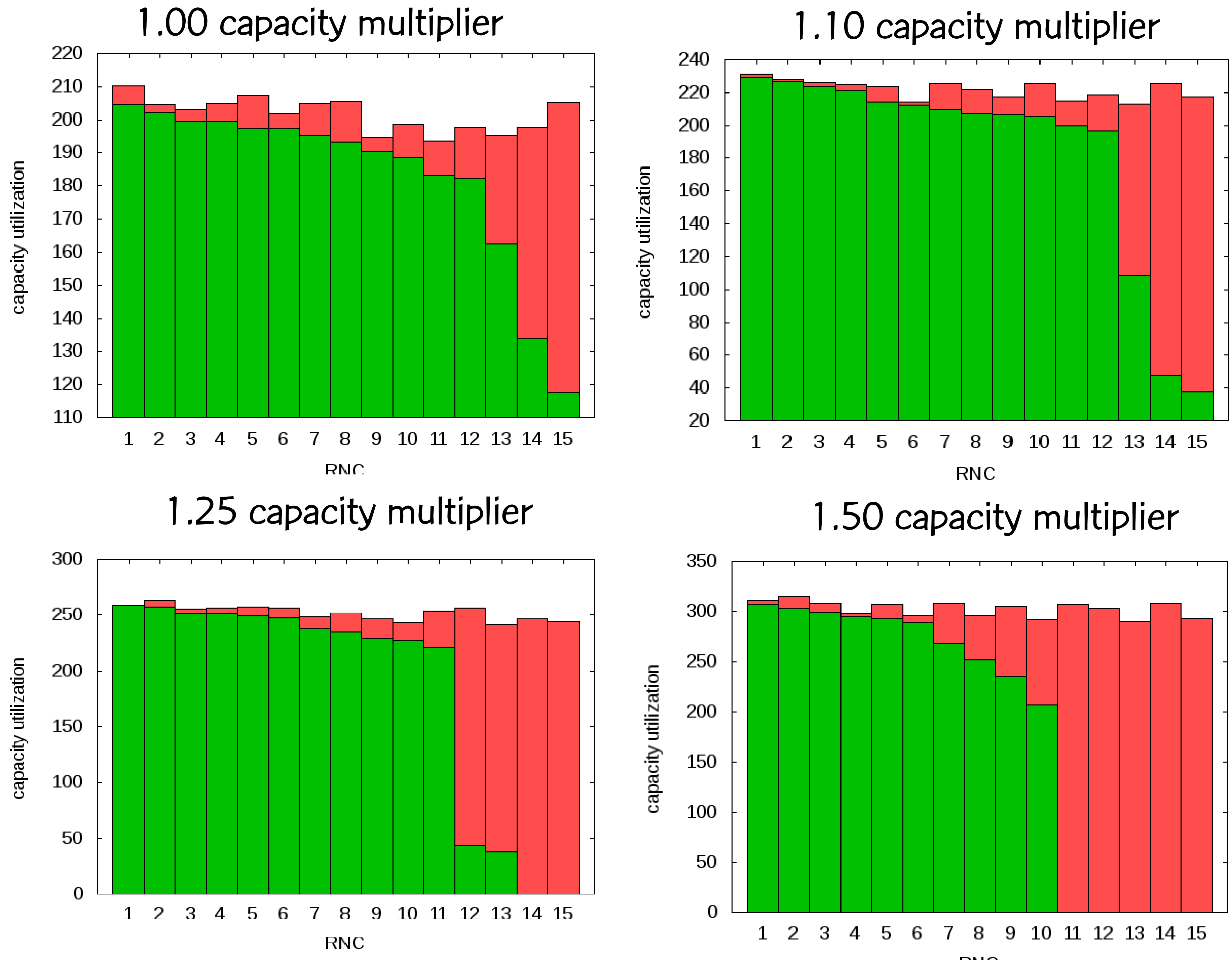


reduction

7%

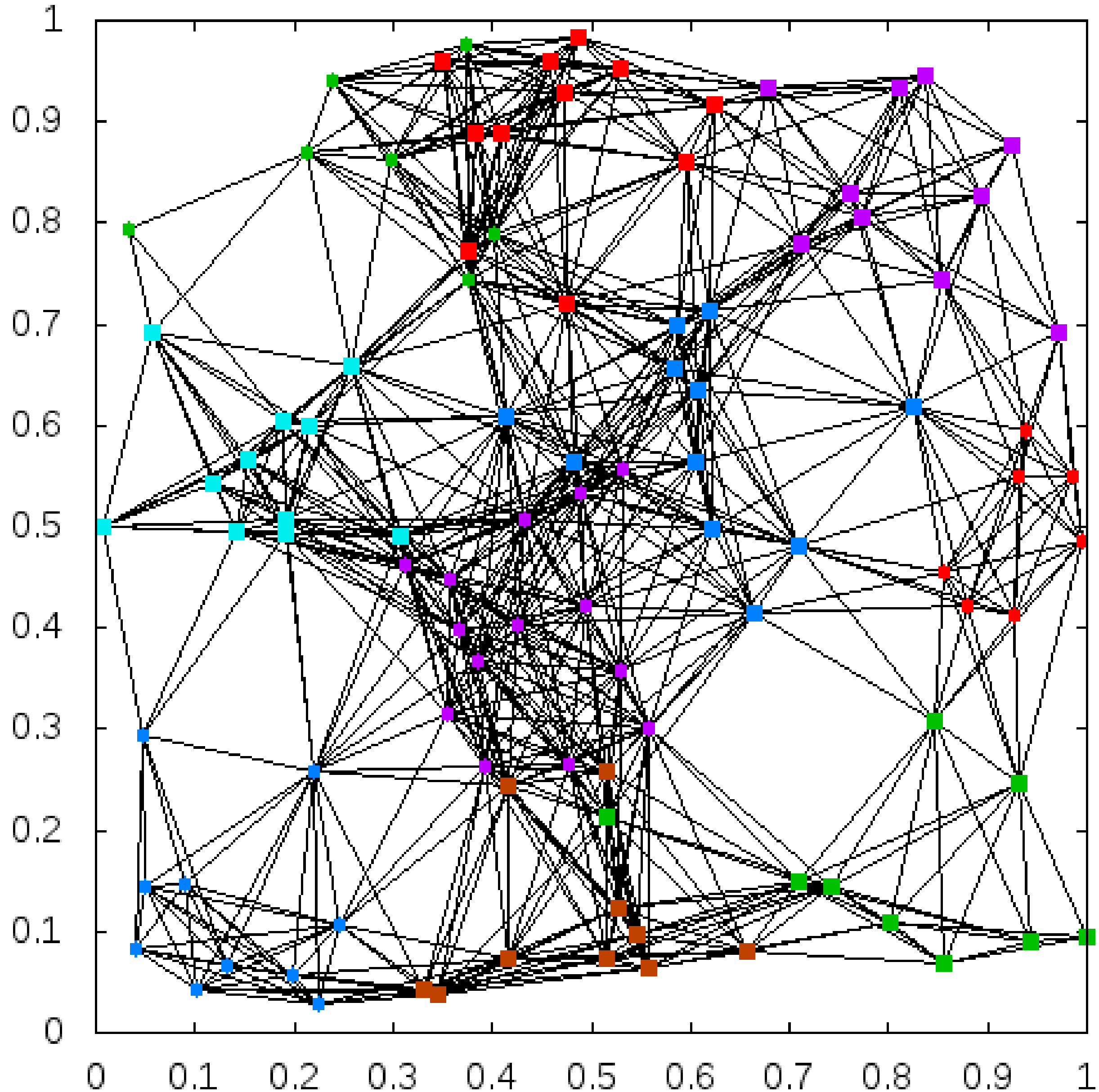
17%

26%


1.00 capacity multiplier
1.10 capacity multiplier
1.25 capacity multiplier
1.50 capacity multiplier
capacity utilization
RNC
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
110 120 130 140 150 160 170 180 190 200 210 220
20 40 60 80 100 120 140 160 180 200 220 240
0 50 100 150 200 250 300
0 50 100 150 200 250 300 350

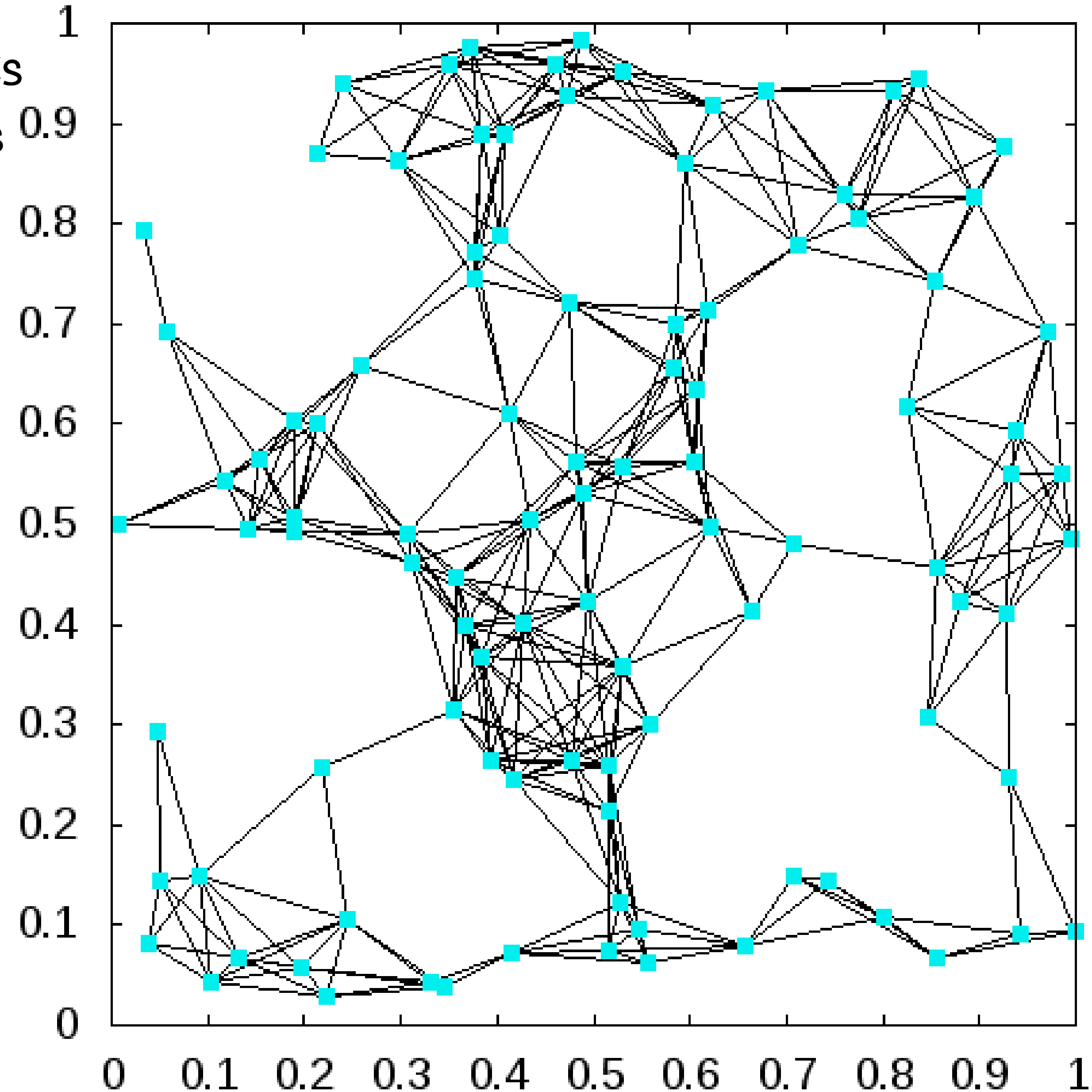
Solution with 70990 handovers

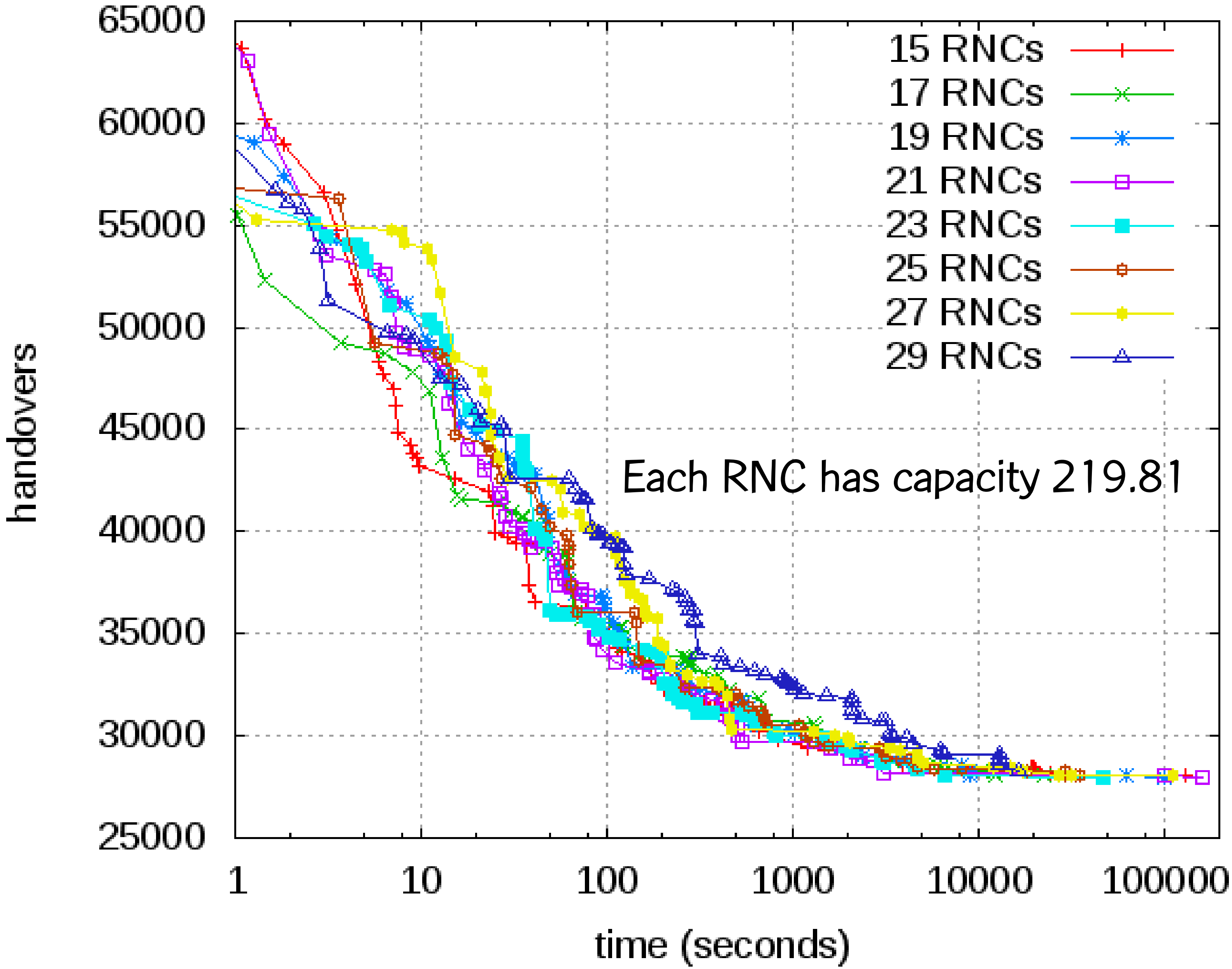
15 RNCs
100 towers
r = 0.25



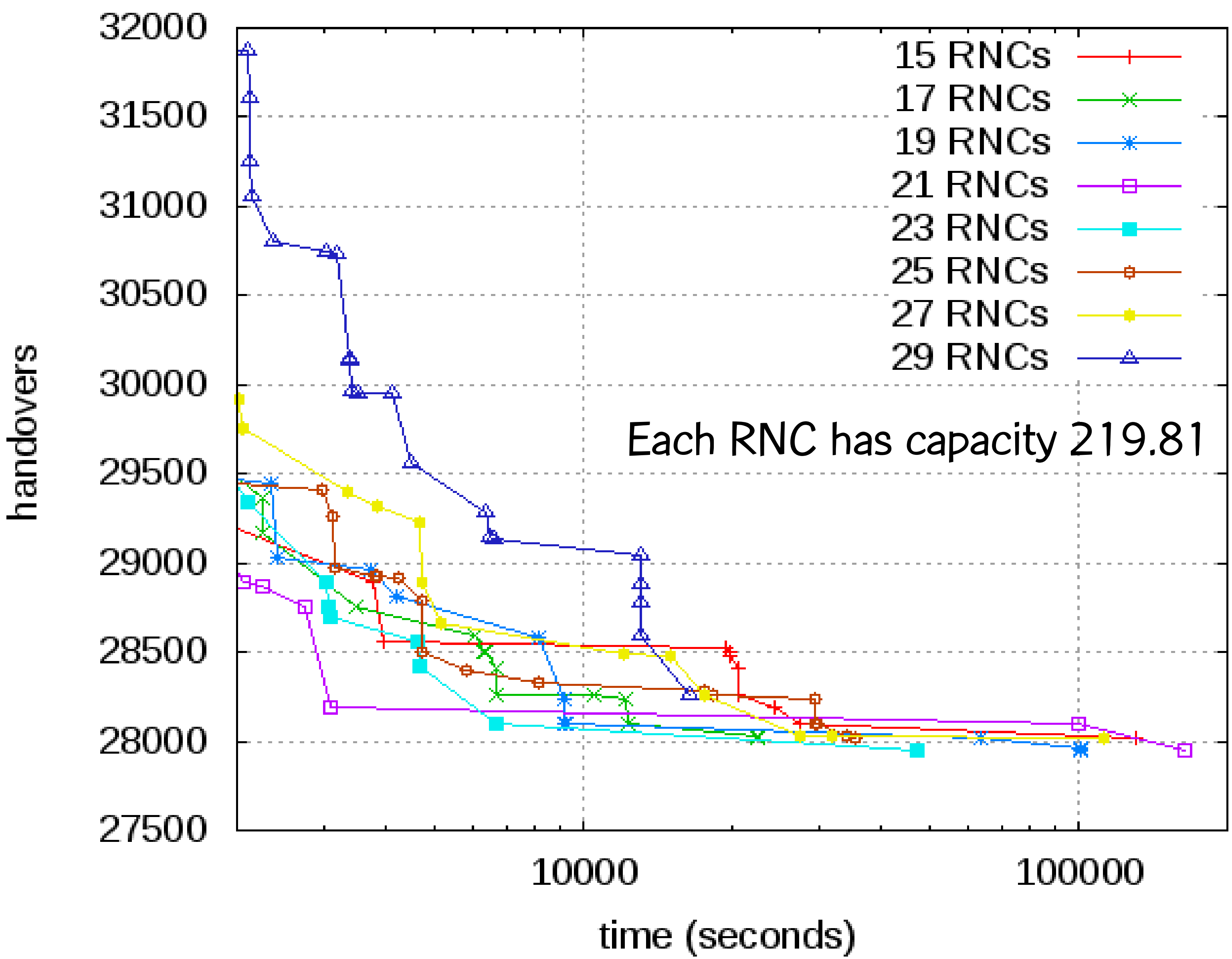
1.50 capacity multiplier

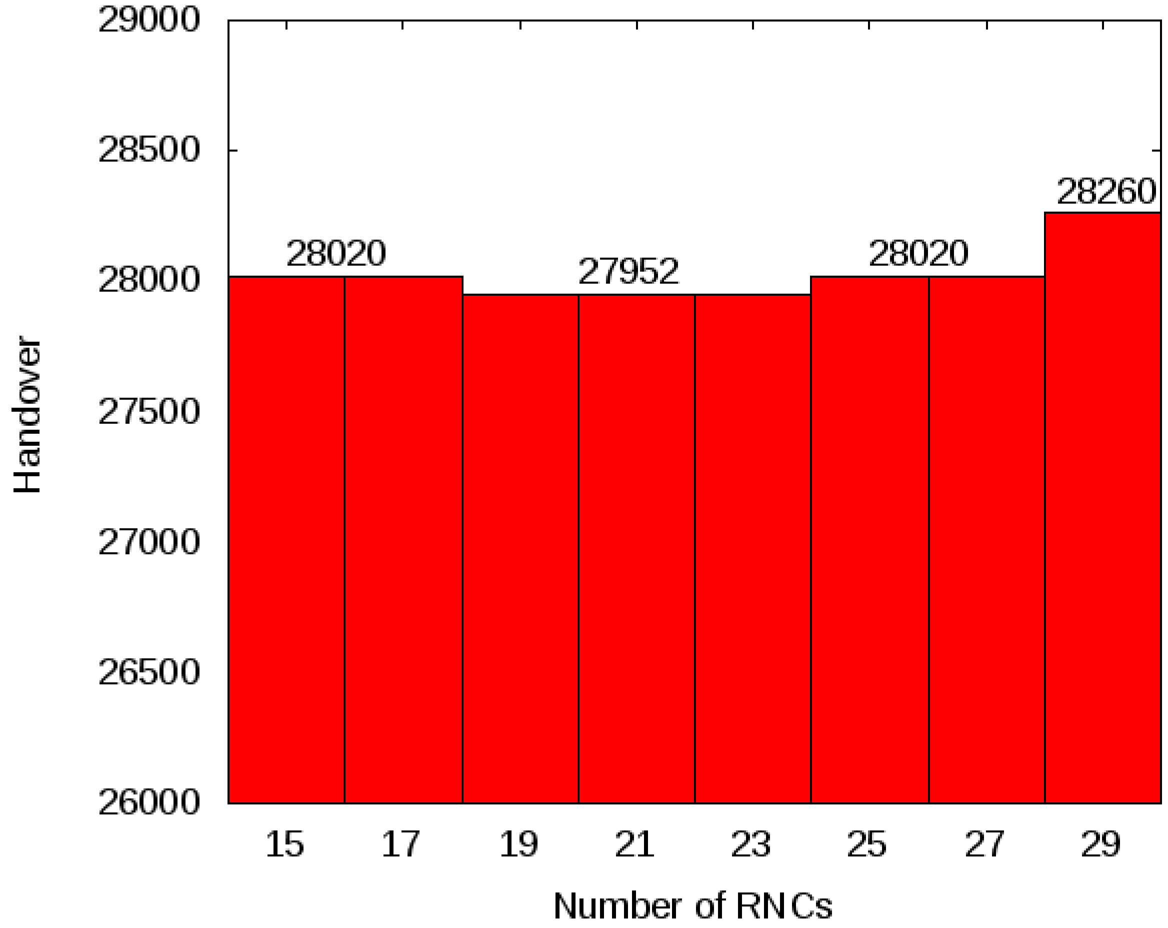
15-29 RNCs
100 towers
r = 0.17

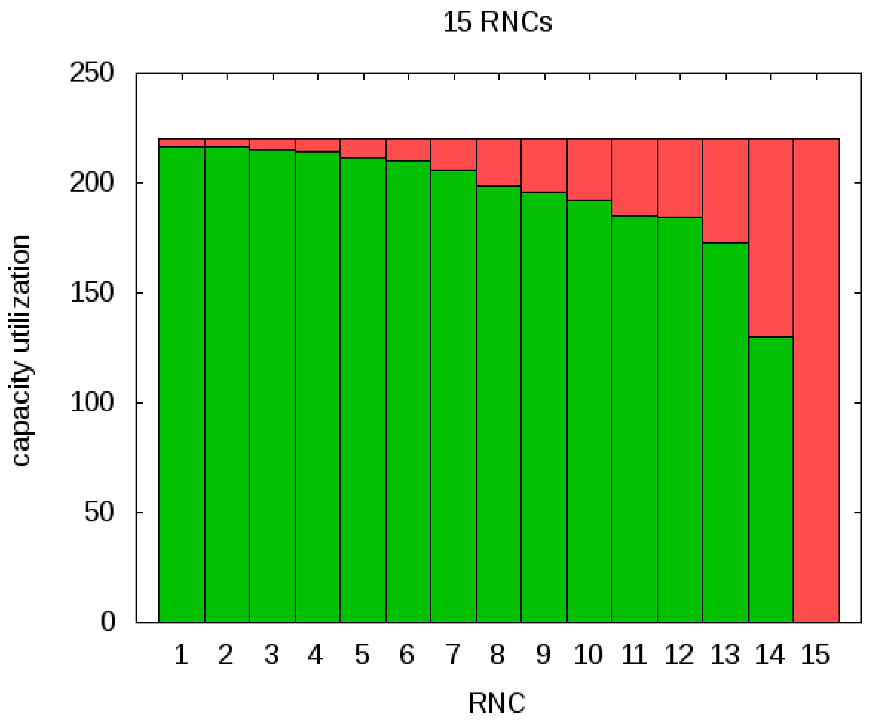


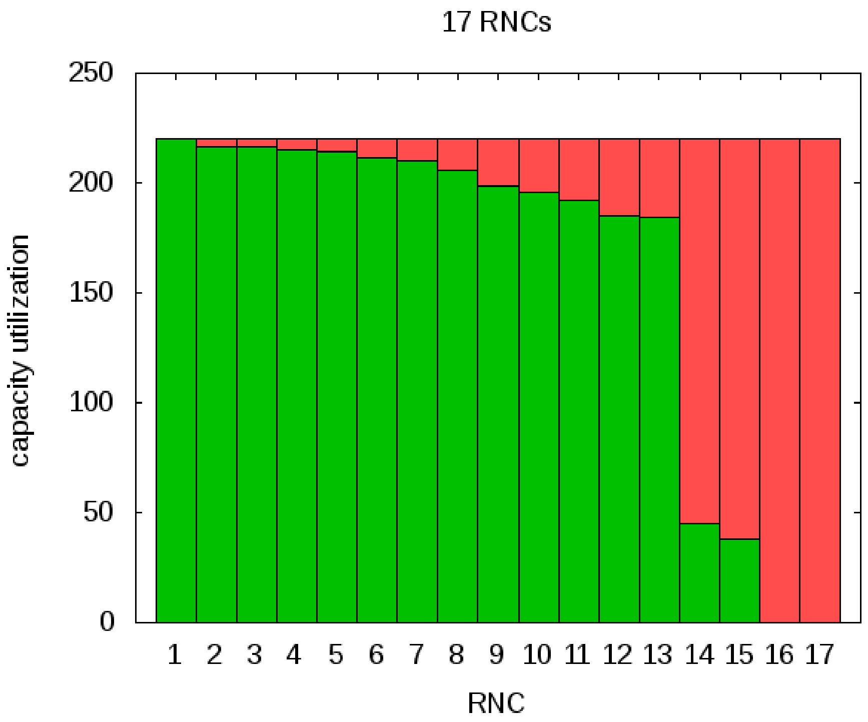


65000
60000
55000
50000
45000
40000
35000
30000
25000
handovers
1
10
100
1000
10000
100000
time (seconds)
15 RNCs
17 RNCs
19 RNCs
21 RNCs
23 RNCs
25 RNCs
27 RNCs
29 RNCs
Each RNC has capacity 219.81

