

# GRASP with path-relinking for generalized quadratic assignment problem

VIII Metaheuristics International Conference  
MIC 2009 ♣ Hamburg, Germany ♣ July 13-16, 2009

Mauricio G. C. Resende  
AT&T Labs Research  
Florham Park, New Jersey  
[mgcr@att.com](mailto:mgcr@att.com)

Joint work with  
Ricardo M. A. Silva (UFLA, Brazil &  
post-doc at AT&T Labs Research)  
and Geraldo R. Mateus (UFMG, Brazil)



# Summary

Generalized quadratic assignment problem (GQAP)

GRASP with path-relinking for GQAP

GRASP construction

Local search

Path-relinking

Experimental results

# Generalized quadratic assignment problem

July 2009