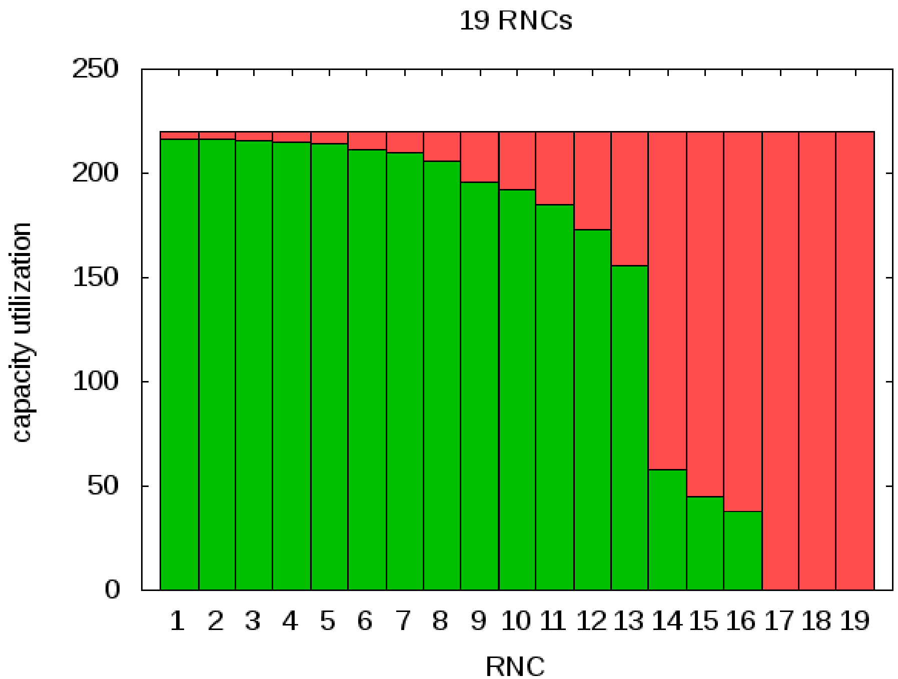

32000
31500
31000
30500
30000
29500
29000
28500
28000
27500
handovers
10000
100000
time (seconds)
15 RNCs
17 RNCs
19 RNCs
21 RNCs
23 RNCs
25 RNCs
27 RNCs
29 RNCs
Each RNC has capacity 219.81

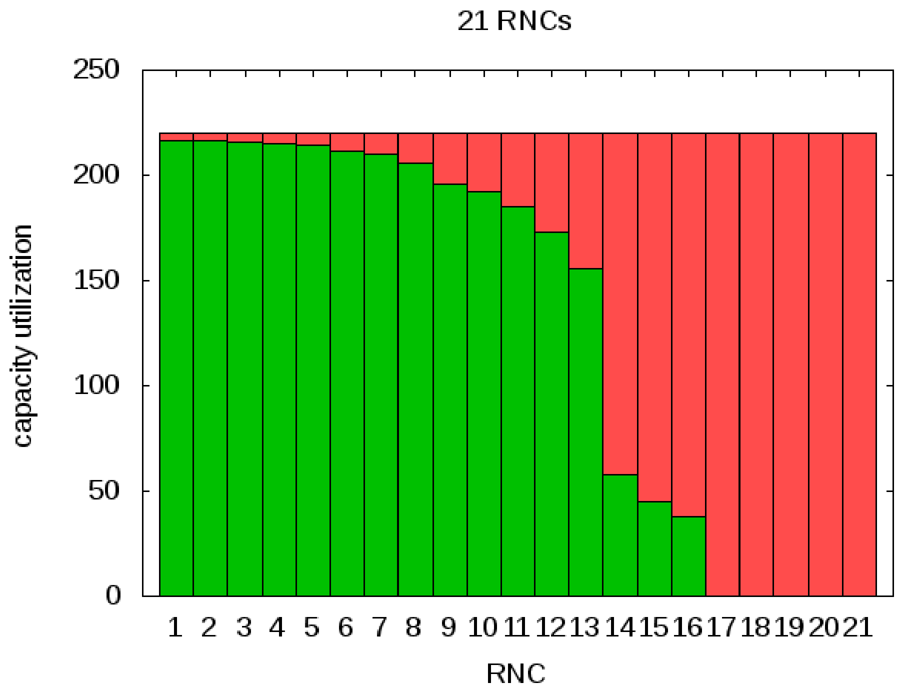

29000
28500
28000
27500
27000
26500
26000
Handover
28020
27952
28020
28260
15
17
19
21
23
25
27
29
Number of RNCs

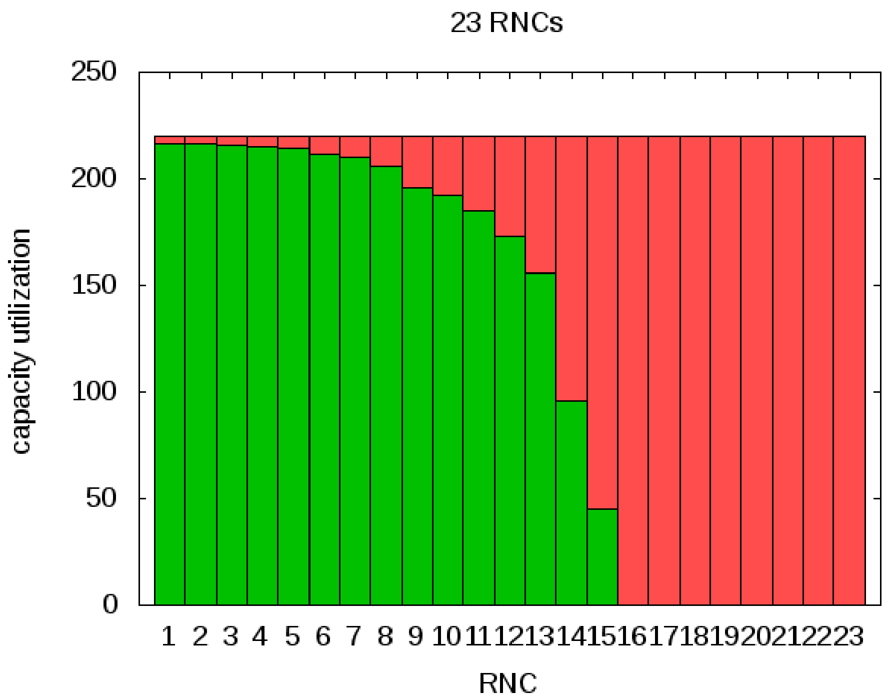




28020 handovers

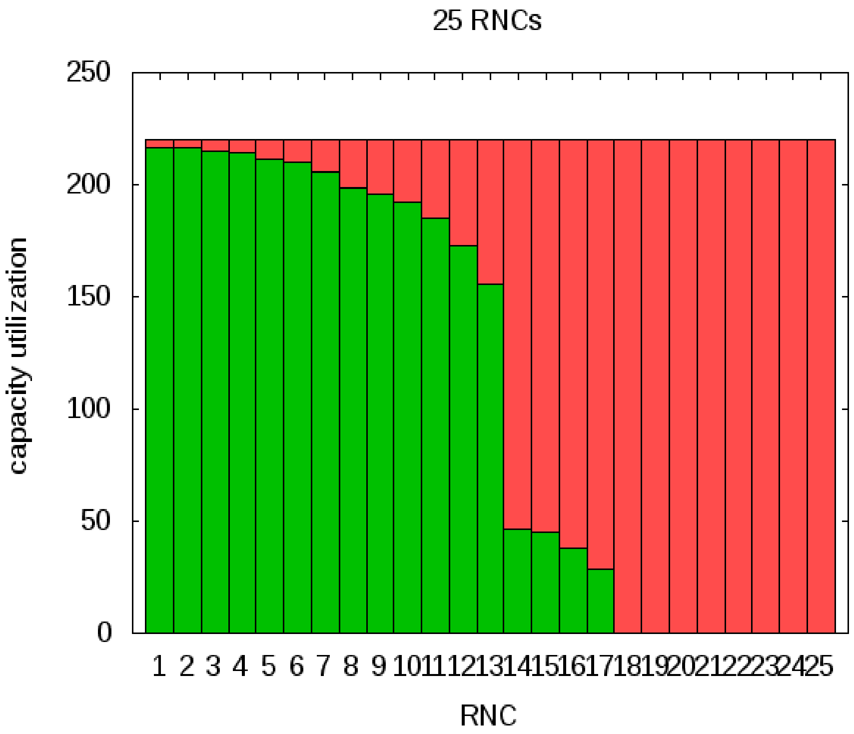


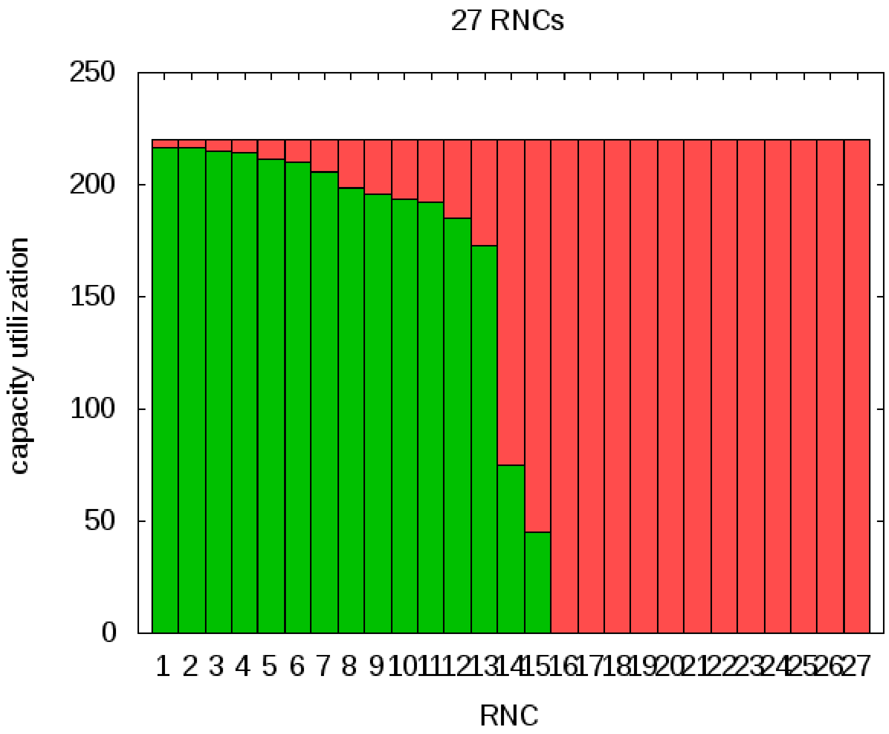




27952 handovers

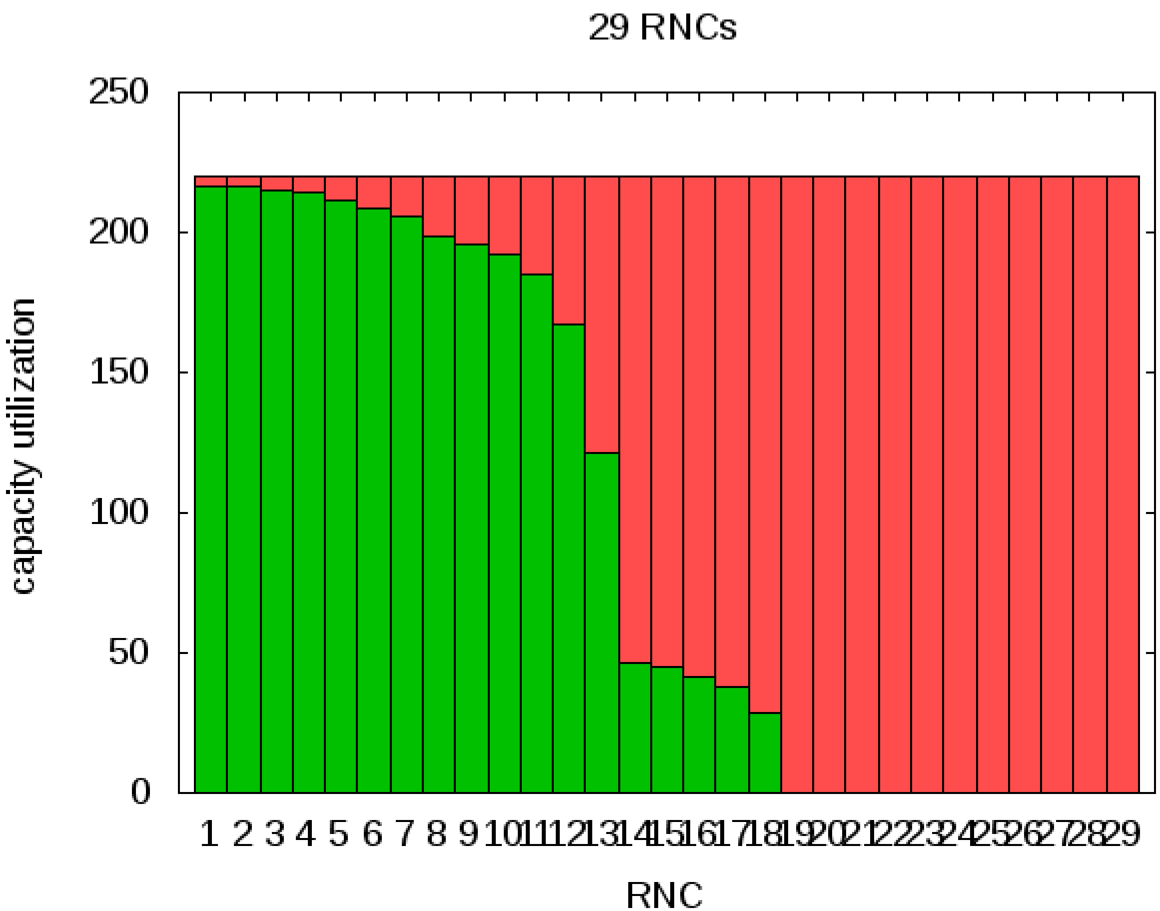
at&t
Your world. Delivered.



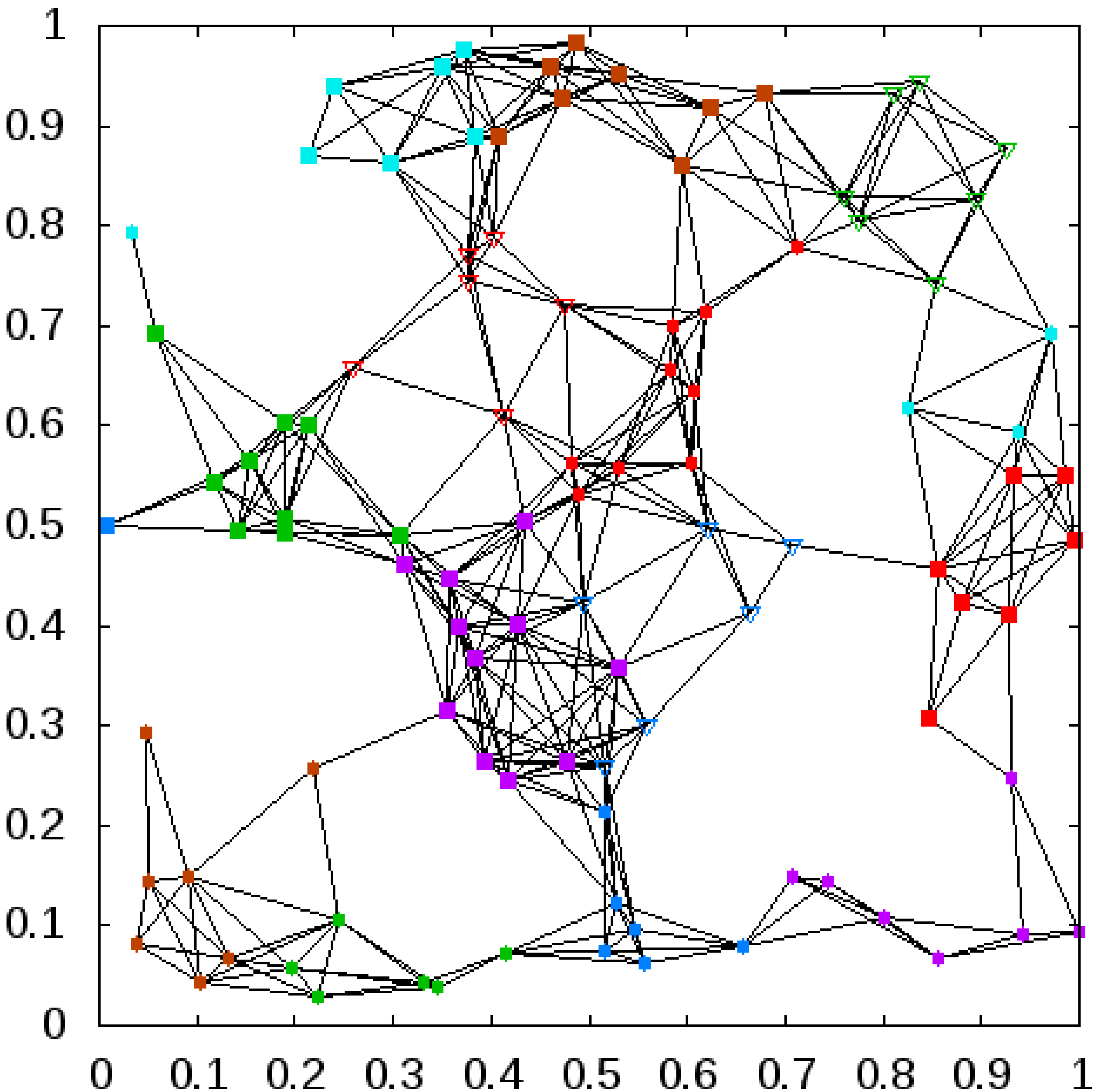


28020 handovers

28260 handovers



15 RNCs
100 towers
r = 0.17



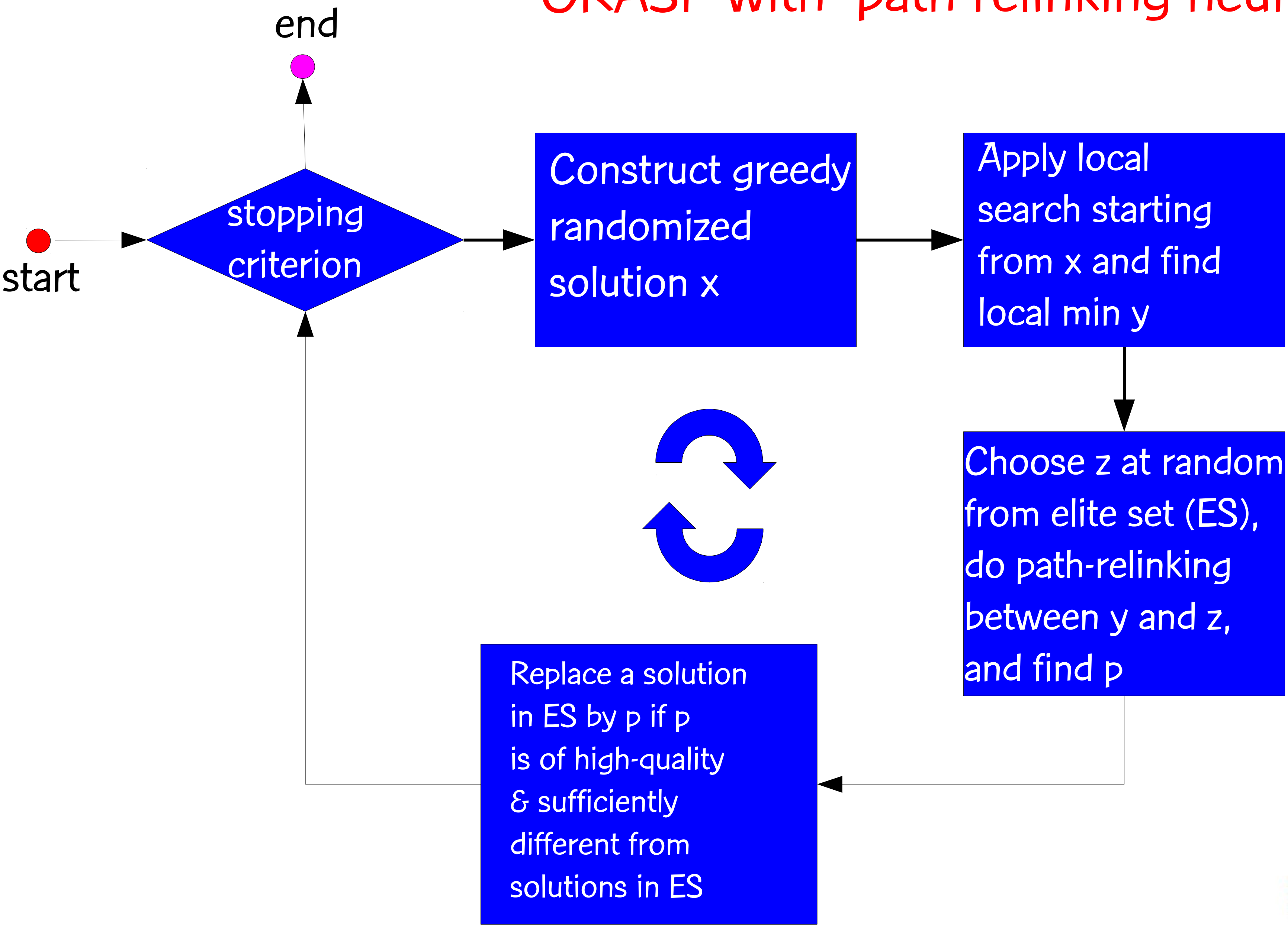
27952 handovers

15 RNCs

Solution from run with 23 RNCs

Solution method

GRASP with path-relinking heuristic



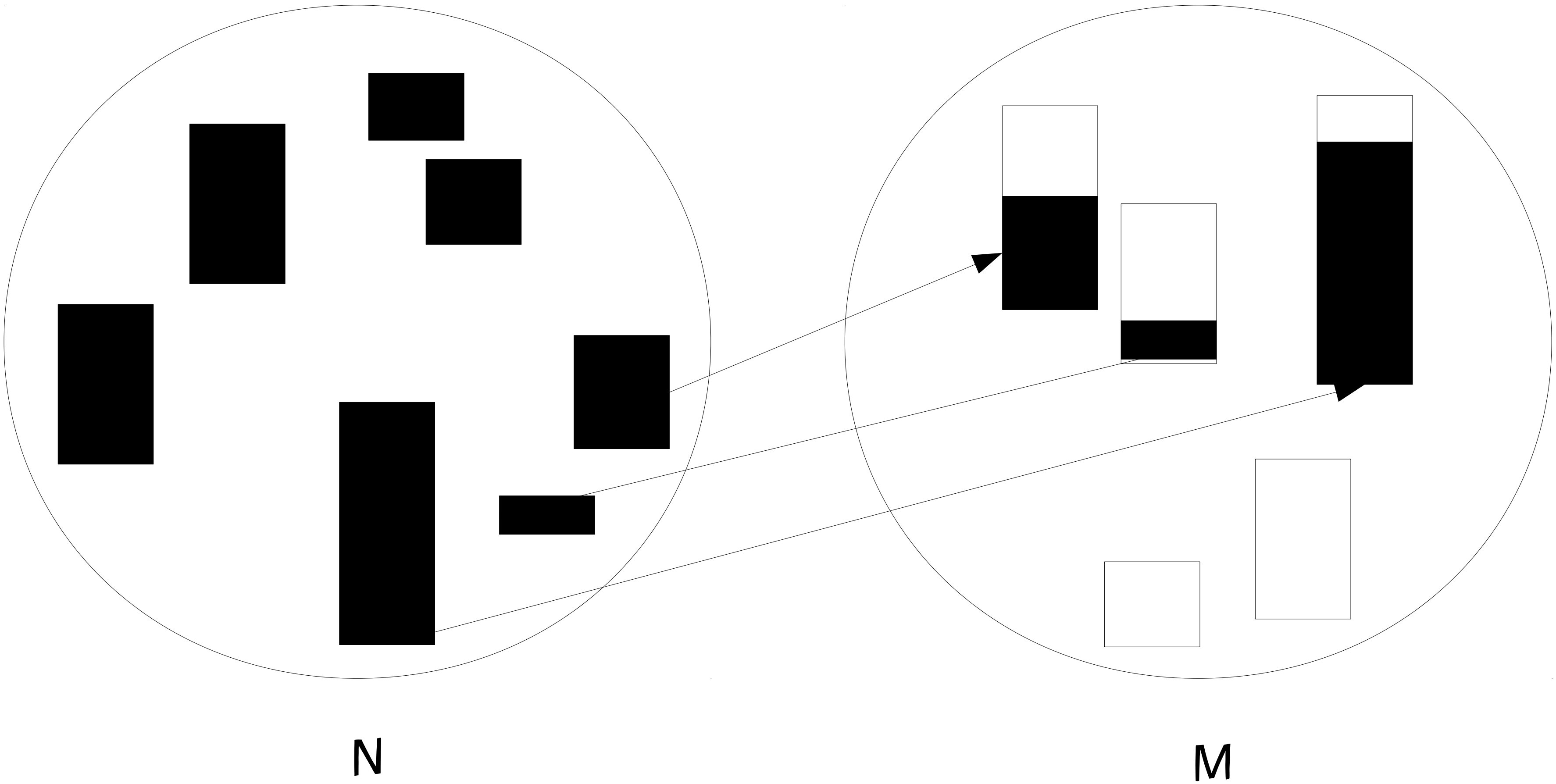
Components

Construction of greedy randomized solution

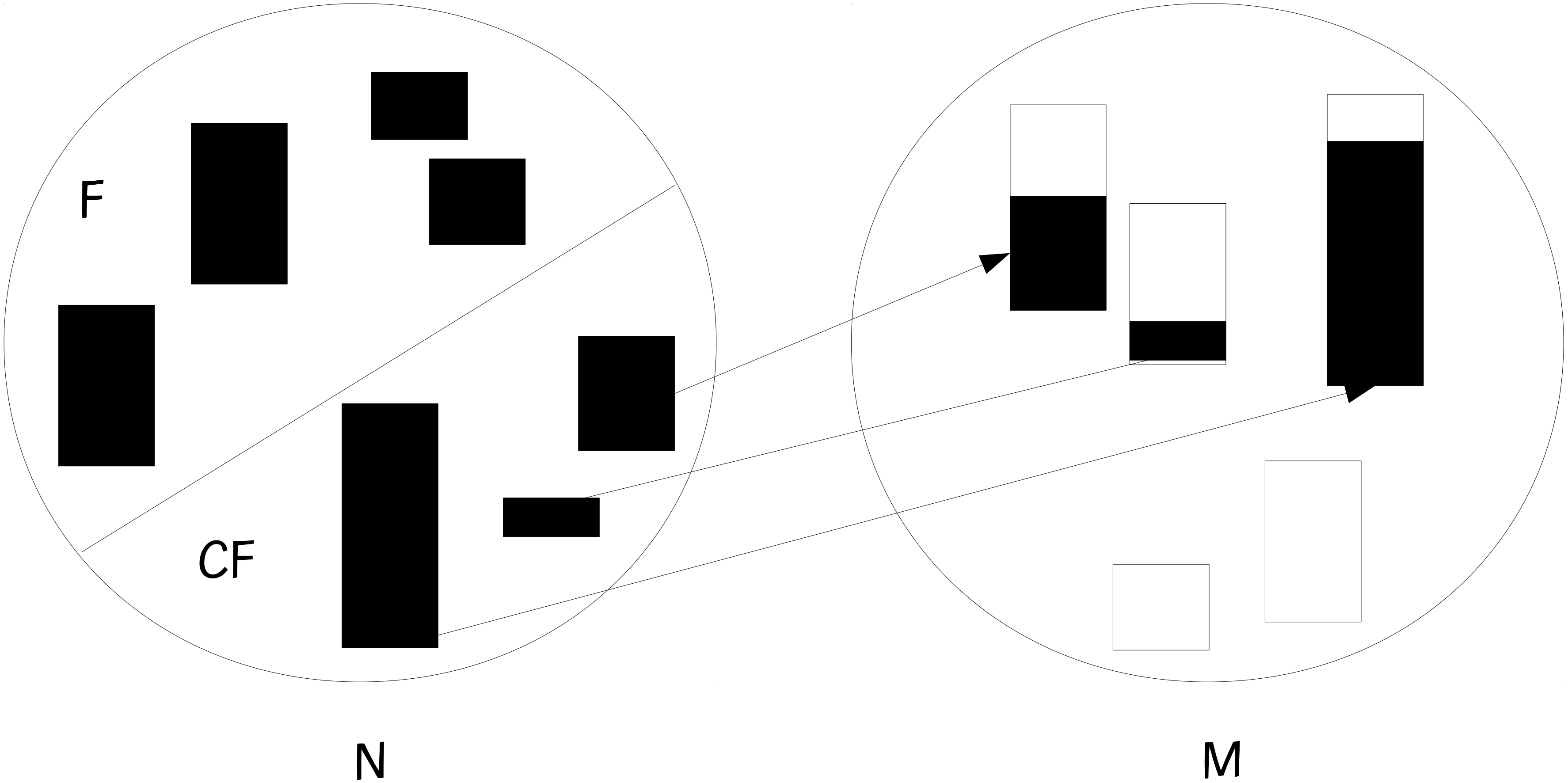
Local search

Path-relinking

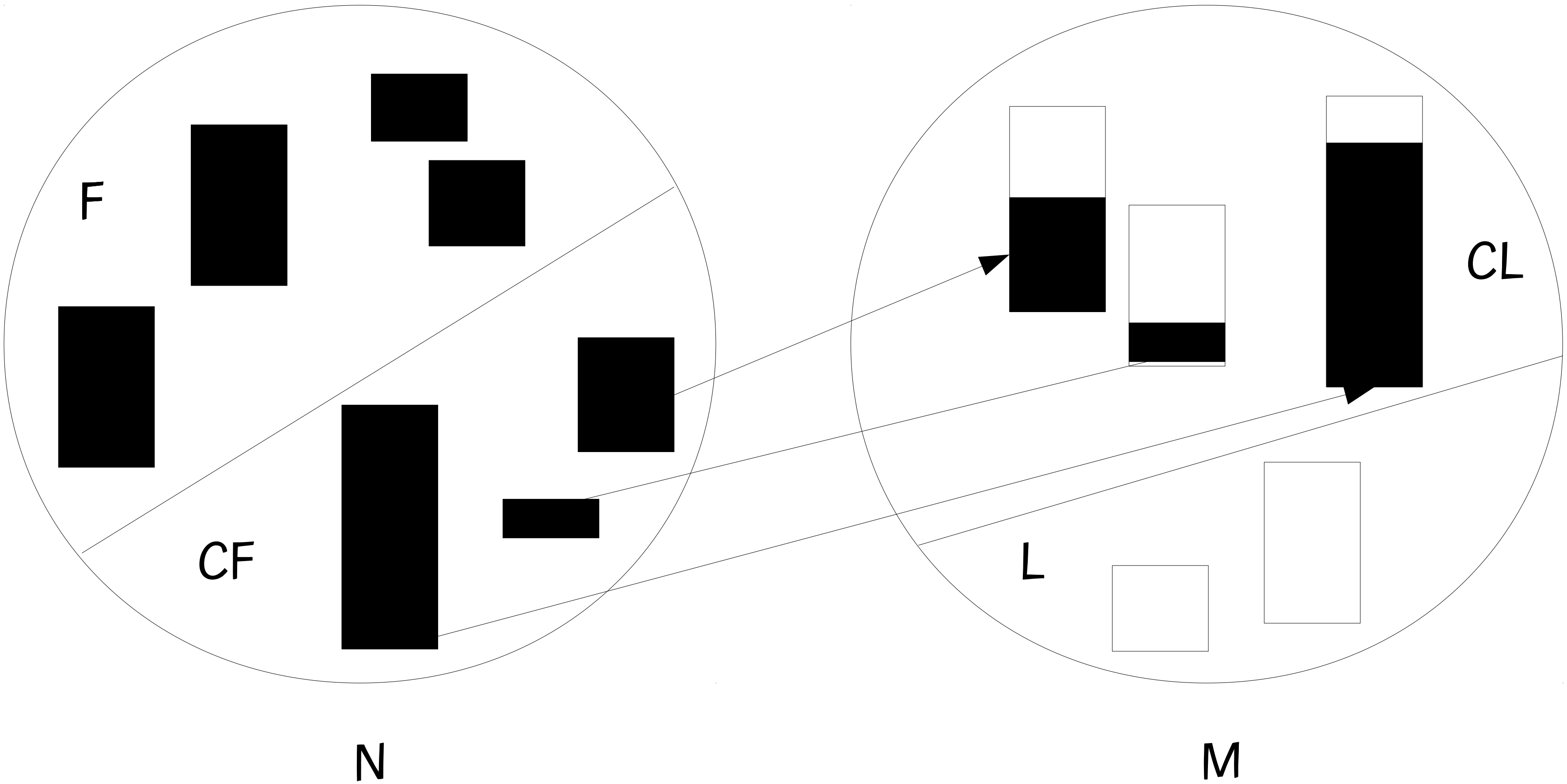
GRASP construction for GQAP



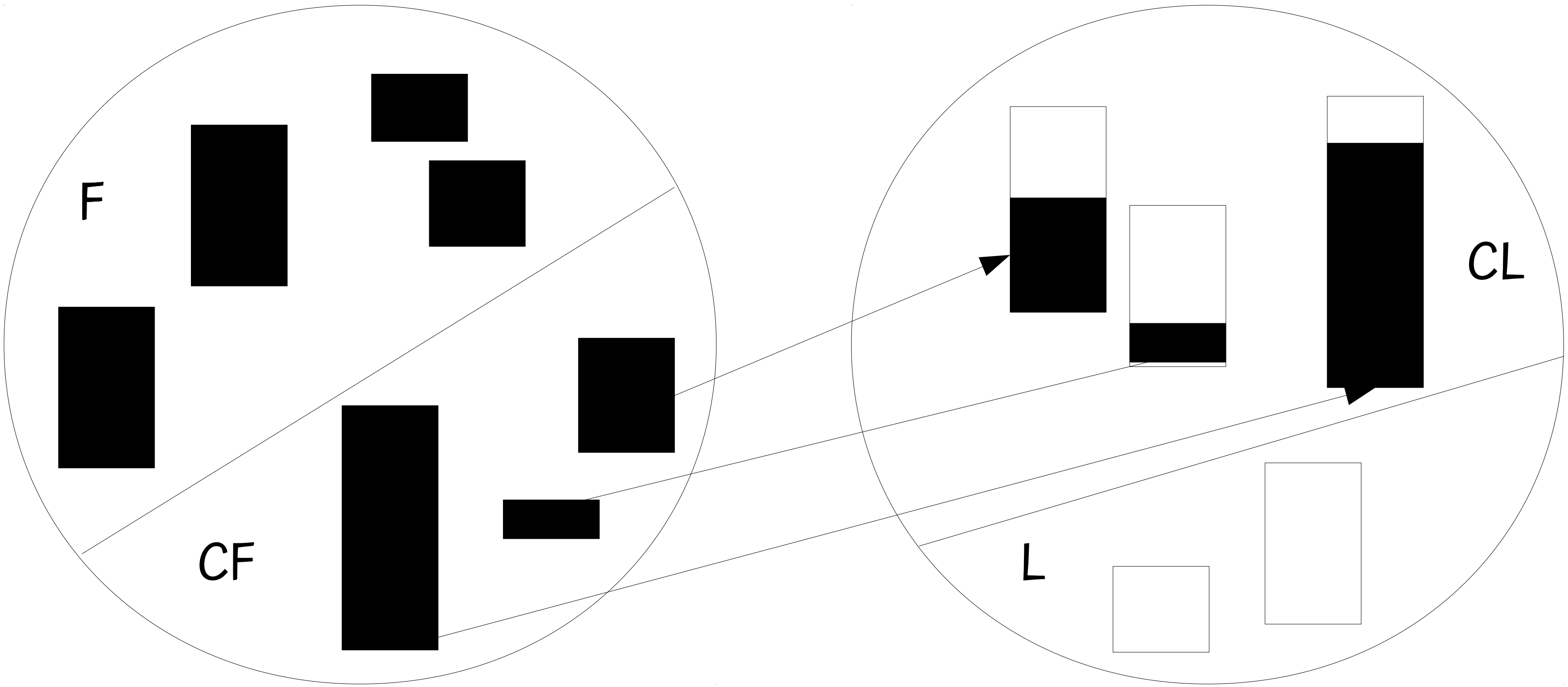
Suppose a number of assignments have already been made



$N = F \cup CF$, where CF is the set of assigned facilities and F the set of facilities not yet assigned to some location



$M = L \cup CL$, where CL is the set of previously chosen locations and L the set of unselected locations.

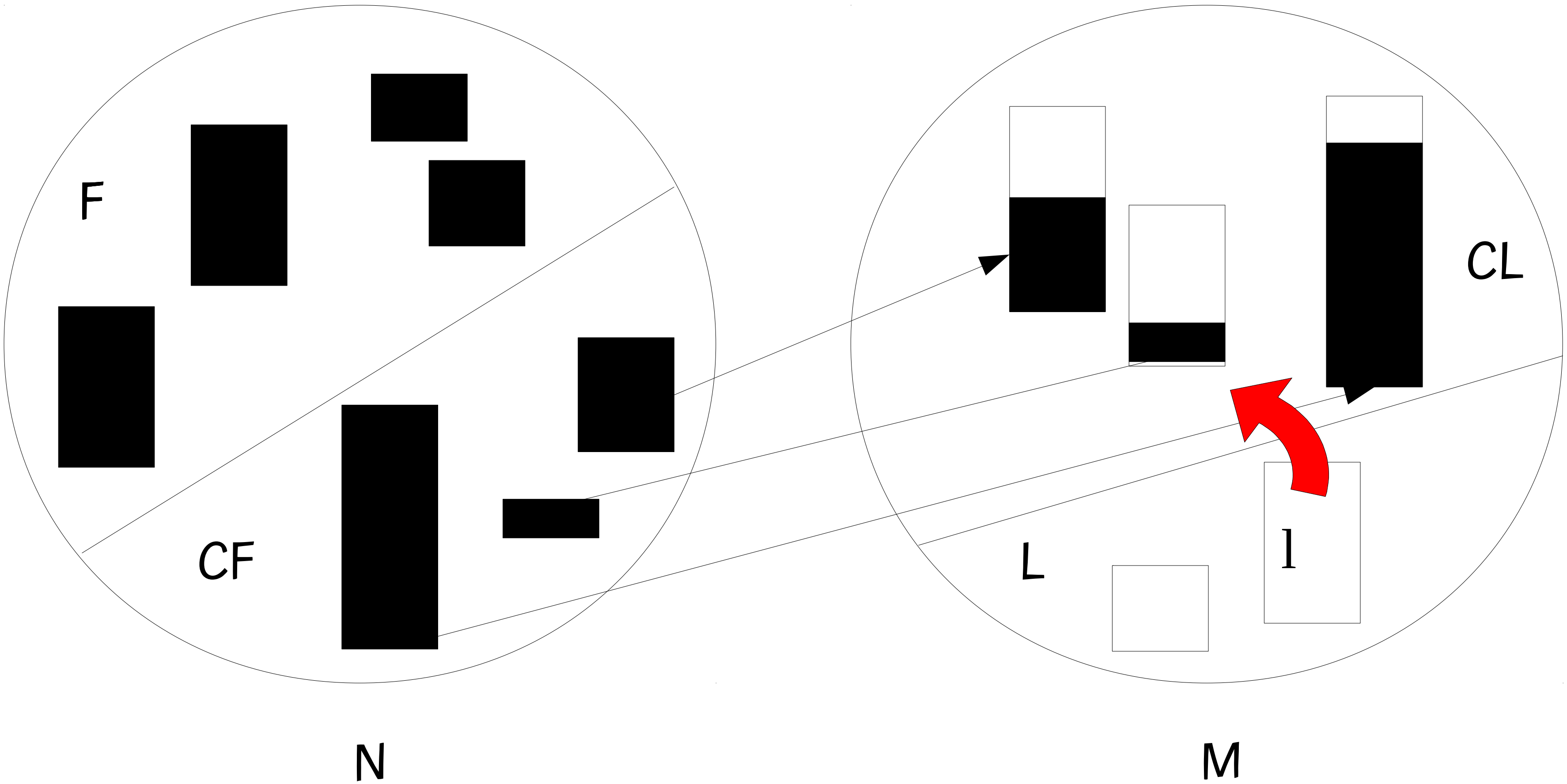


M

Components of construction procedure:

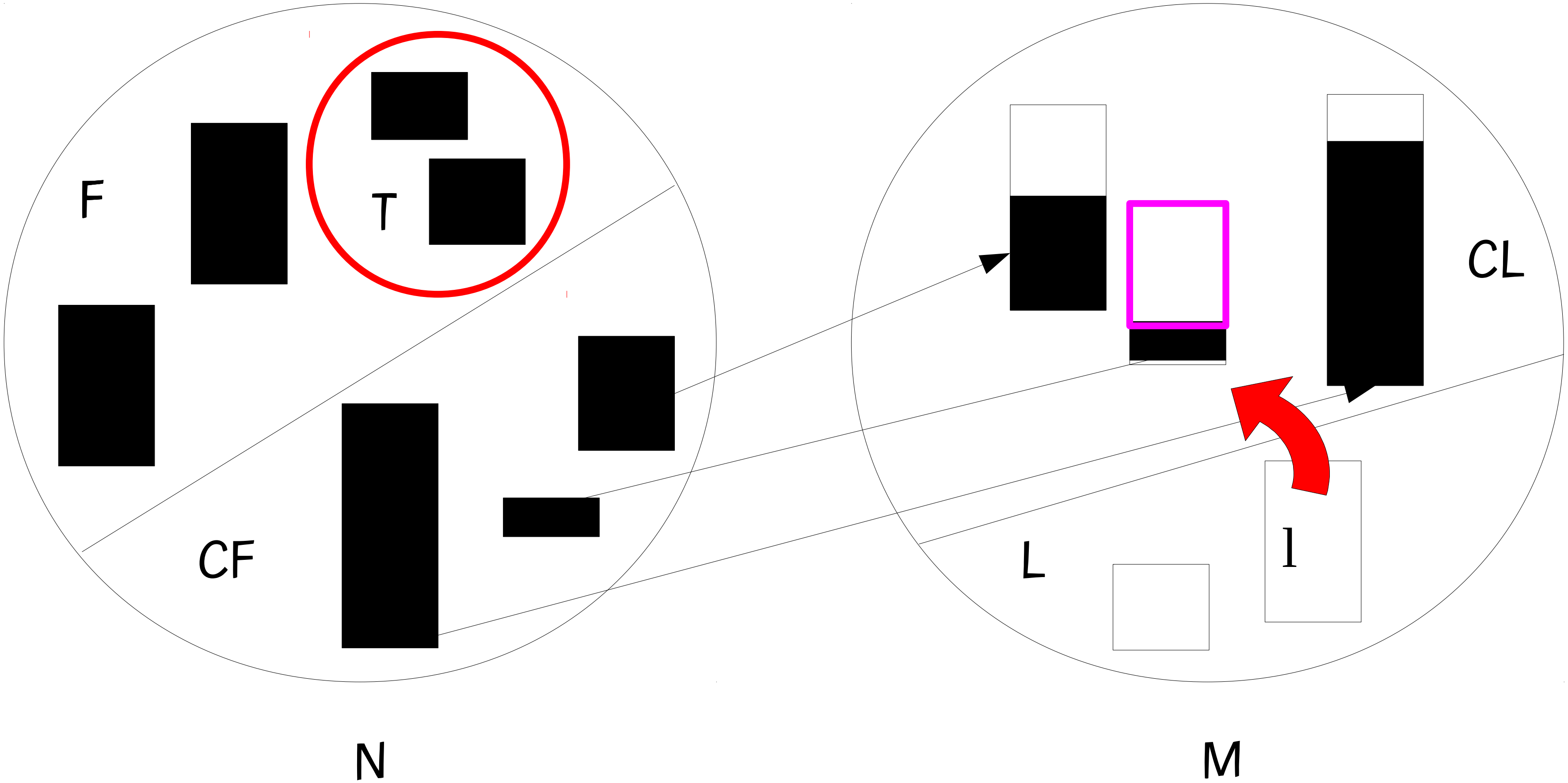
- procedure to select a NEW location from set L;
- procedure to select a facility from set F;
- procedure to select a location from set CL;

Procedure to select a NEW location from set L



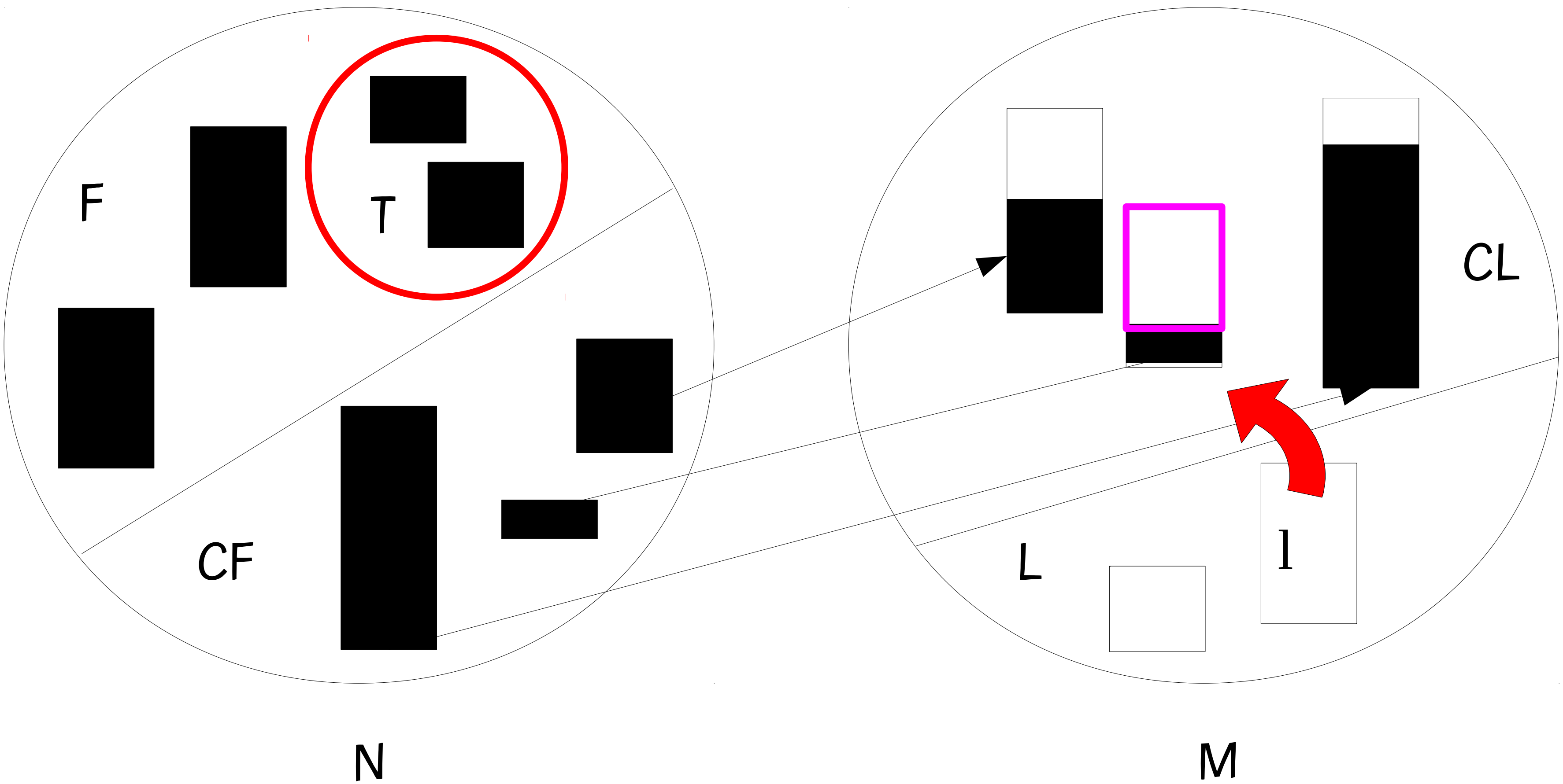
With probability P, randomly select a new location l from L, favoring those having high capacity and those close to all locations in CL, and move location l to CL.

Procedure to select a new location from set L



The probability P is equal to 1−(|T|/|F|), where the set T consists of all unassigned facilities with demands less than or equal to the maximum available capacity of locations in CL.

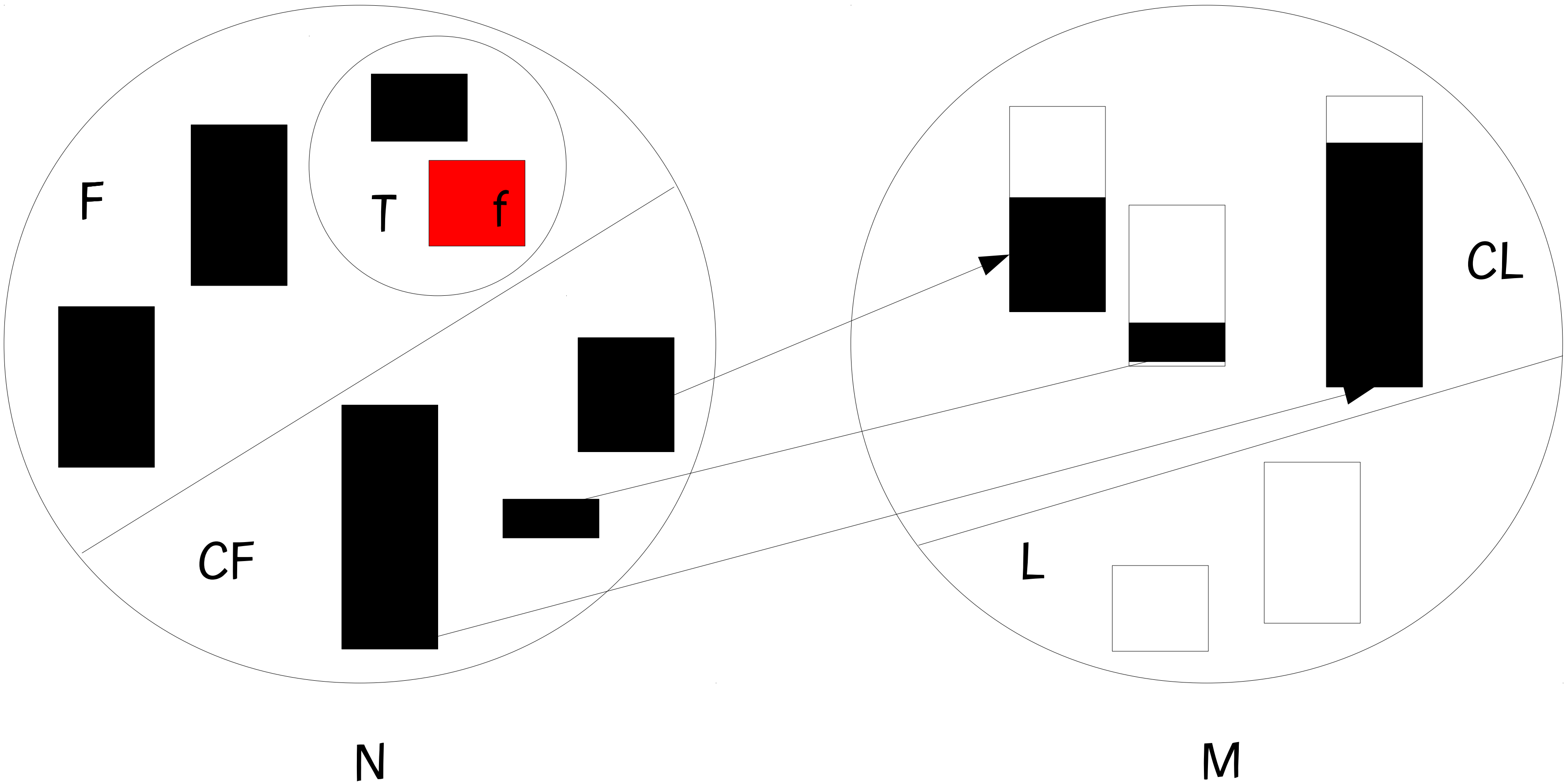
Procedure to select a new location from set L



With P = 1−(|T|/|F|):

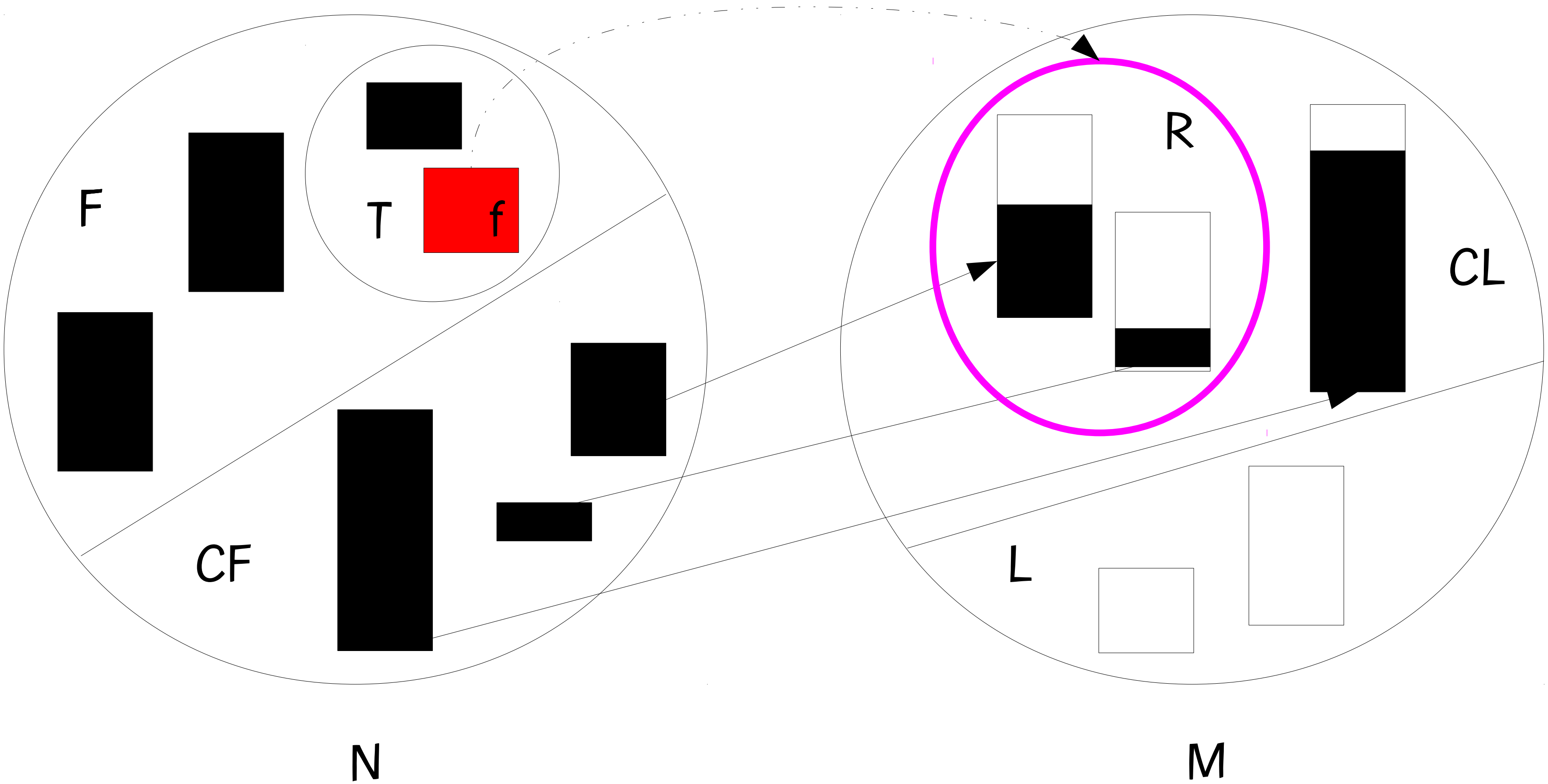
- if |T| is much less than |F|, then P tends to 1, which increases the need for a new location;
- if |T| tends to |F|, then P tends to 0, which reduces the need for a new location;

Facility selection procedure



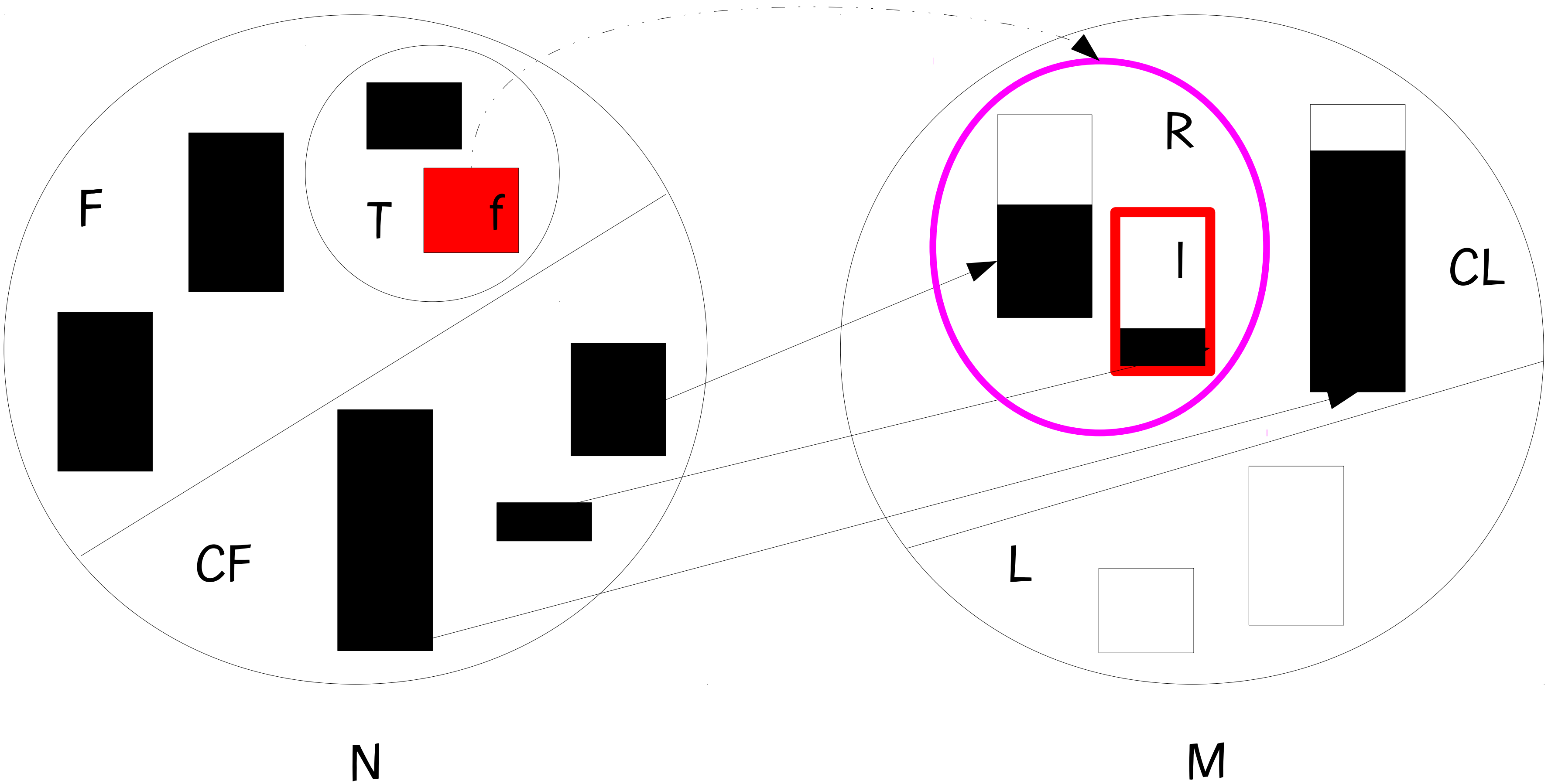
Randomly select a facility $f \in T$ favoring facilities that have high demand and high flows to other facilities.

Procedure to select a location from CL (step 1)



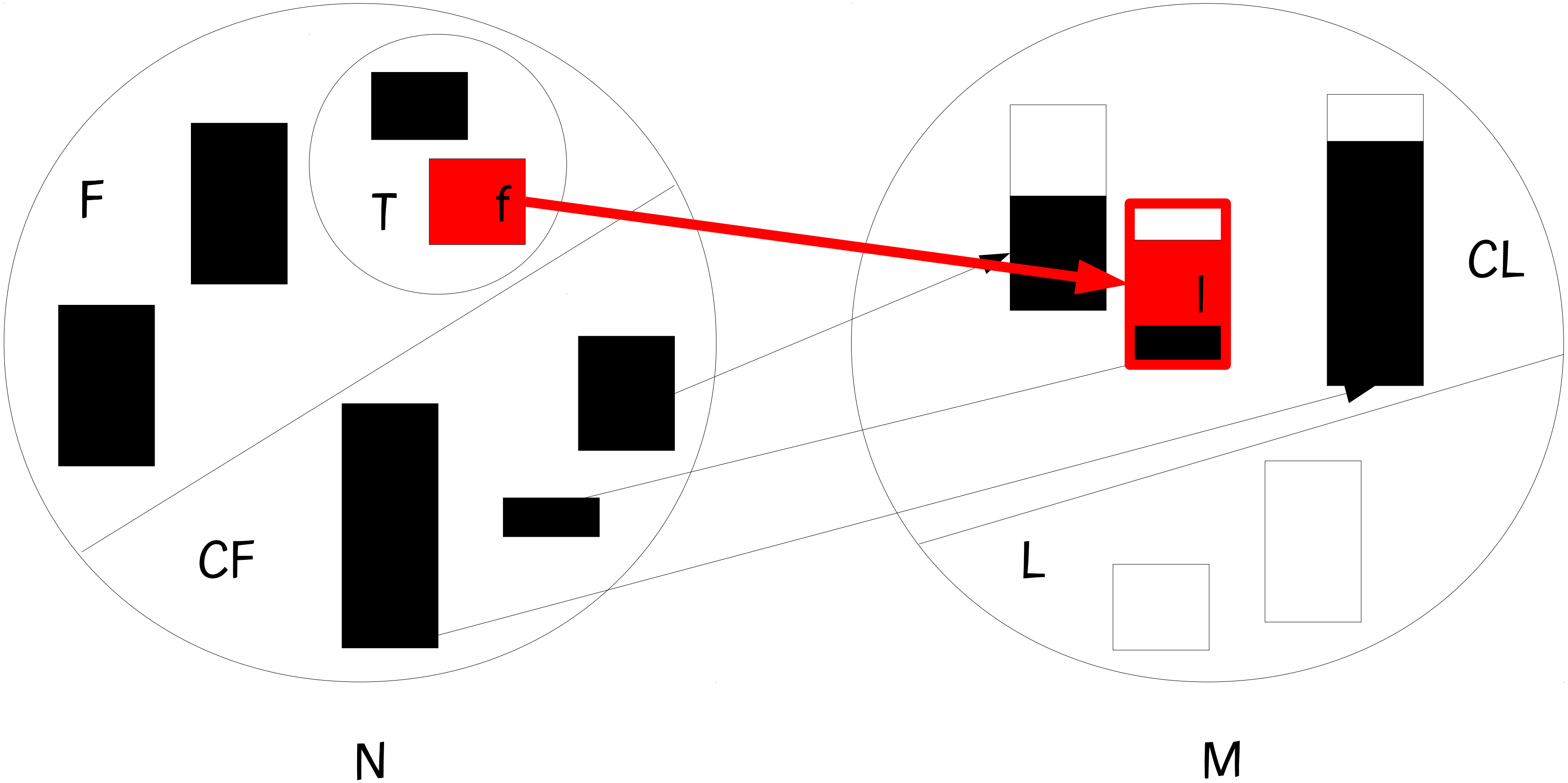
1. Let set R to be all locations in CL having slack greater than or equal to demand of facility f;

Procedure to select a location from CL (step 2)



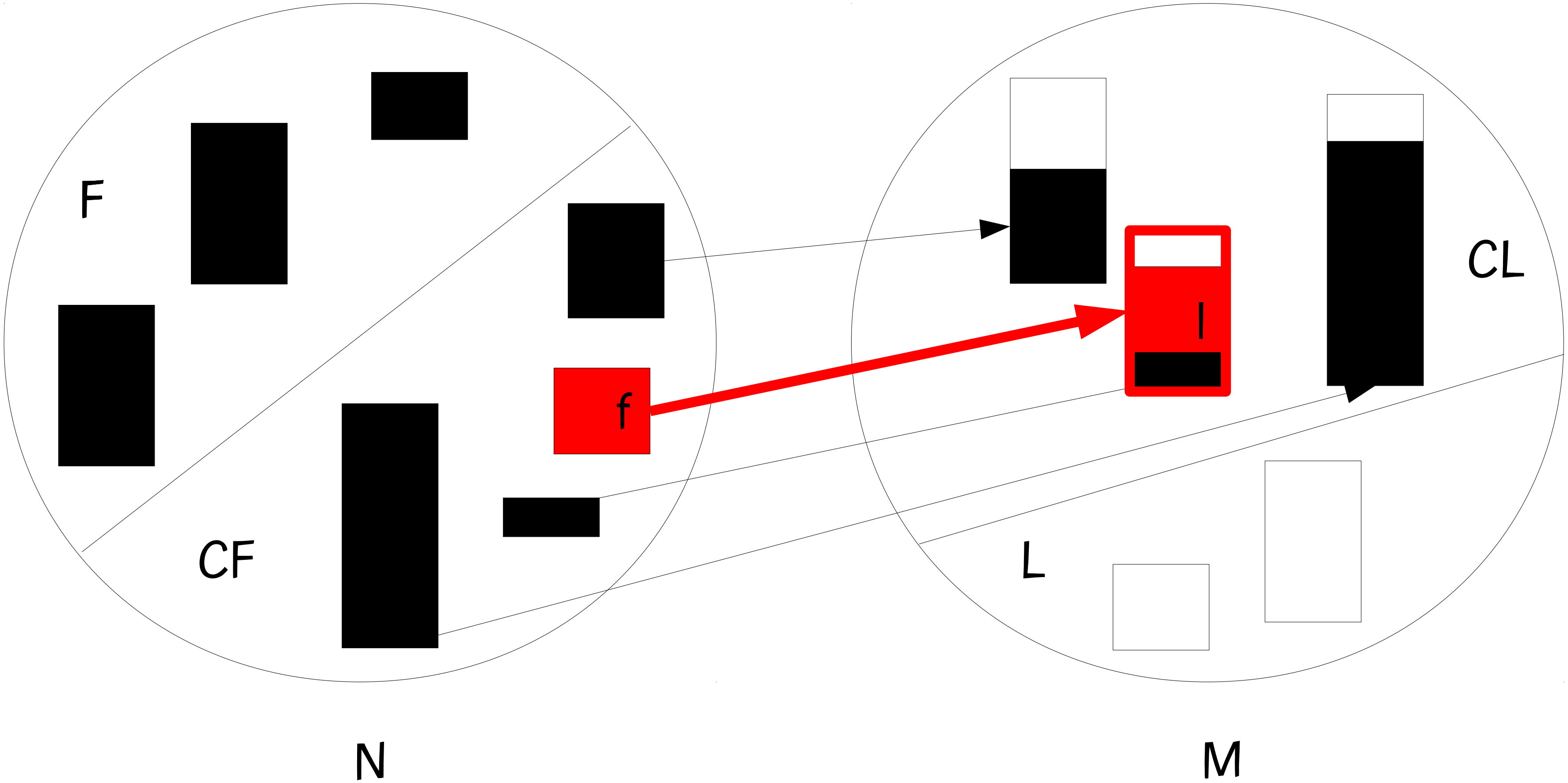
2. Randomly select a location l ∈ R favoring those having high available capacity and those close to high-capacity locations in CL;

Assignment procedure



Assign facility f to location l

Assignment procedure



Update sets F, CF, and slack of location l

Considerations about the construction procedure

The procedure is not guaranteed to produce a feasible solution.

To address this difficulty, the construction procedure is repeated a maximum number of times or until all facilities are assigned (i.e. until $F=\emptyset$).

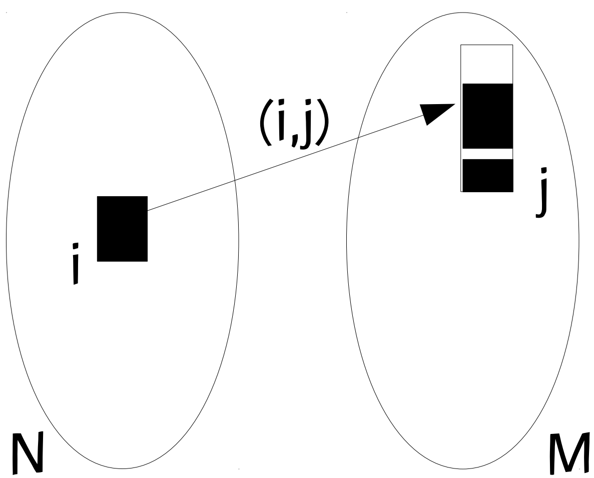
At start of construction, a location l is selected from the set L with probability proportional to its capacity. Location l is placed in CL.

Local search for GQAP

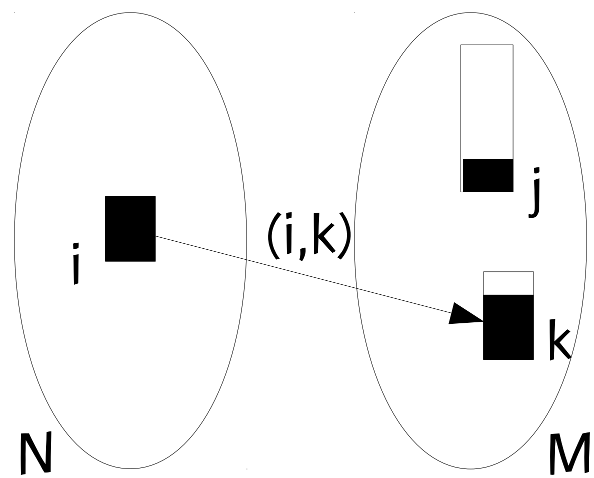
Local search

1-move and 2-move neighborhoods from solution p are used in our local search.

1-move: changing one facility-to-location assignment in p



solution p



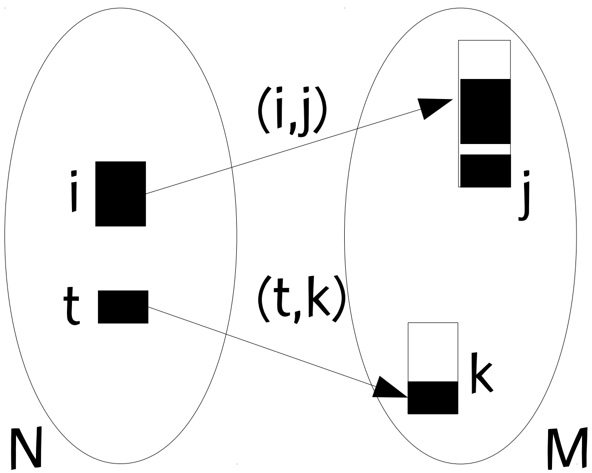
1-move neighbor of p

Local search

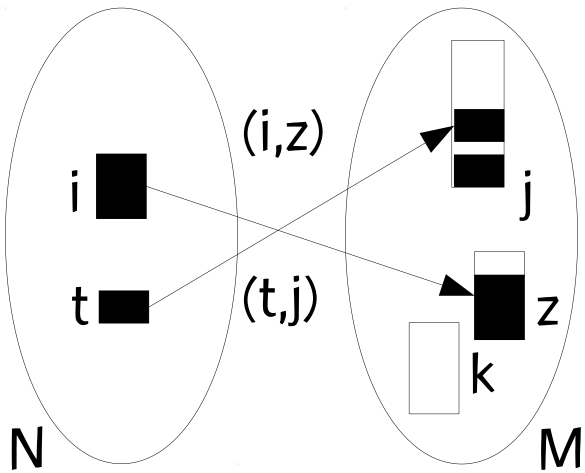
1-move and 2-move neighborhoods from solution p are used in our local search.

1-move: changing one facility-to-location assignment in p

2-move: changing two facility-to-location assignment in p.

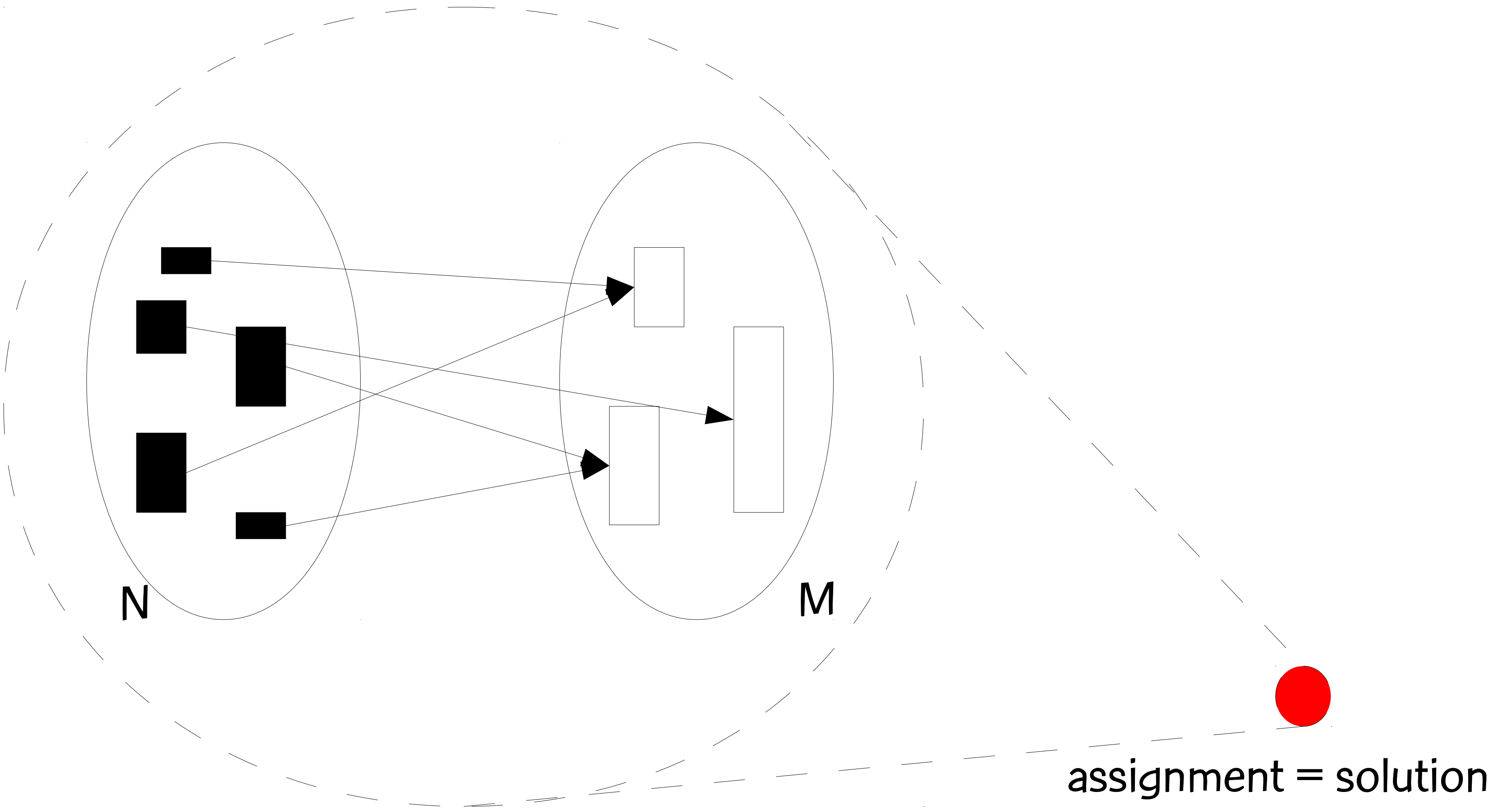


solution p

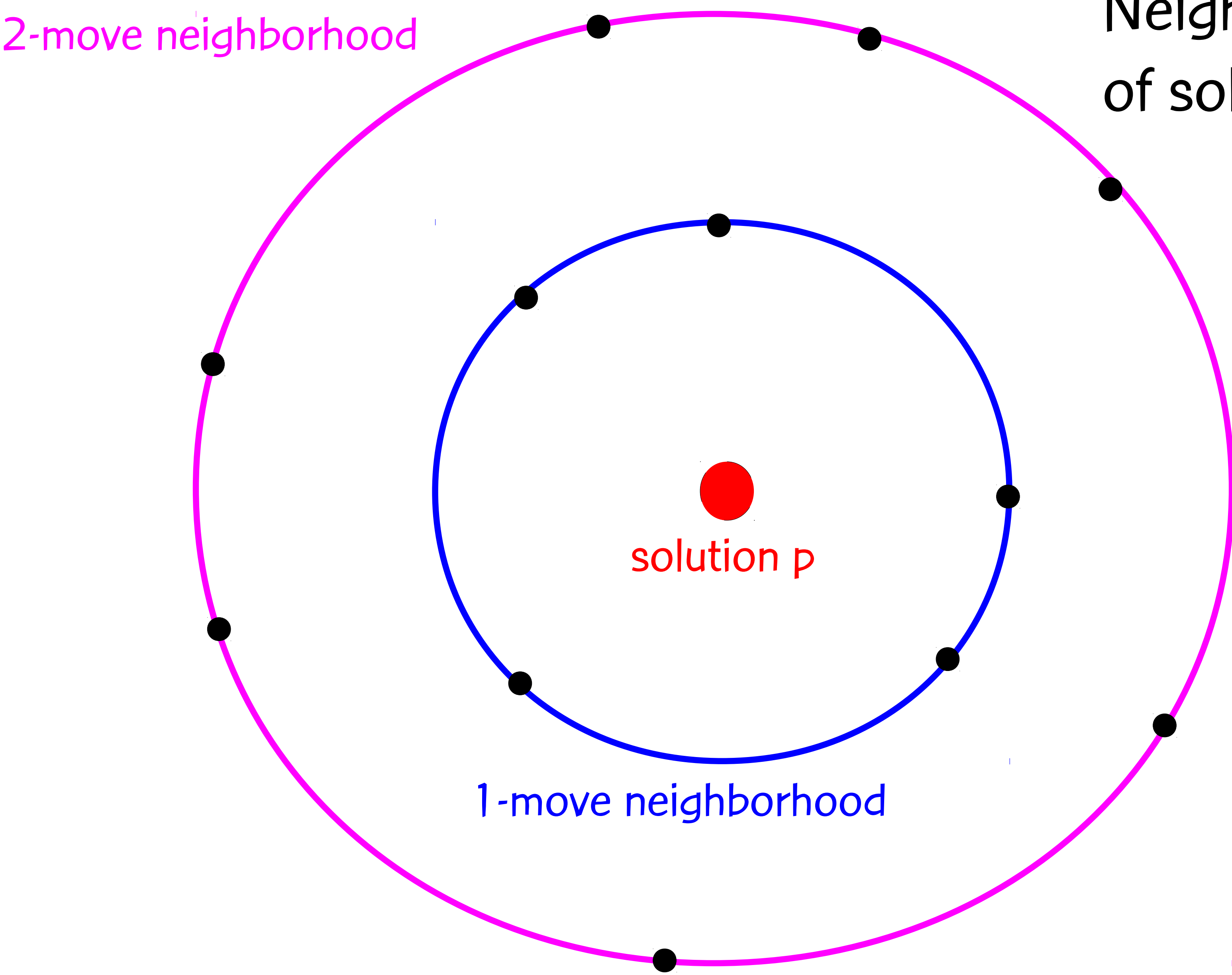


2-move neighbor of p

Assignment representation



Neighborhood of solution p



Traditional local search approaches

Best improving approach:

Evaluate all 1-move and 2-move neighborhood solutions and select the best improving solution

First improving approach:

1: From solution p, to evaluate its 1-move neighbors until the first improving solution q is found.

2: If q does not exist, continue search in the 2-move neighborhood.

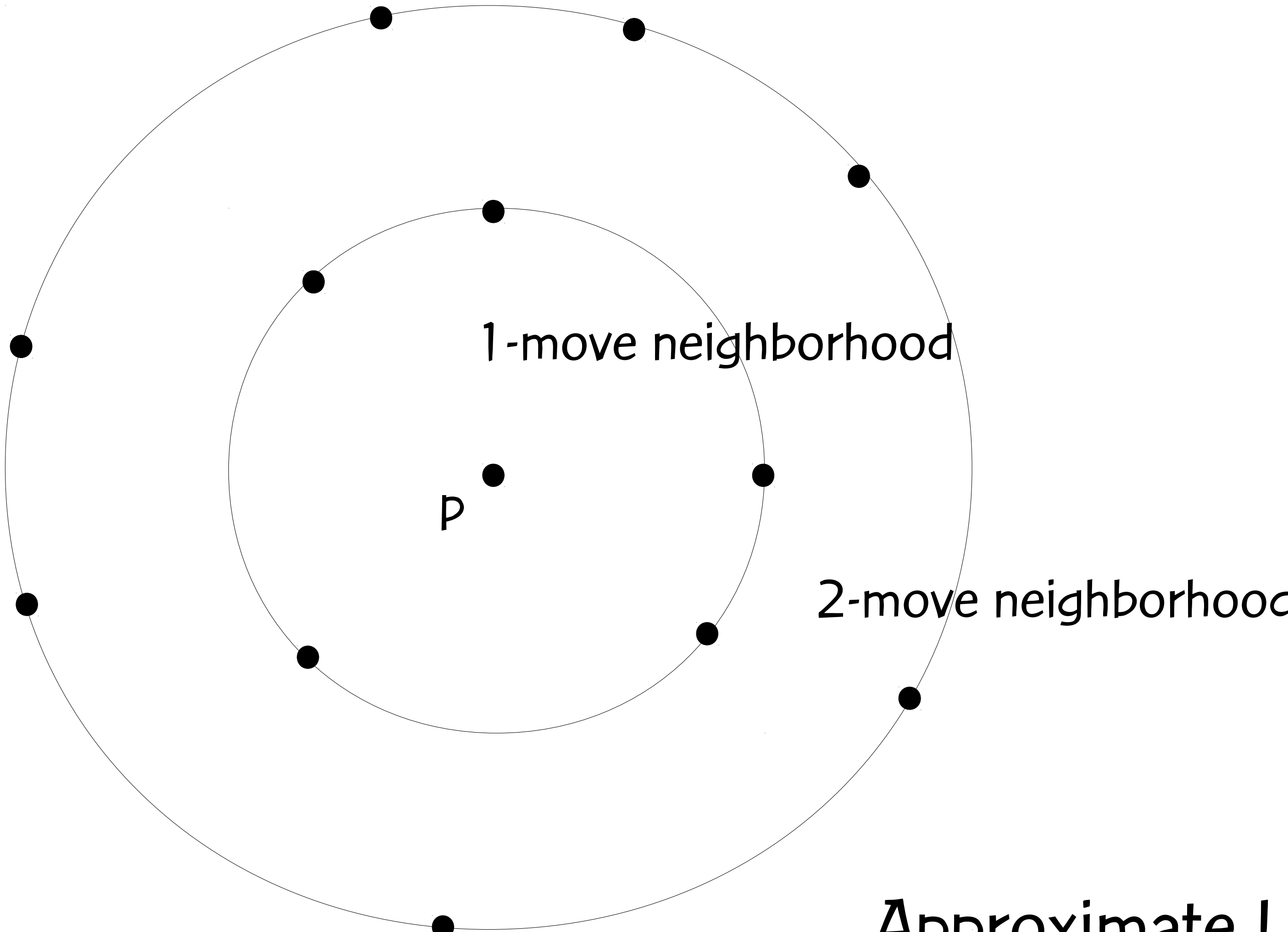
3: If q does not exist in the 2-move neighborhood, stop. Otherwise, assign $p = q$ and go to step 1.

Approximate local search

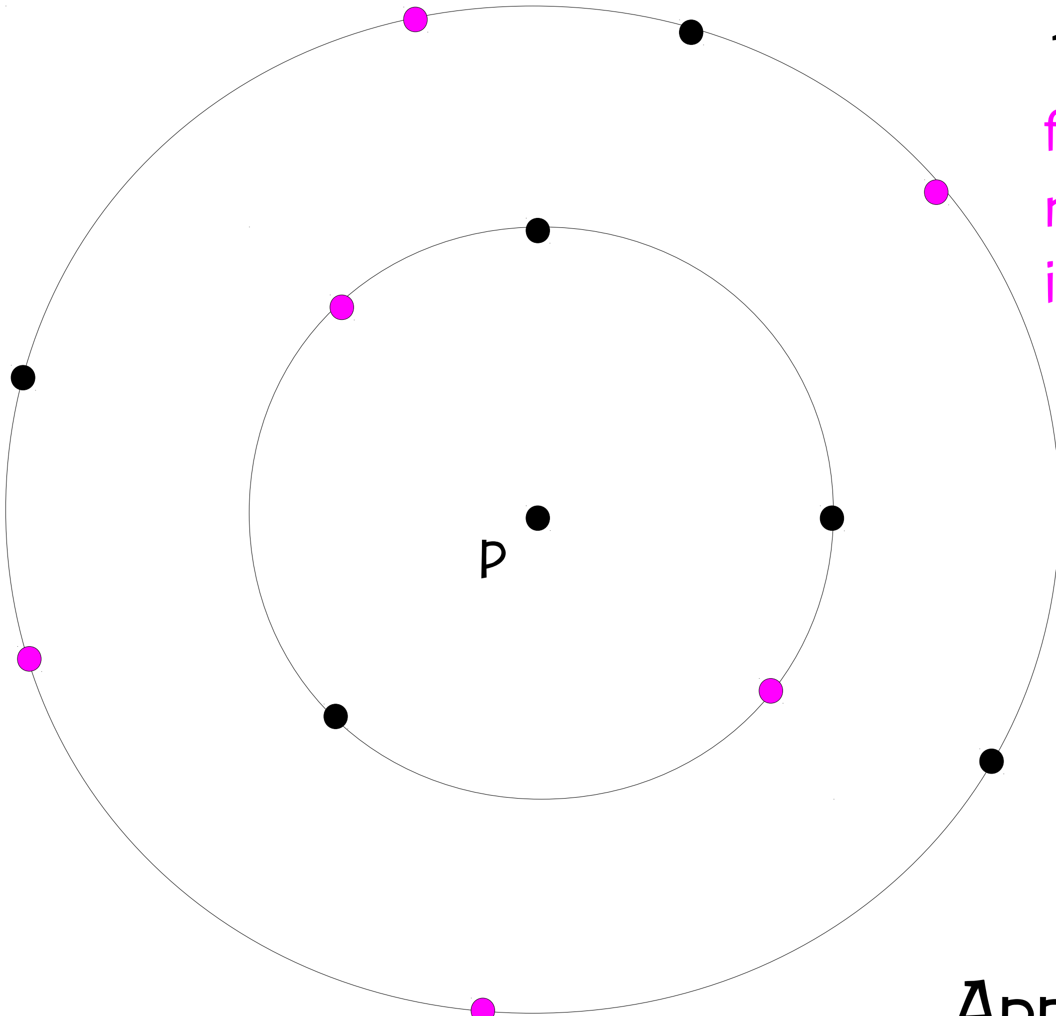
Tradeoff between best & first improvement: sample the neighborhood of solution p.

Neighborhoods can be very large for best improvement

Local search can take very long

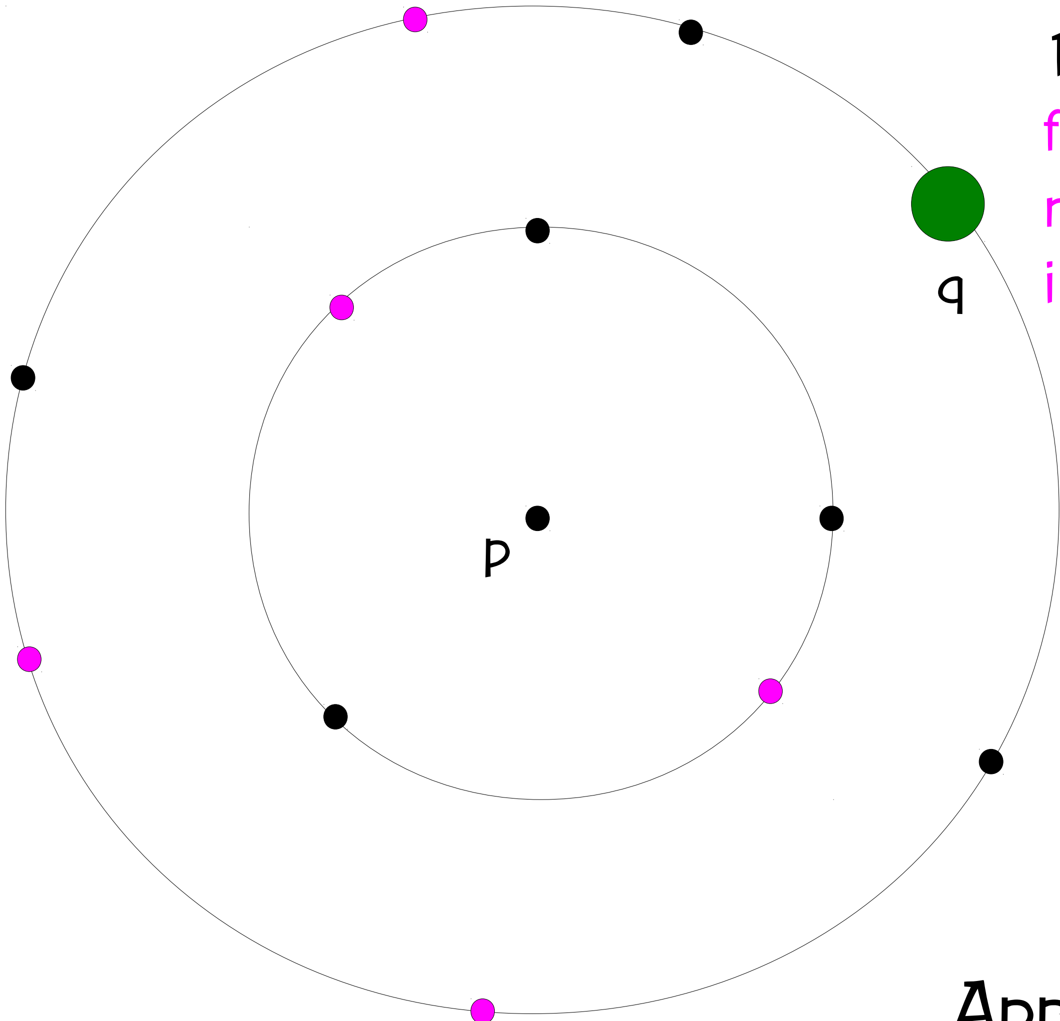


Approximate Local Search



1. Sample k improving solutions from 1-move and 2-move neighborhood of p and place them in an elite set E.

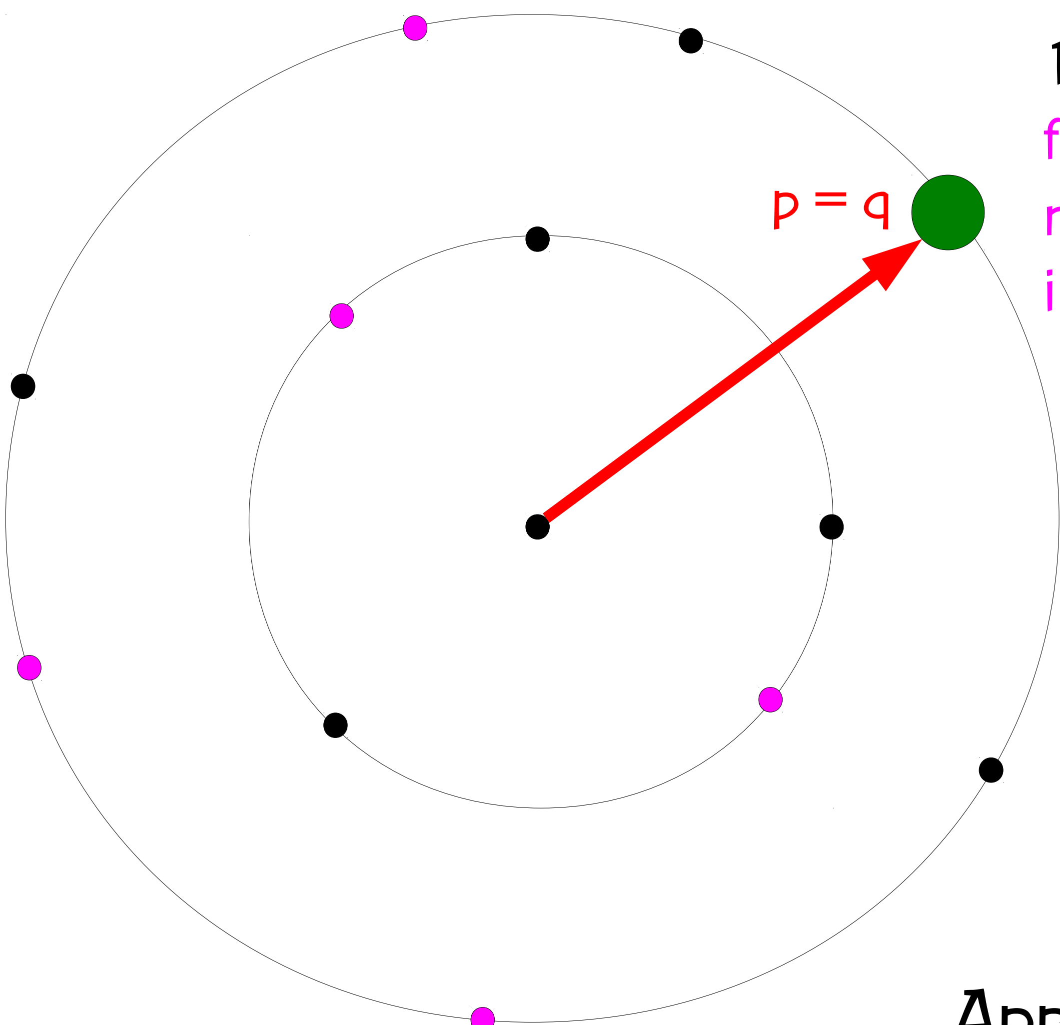
Approximate Local Search



1. Sample k improving solutions from 1-move and 2-move neighborhood of p and place them in an elite set E.

2. Select the best solution **q** from elite set E.

Approximate Local Search



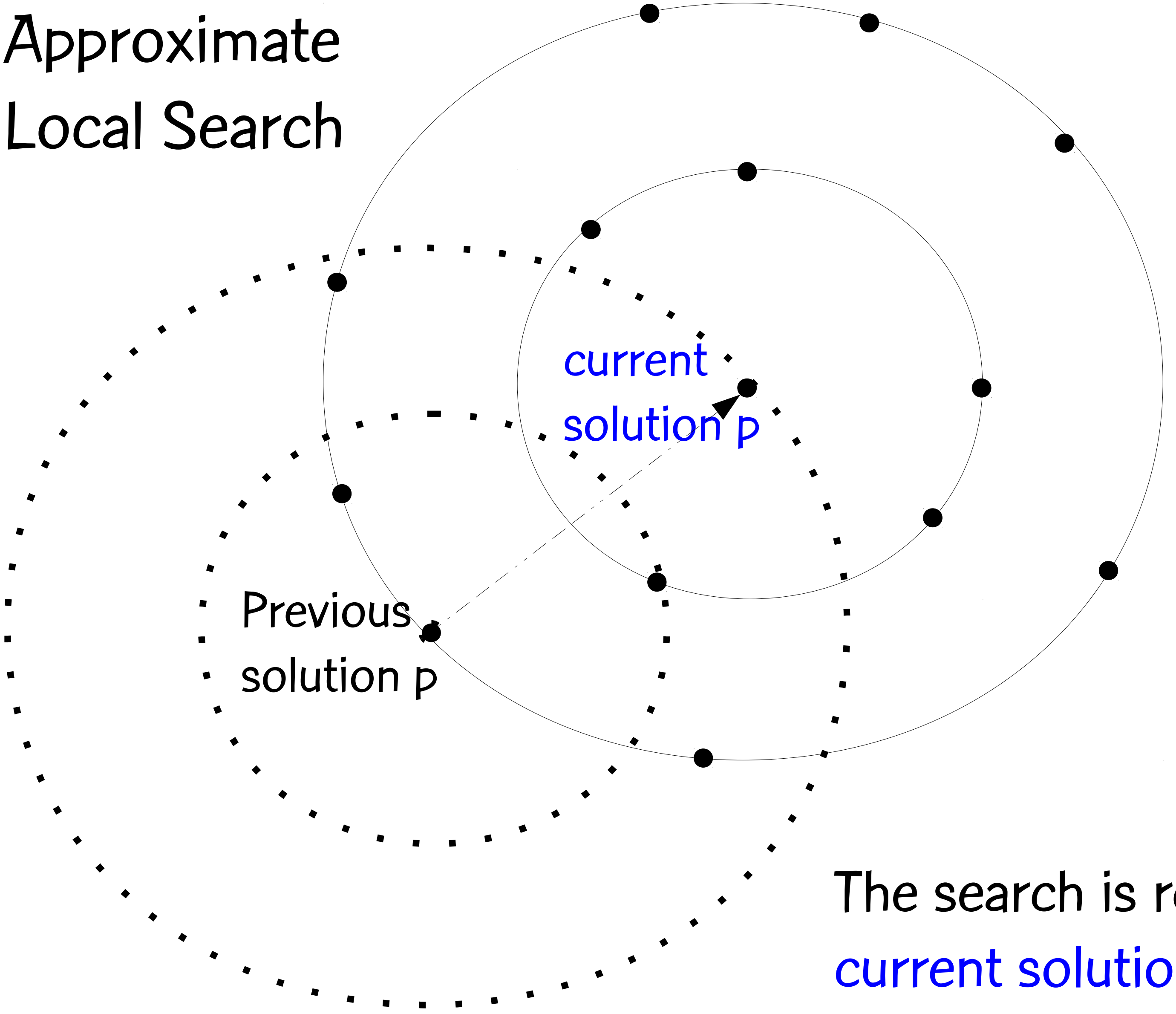
1. Sample k improving solutions from 1-move and 2-move neighborhood of p and place them in an elite set E.

2. Select the best solution **q** from elite set E.

3. Update p = q

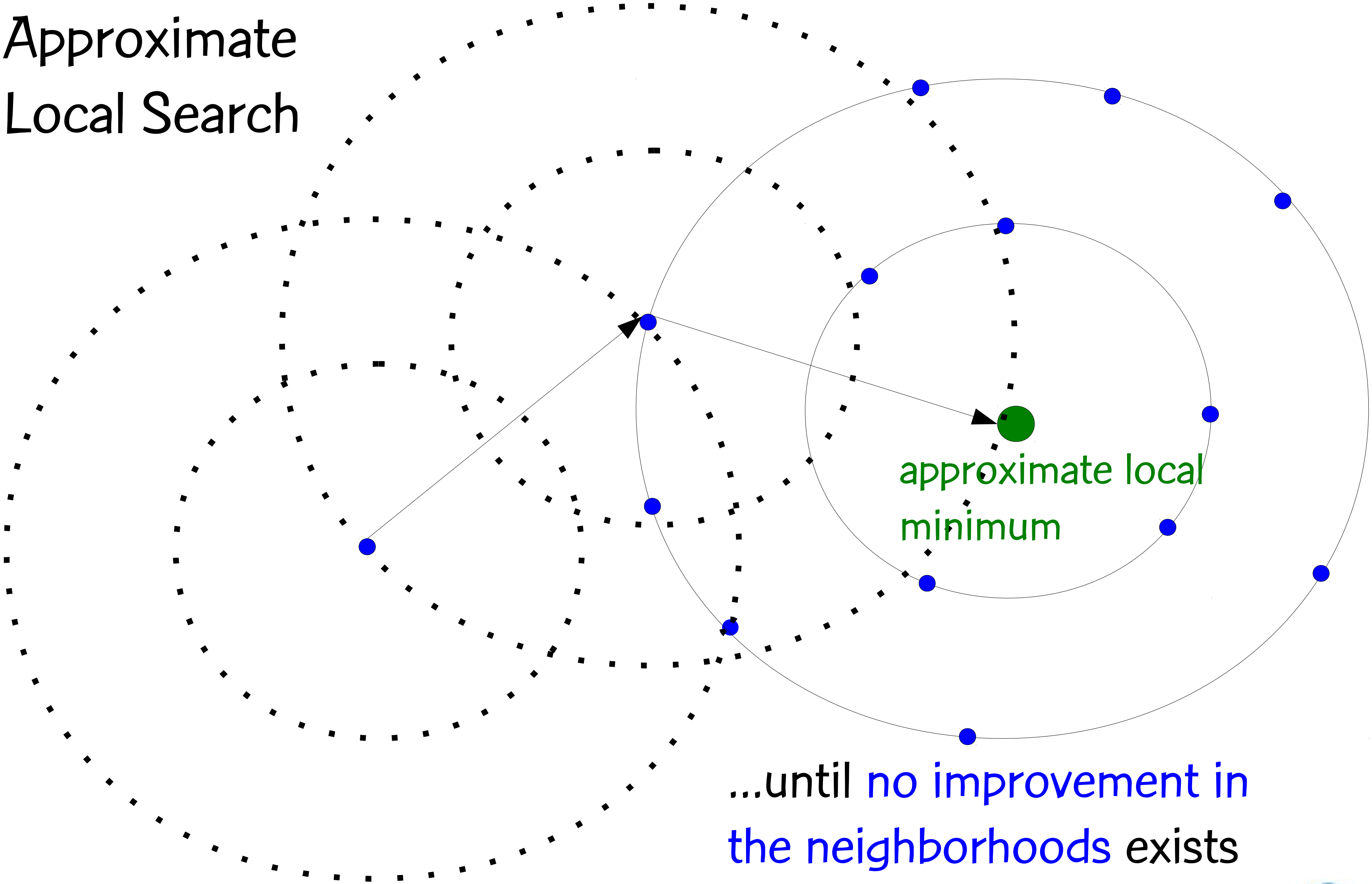
Approximate Local Search

Approximate Local Search



The search is repeated from current solution p until ....

Approximate Local Search

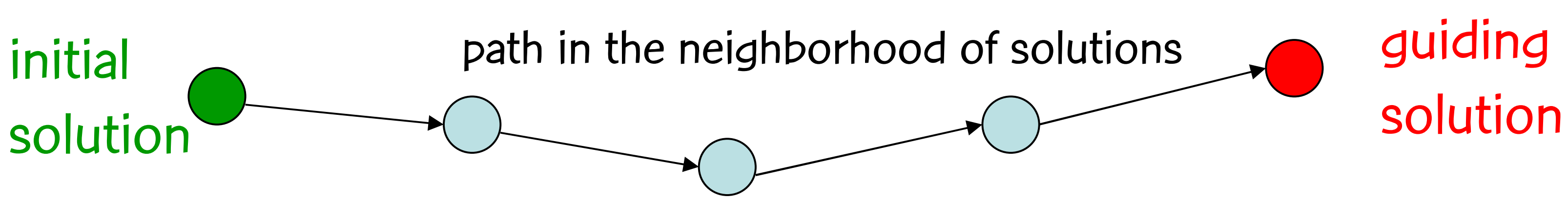


...until no improvement in the neighborhoods exists

Path-relinking for GQAP

Path-relinking (Glover, 1996)

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:

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:

Path-relinking

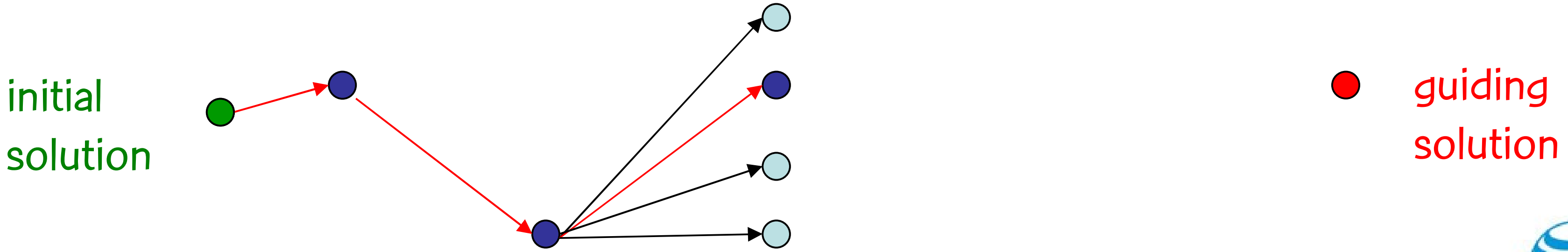
Path is generated by selecting moves that introduce in the initial solution attributes of the guiding solution.

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

Path-relinking

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

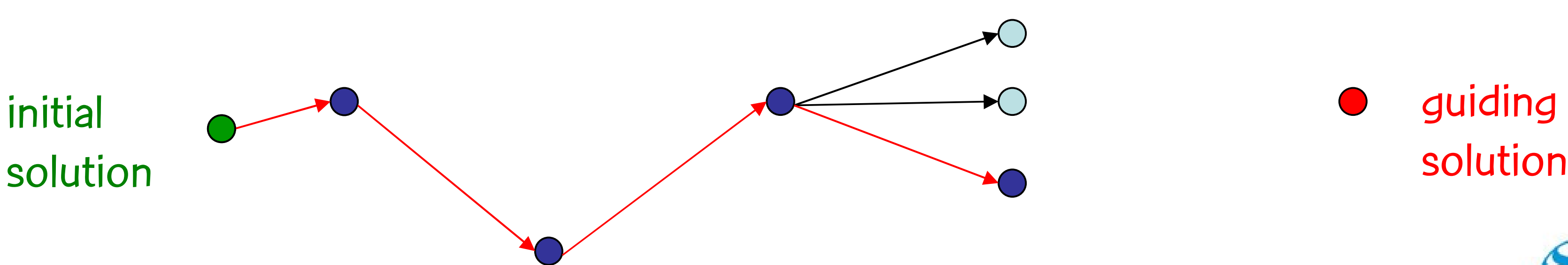
At each step, all moves that incorporate attributes of the guiding solution are evaluated and the best move is selected:



Path-relinking

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

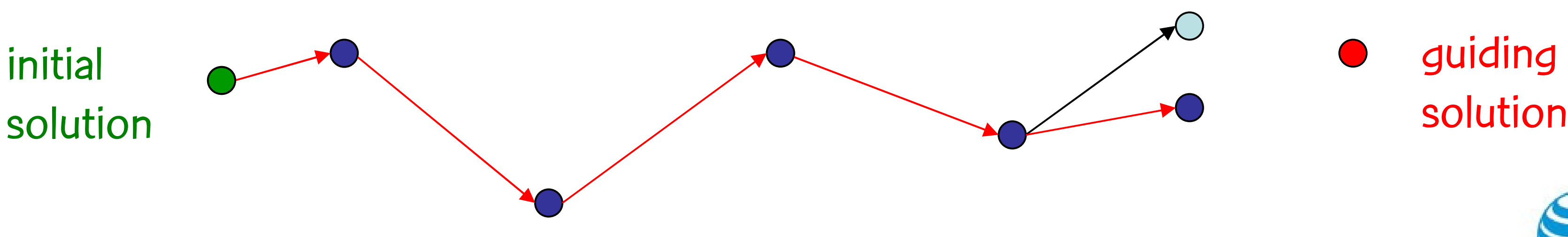
At each step, all moves that incorporate attributes of the guiding solution are evaluated and the best move is selected:



Path-relinking

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

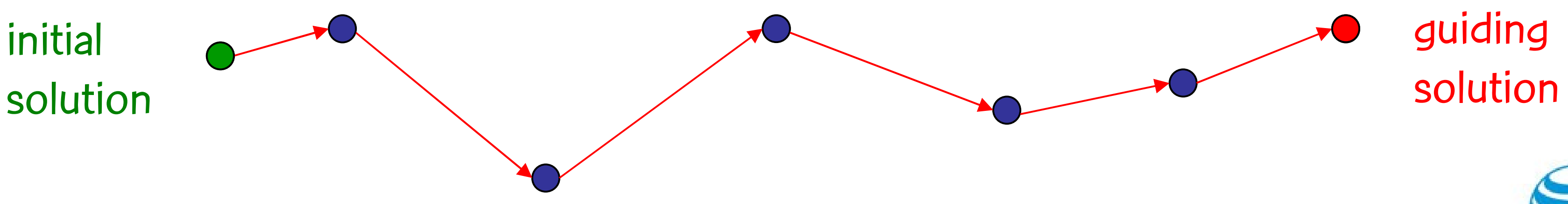
At each step, all moves that incorporate attributes of the guiding solution are evaluated and the best move is selected:

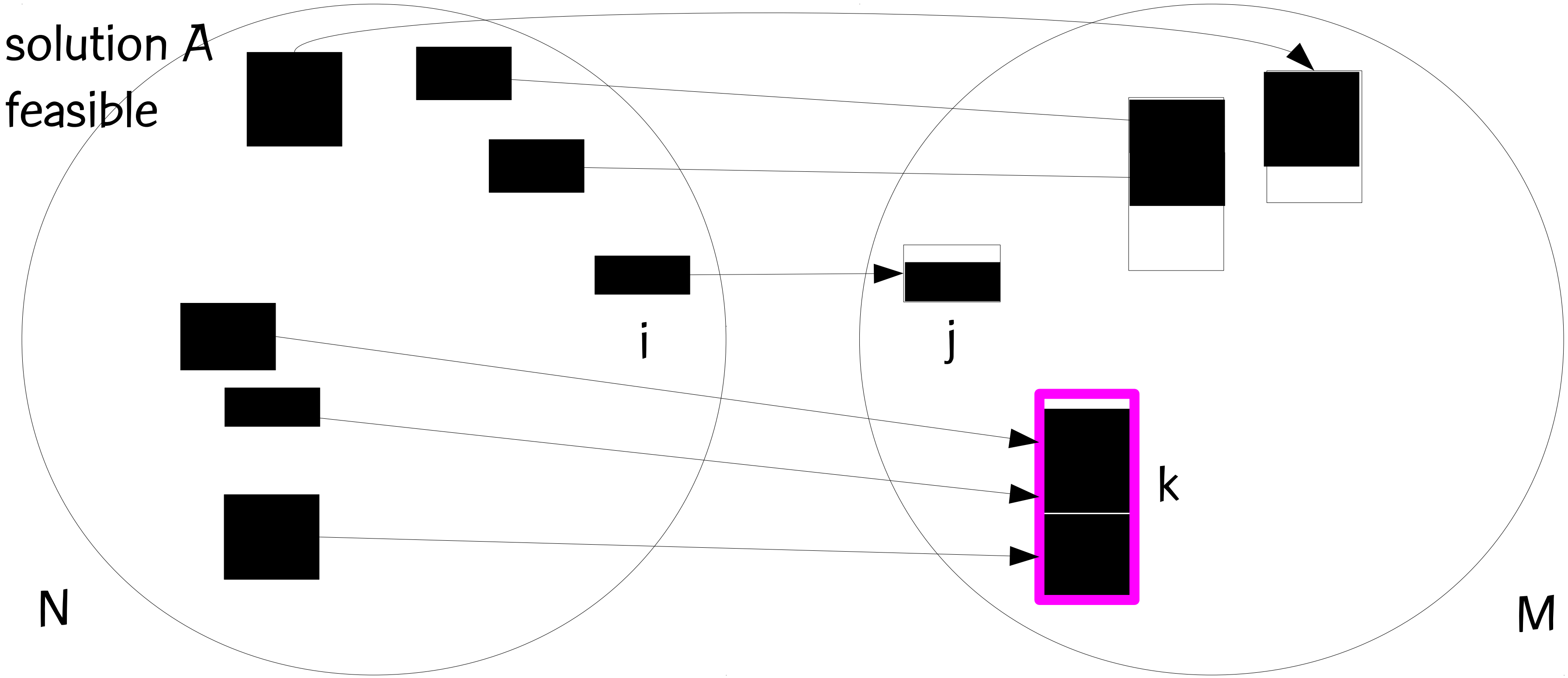


Path-relinking

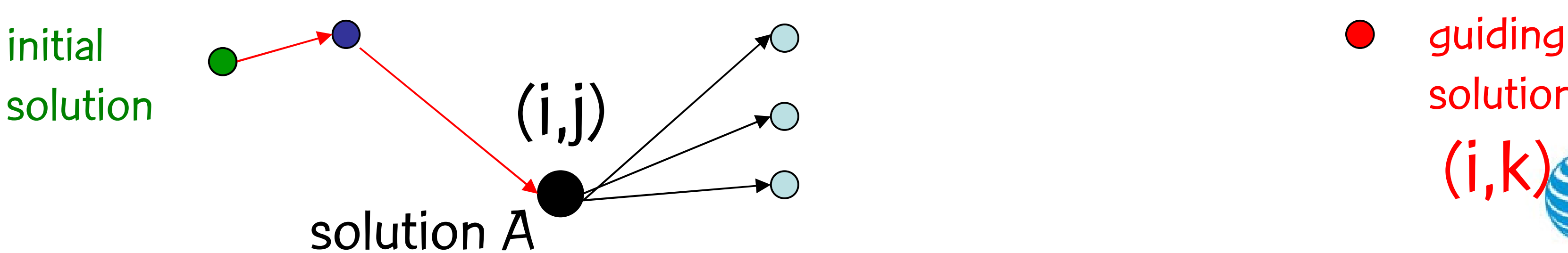
Path is generated by selecting moves that introduce in the initial solution attributes of the guiding solution.

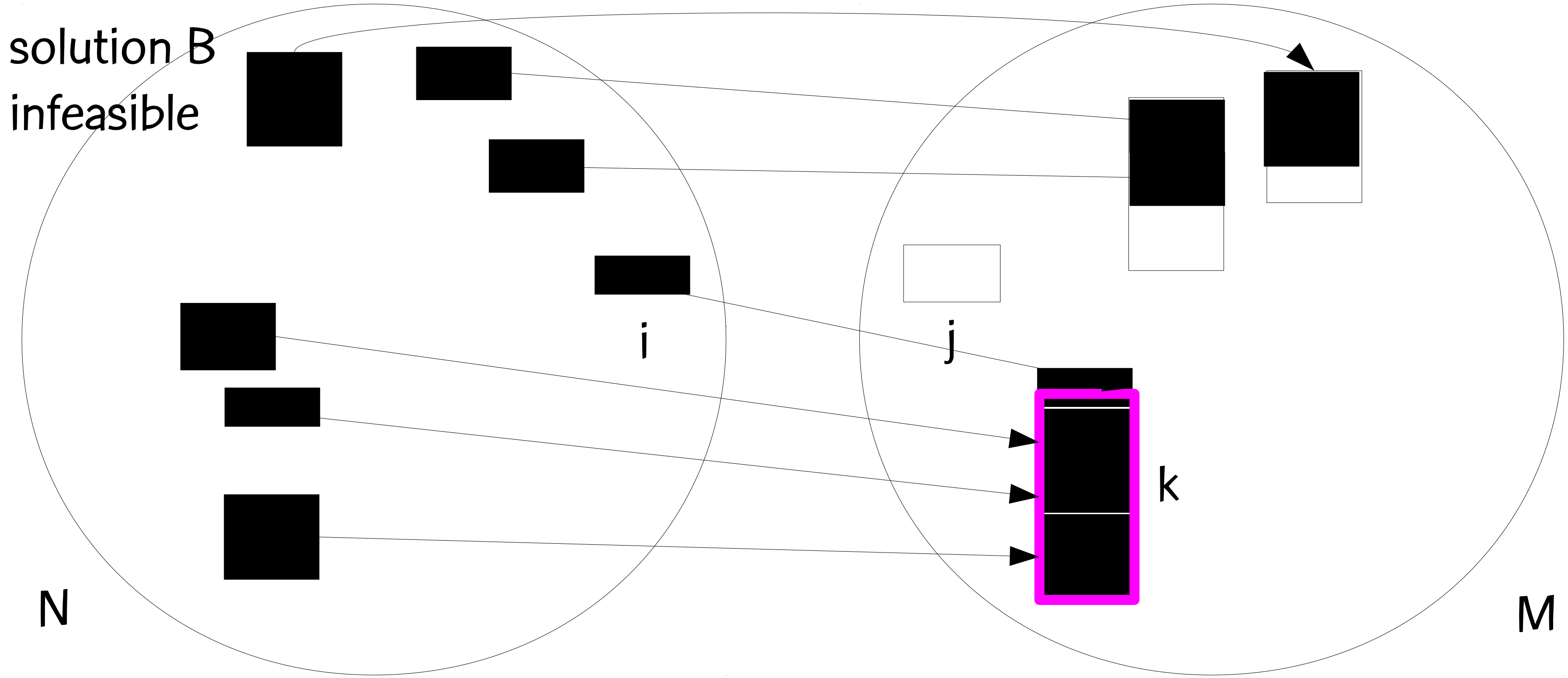
At each step, all moves that incorporate attributes of the guiding solution are evaluated and the best move is selected:



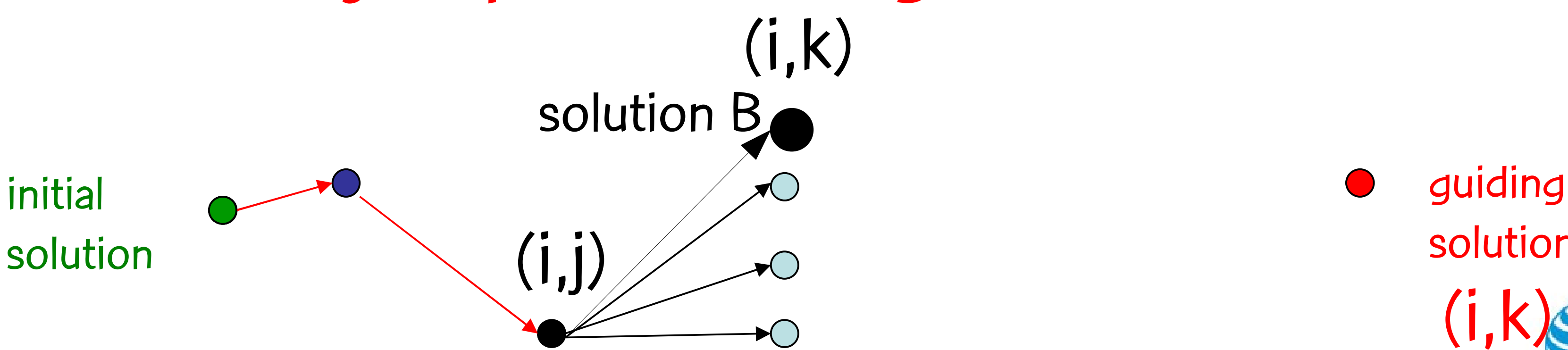


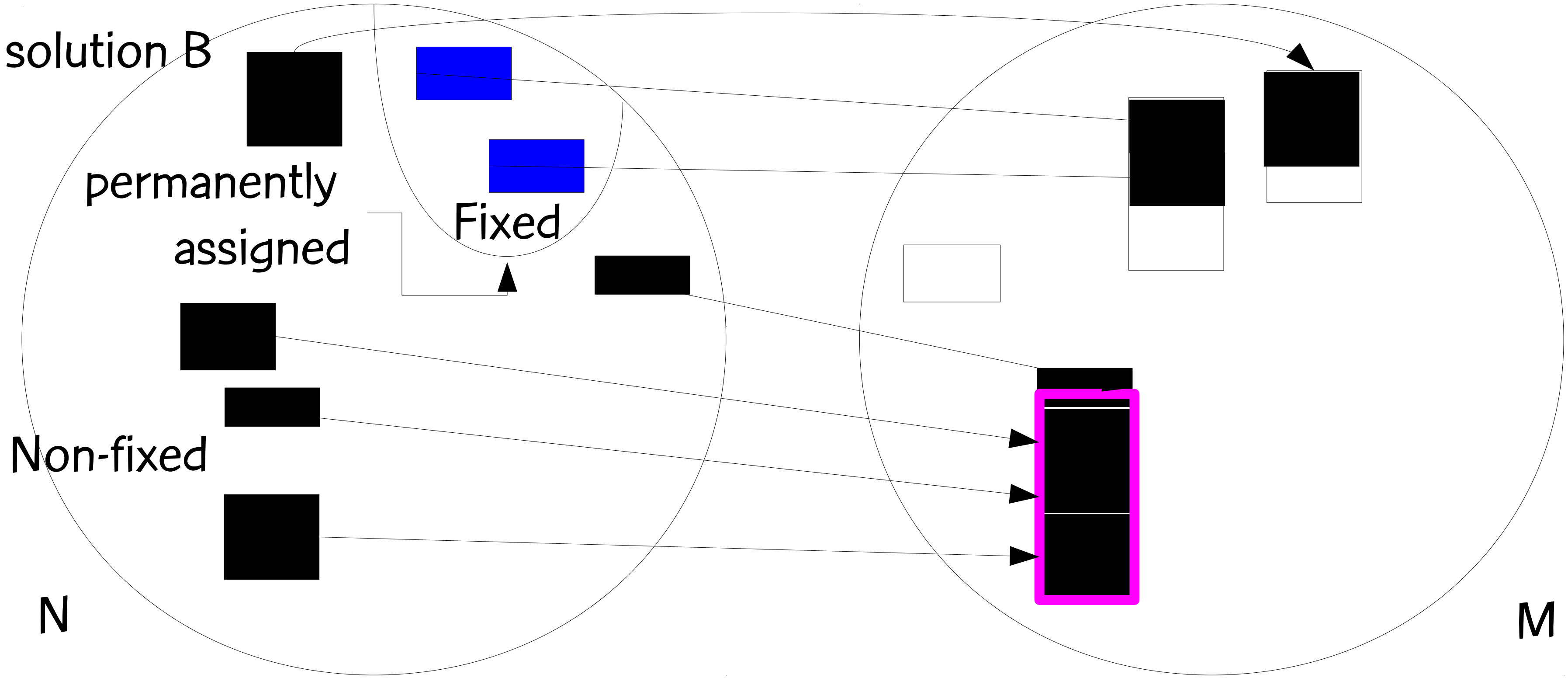
Infeasibility in path-relinking for GQAP



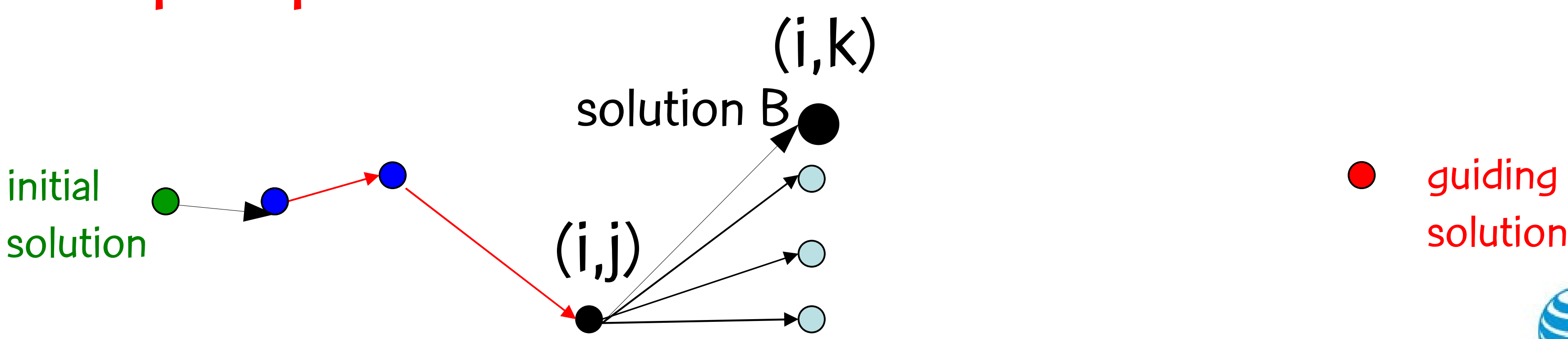


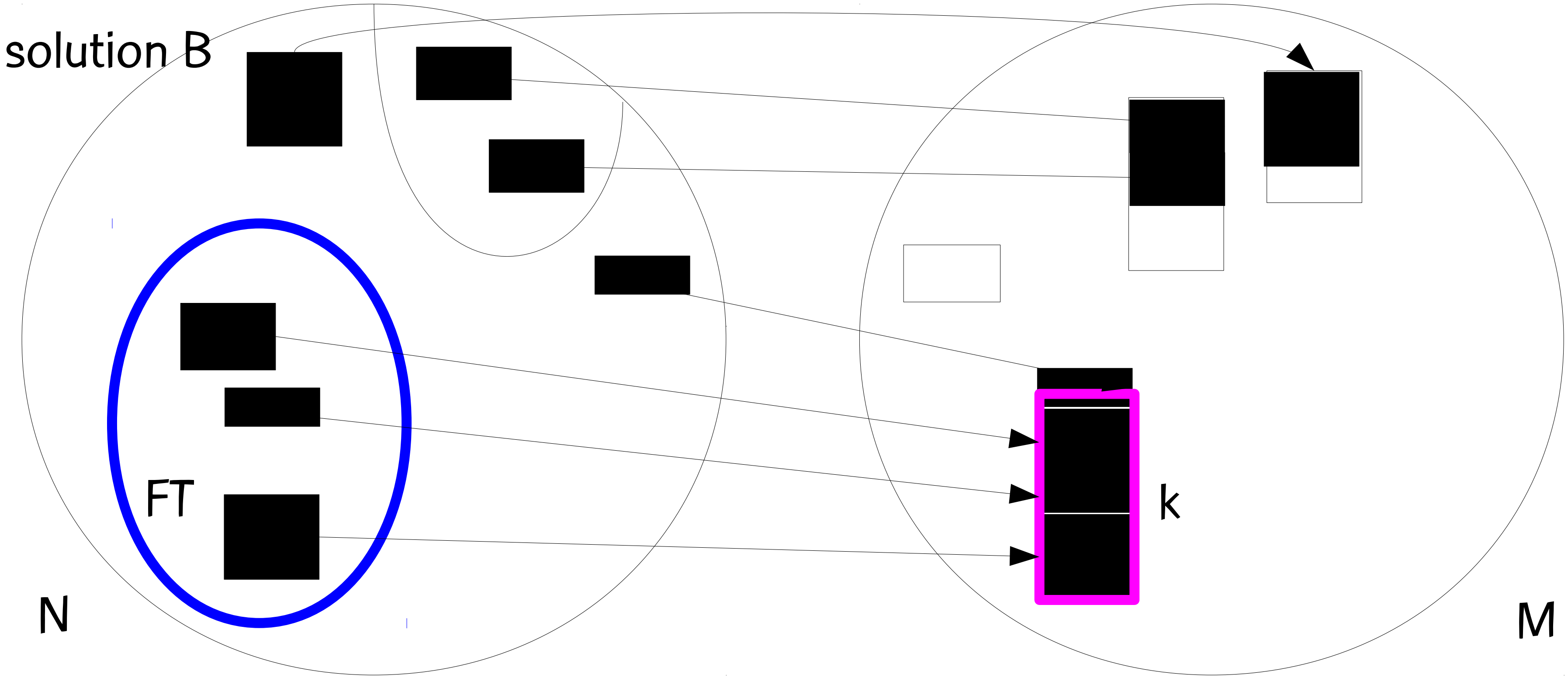
Infeasibility in path-relinking for GQAP



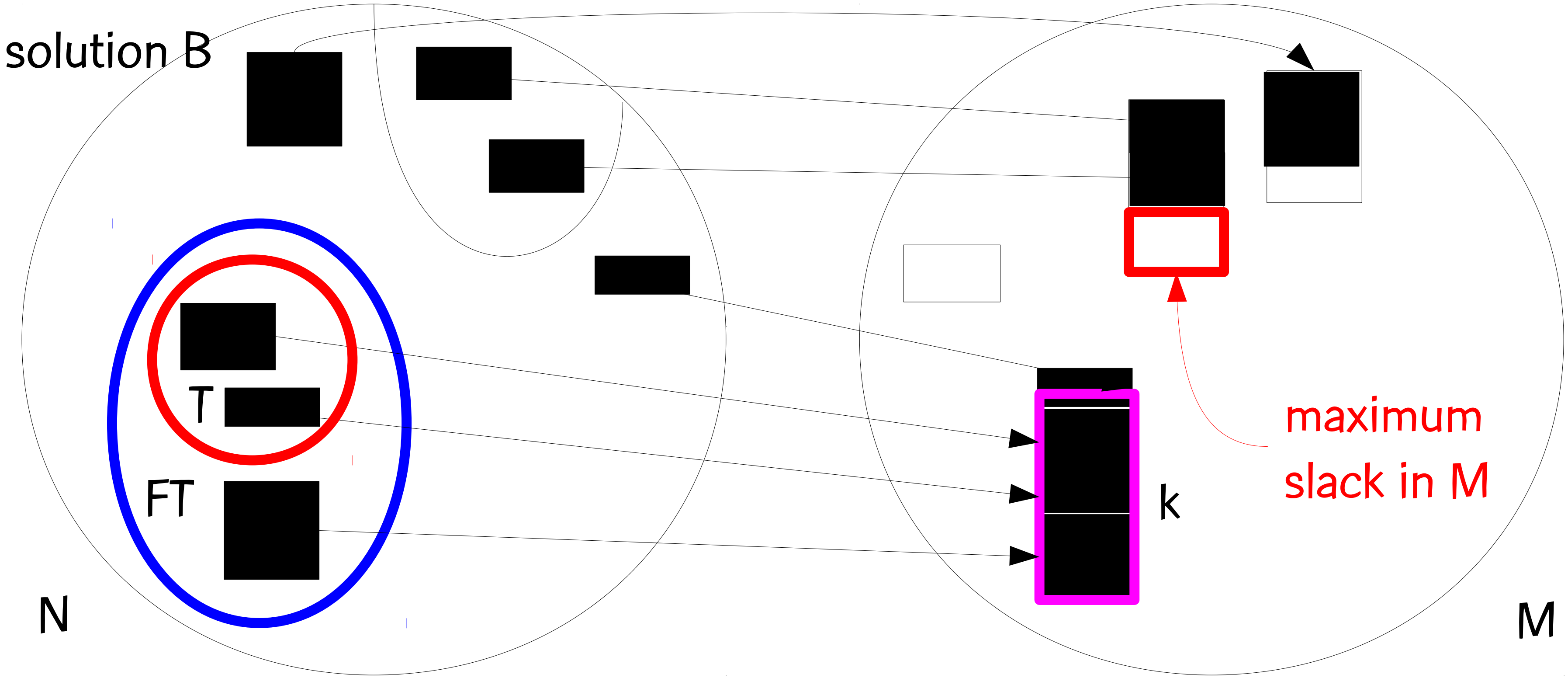


Repair procedure

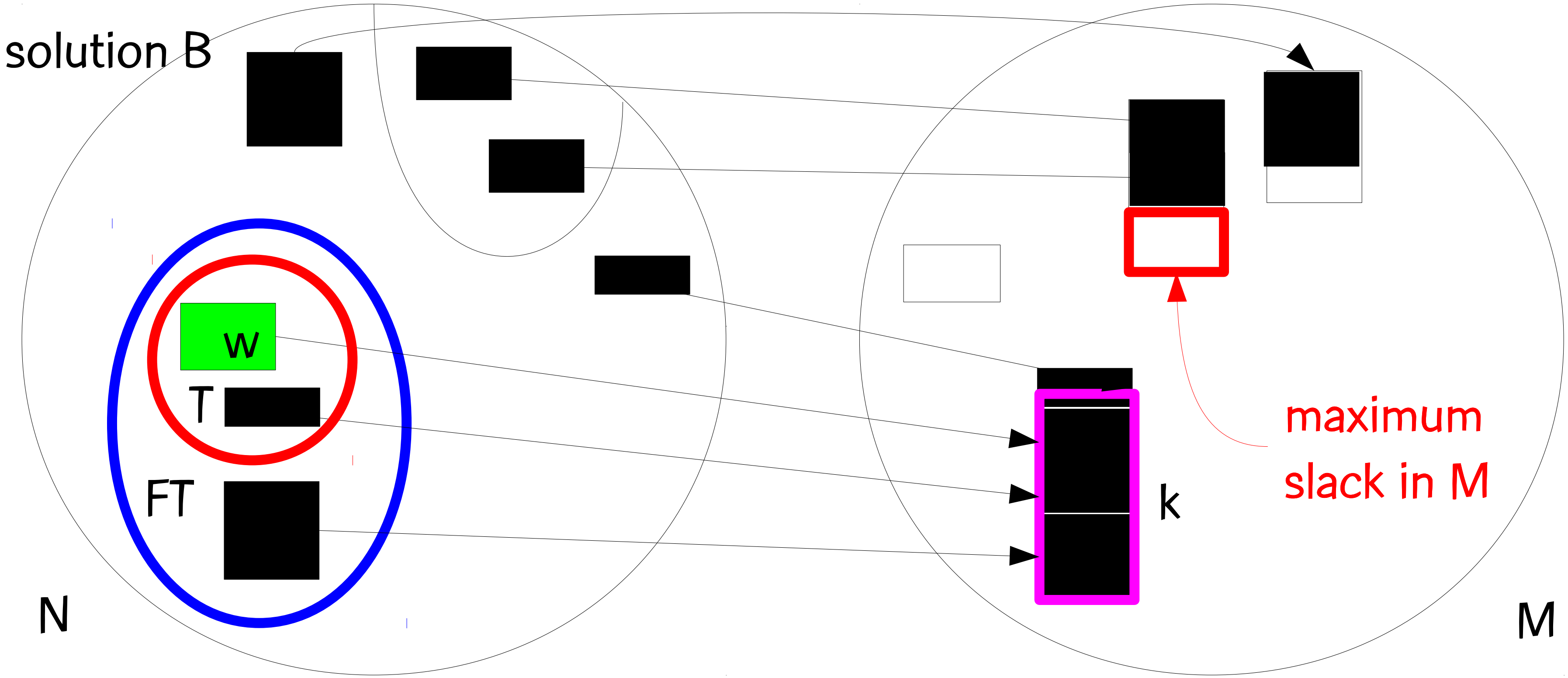




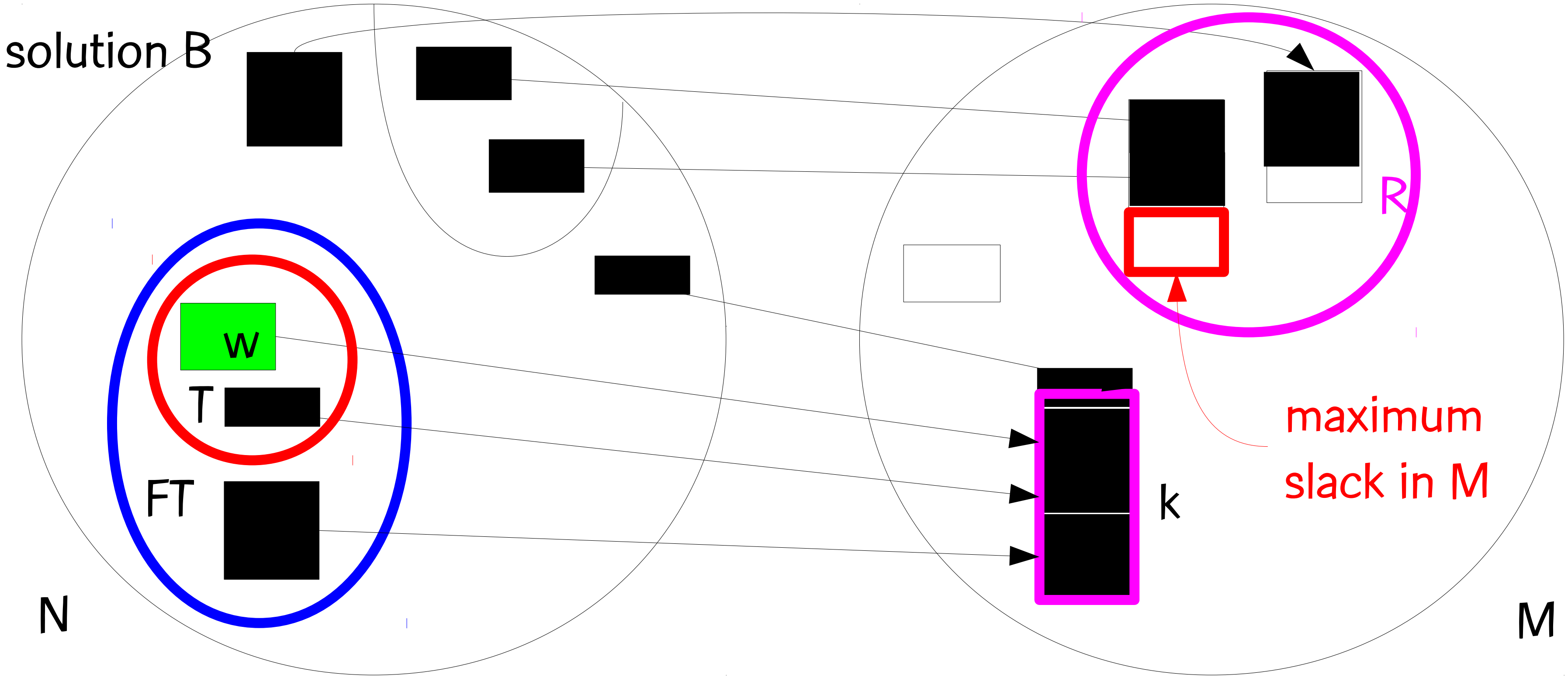
1. Set FT ⊆ non-Fixed: all facilities in solution B assigned to location k



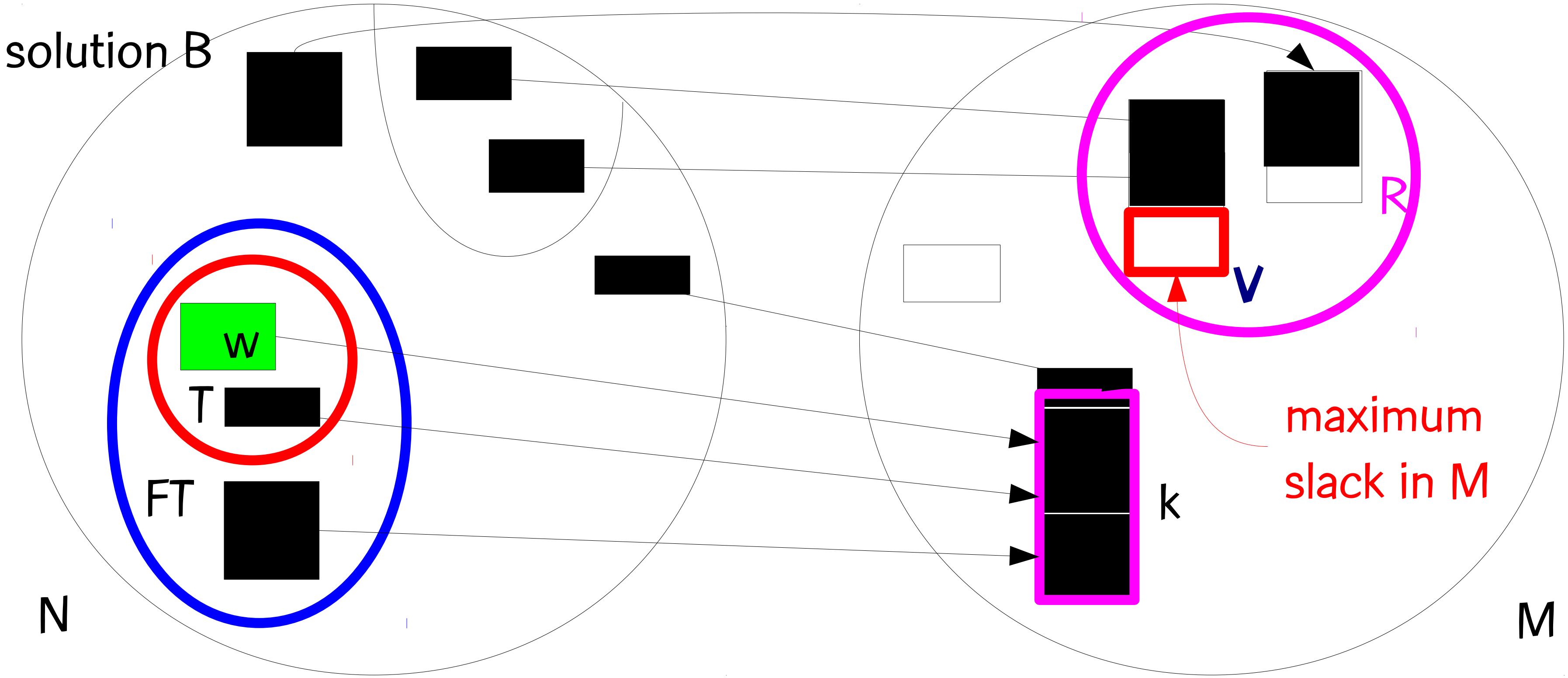
1. Set FT $\subseteq$ non-Fixed: all facilities in solution B assigned to location k
2. Set T $\subseteq$ FT: all facilities in B with demand $\leq$ maximum slack in M



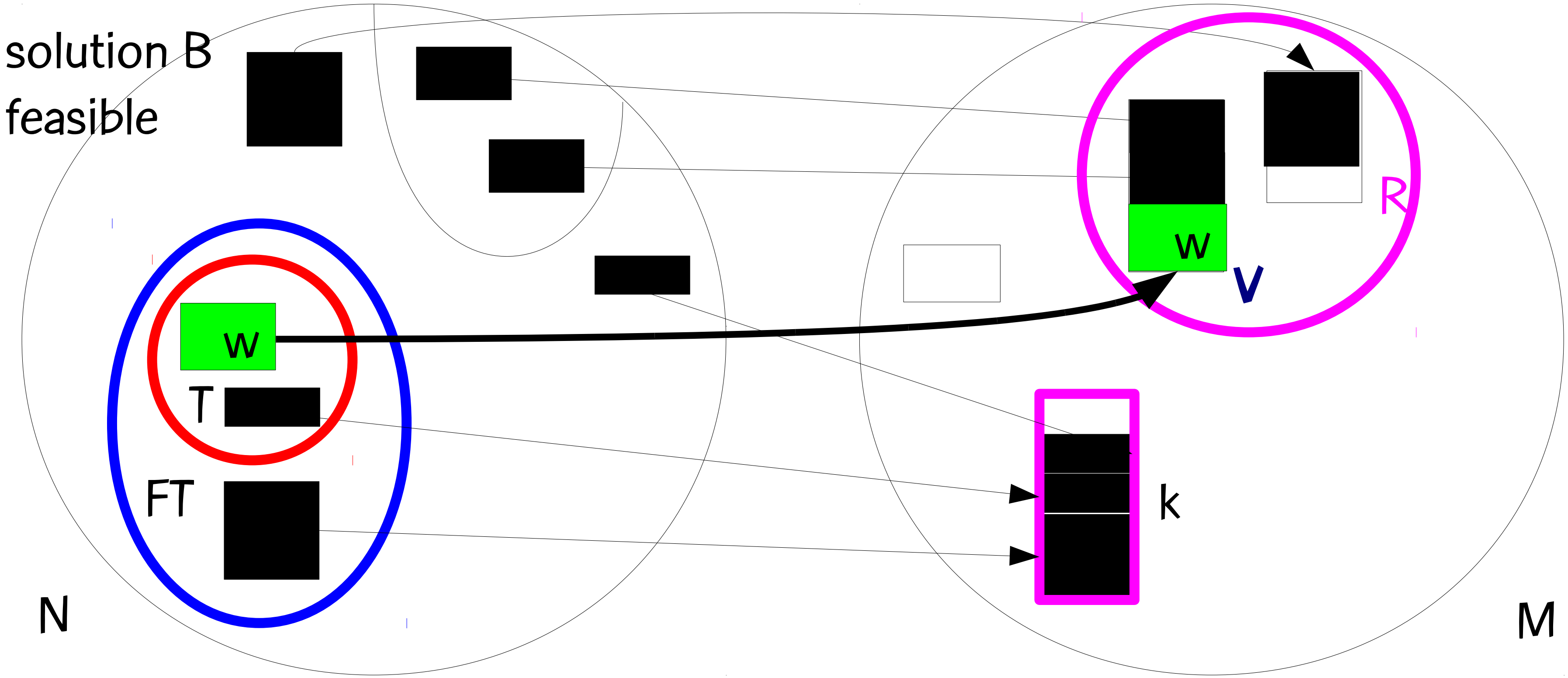
1. Set FT $\subseteq$ non-Fixed: all facilities in solution B assigned to location k
2. Set T $\subseteq$ FT: all facilities in B with demand $\leq$ maximum slack in M
3. Randomly select a facility w $\in$ T favoring those with higher demand



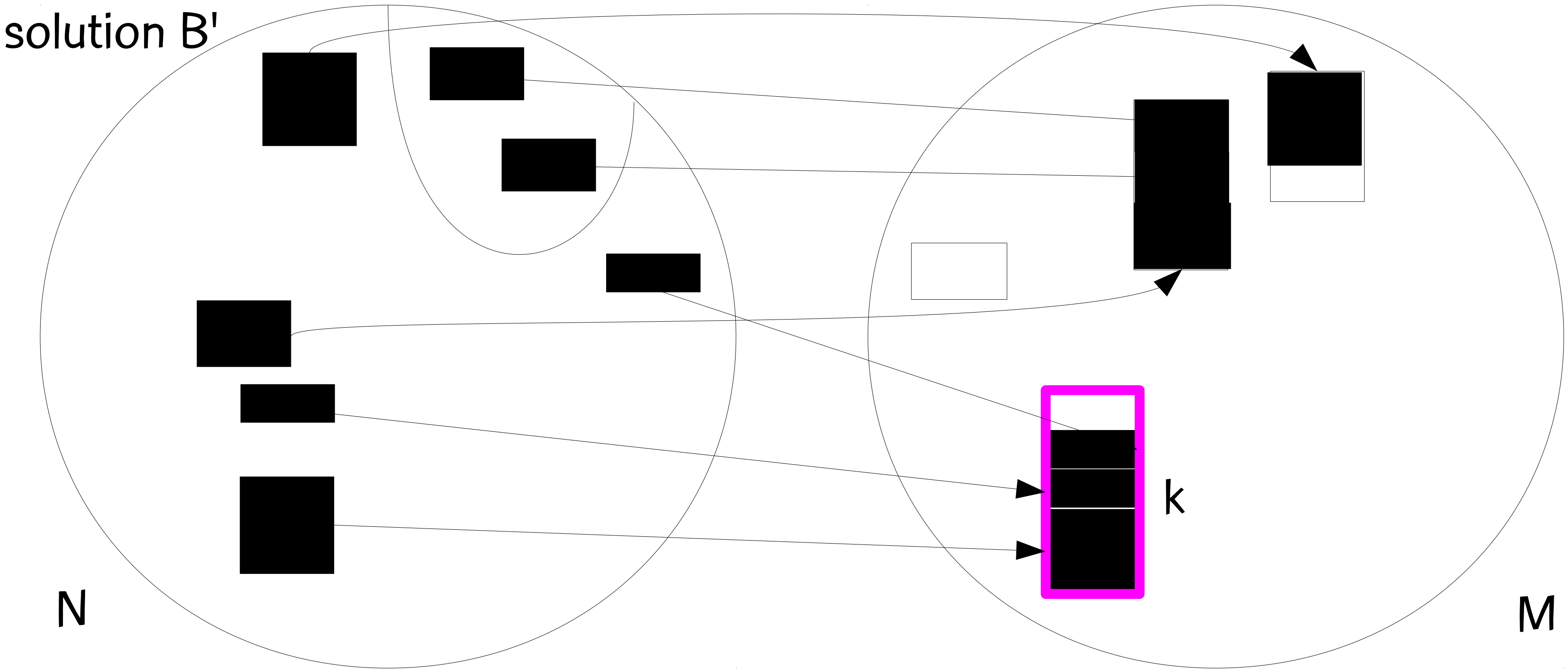
1. Set FT $\subseteq$ non-Fixed: all facilities in solution B assigned to location k
2. Set T $\subseteq$ FT: all facilities in B with demand $\leq$ maximum slack in M
3. Randomly select a facility w $\in$ T favoring those with higher demand
4. Set R $\subseteq$ M: all locations having slack $\geq$ demand of facility w

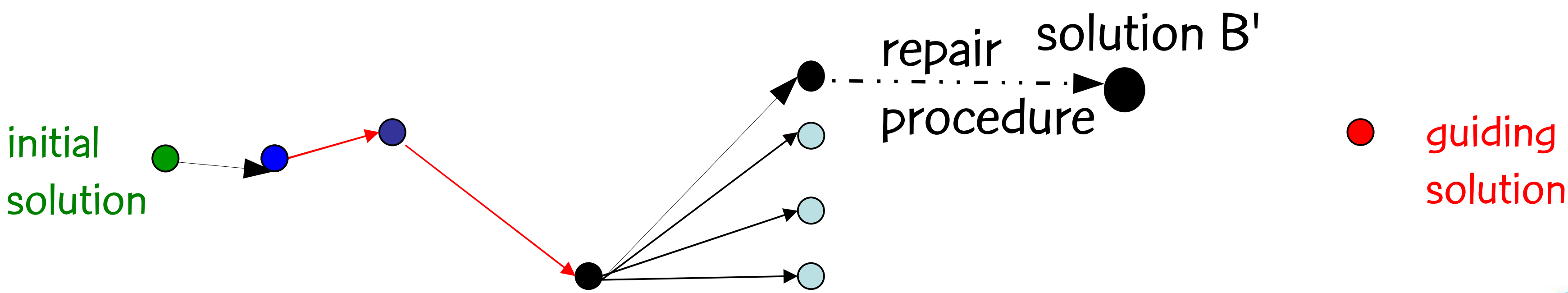


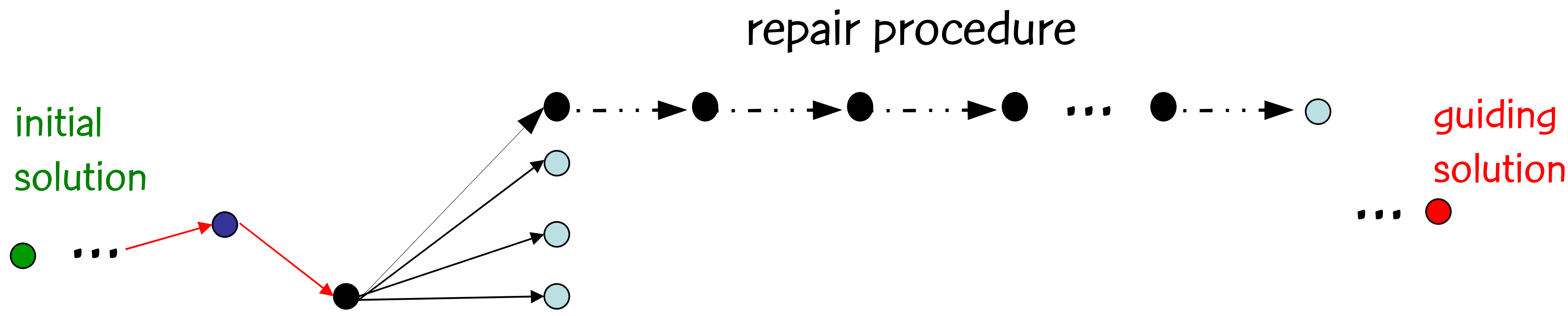
1. Set FT $\subseteq$ non-Fixed: all facilities in solution B assigned to location k
2. Set T $\subseteq$ FT: all facilities in B with demand $\leq$ maximum slack in M
3. Randomly select a facility w $\in$ T favoring those with higher demand
4. Set R $\subseteq$ M: all locations having slack $\geq$ demand of facility w
5. Randomly select a location v $\in$ R (equal probability)



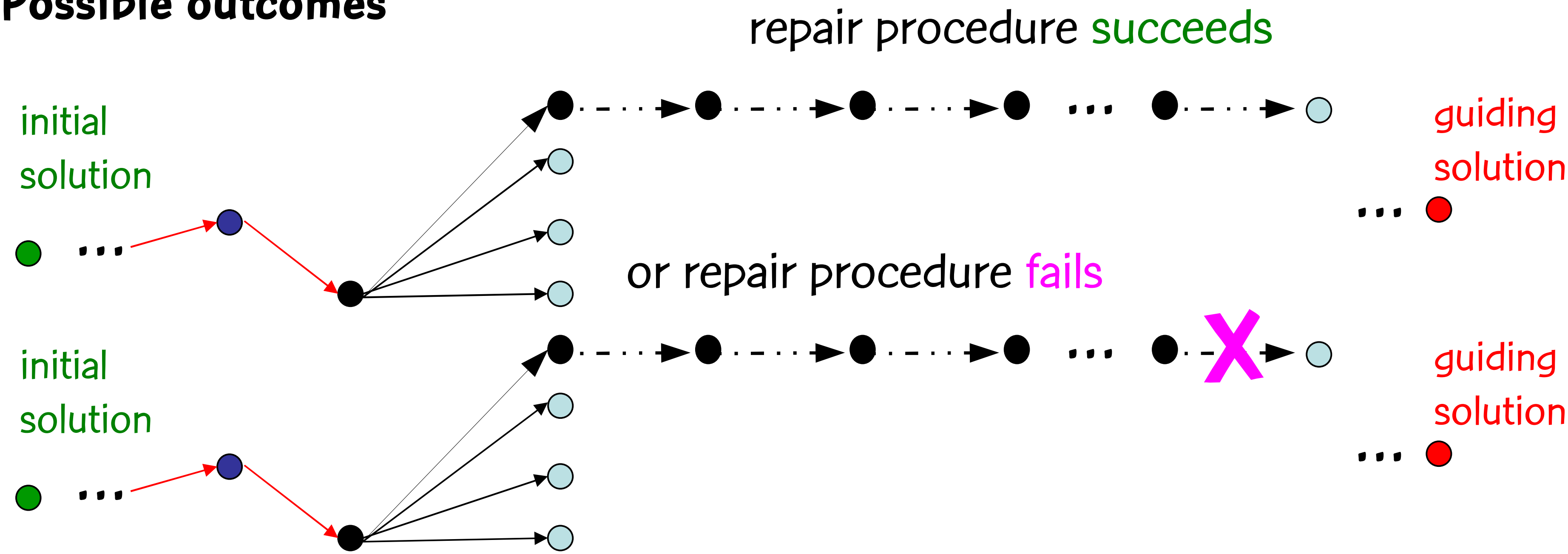
1. Set FT ⊆ non-Fixed: all facilities in solution B assigned to location k
2. Set T ⊆ FT: all facilities in B with demand ≤ maximum slack in M
3. Randomly select a facility w ∈ T favoring those with higher demand
4. Set R ⊆ M: all locations having slack ≥ demand of facility w
5. Randomly select a location v ∈ R (equal probability)
6. Assign facility w to location v


solution B'
k
N
M

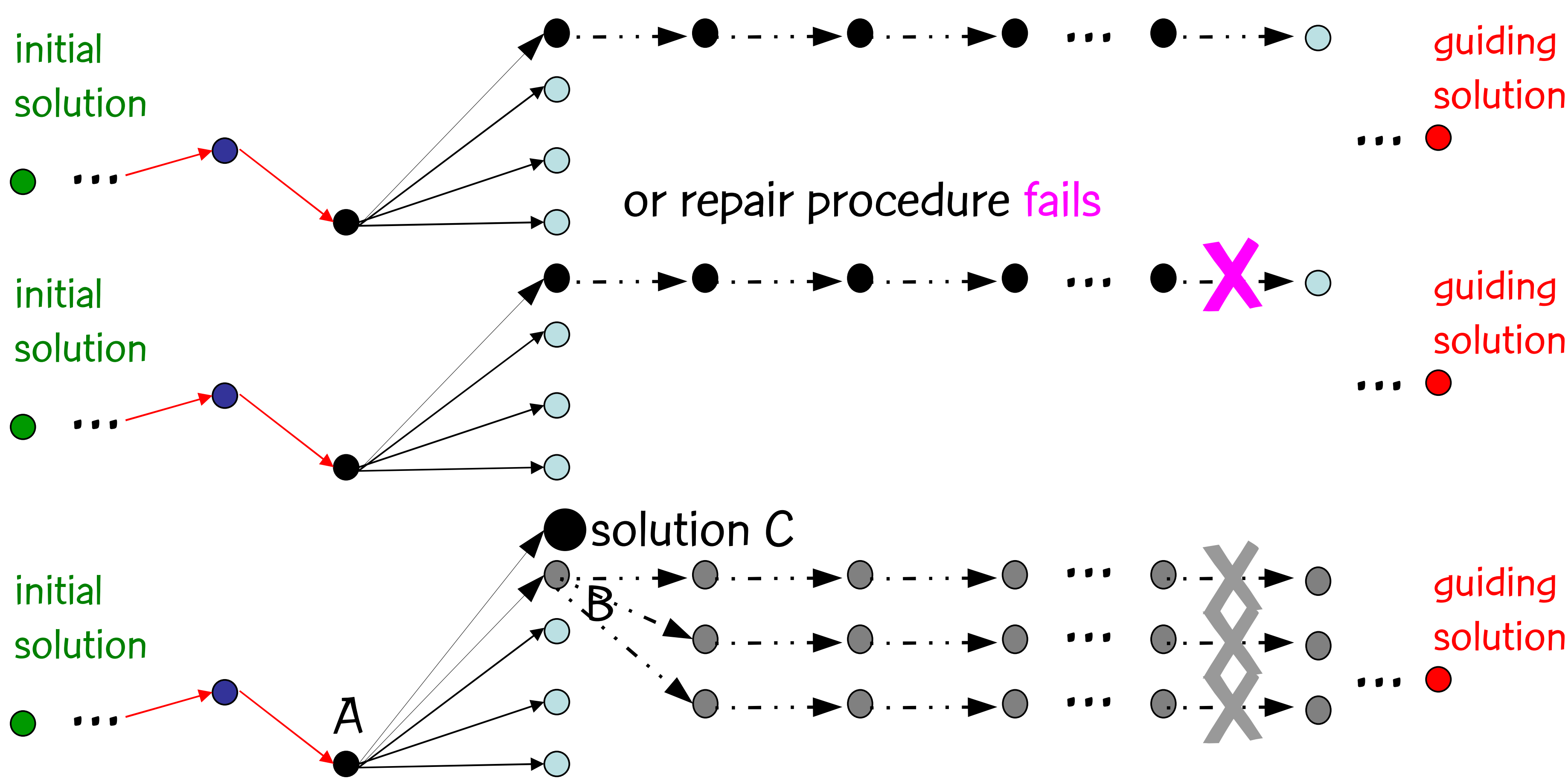

repair
procedure
solution B'
initial
solution
guiding
solution


repair procedure
initial
solution
guiding
solution

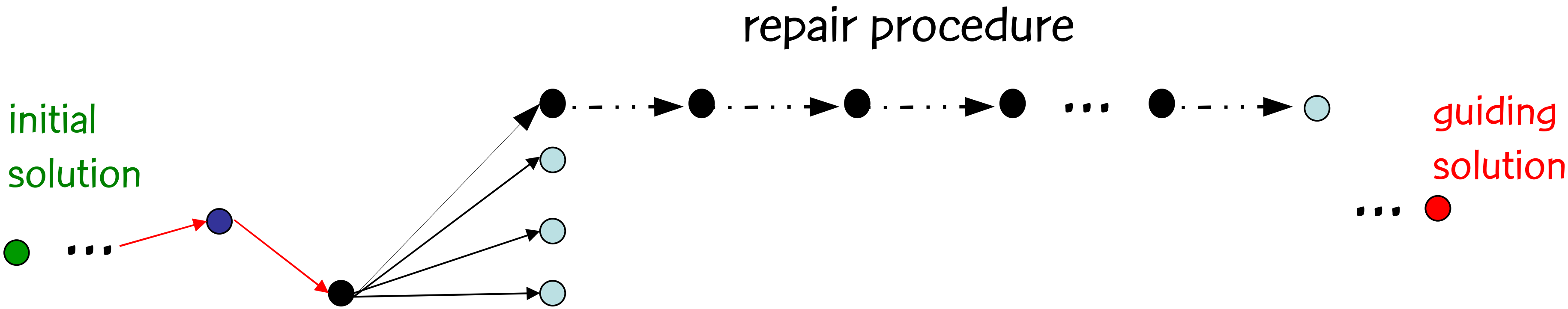
Possible outcomes



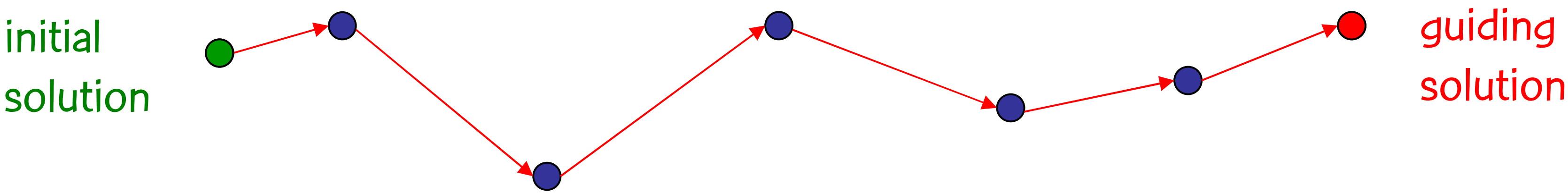
Possible outcomes

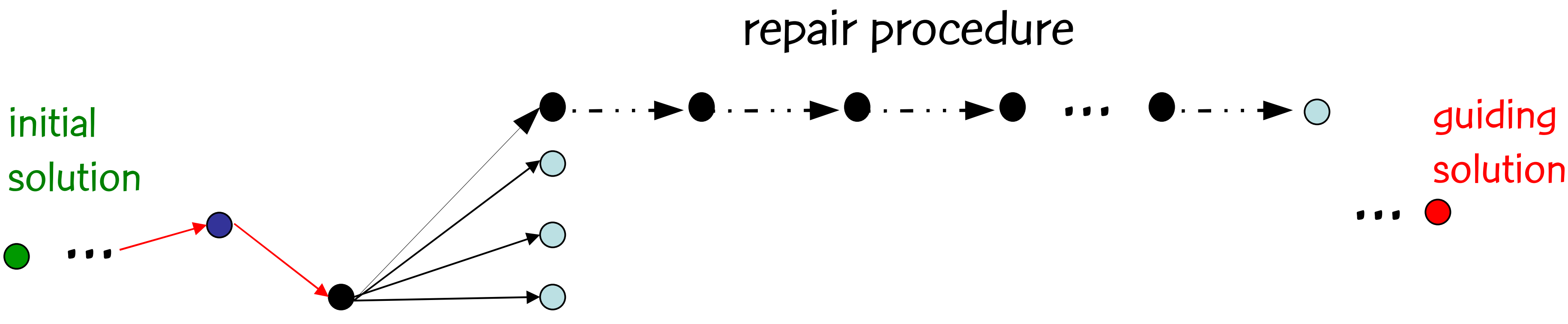


Repeat the repair procedure on solution B a maximum number of times. If a feasible solution is not found, discard B and move to solution C

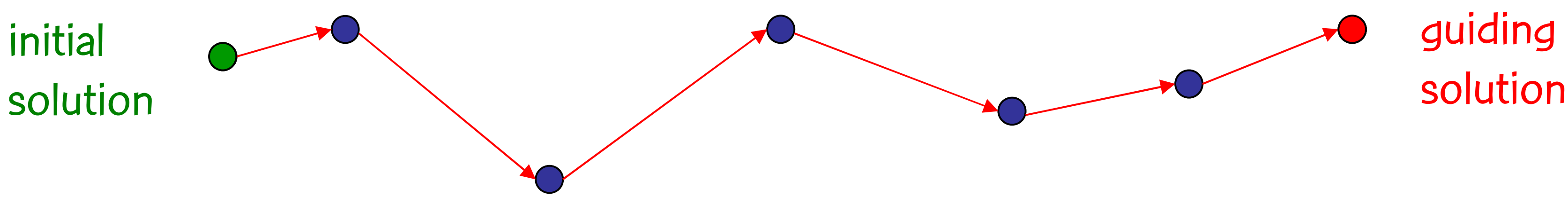


So, instead of a path with feasible solution in one single step ...

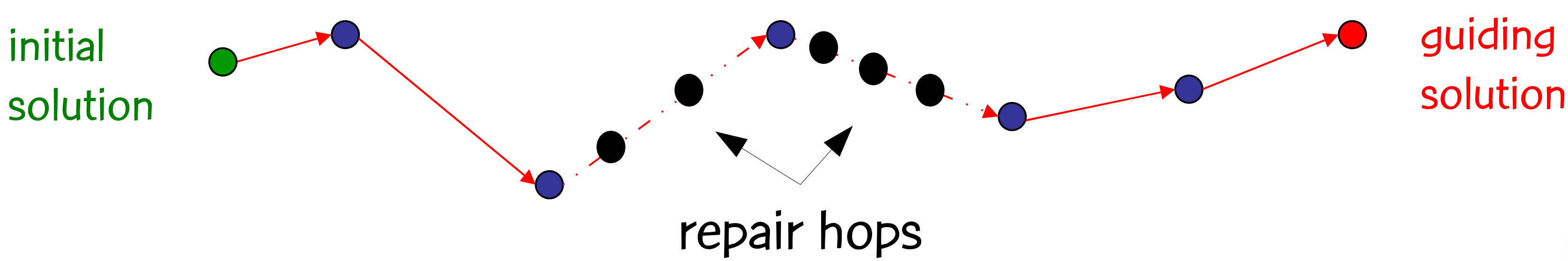




So, instead of a path with feasible solution in one single step ...



We have now a path with eventual intermediate repair hops



Concluding remarks

- We described the handover minimization problem (HMP).
- Objective of handover minimization is to reduce number of dropped calls in a celular network.
- The HMP is a special case of the generalized quadratic assignment problem (GQAP).
- We described a GRASP with path-relinking heuristic for the the GQAP and applied it on instances of the HMP.

Thanks!

These slides as well as related technical reports are available at

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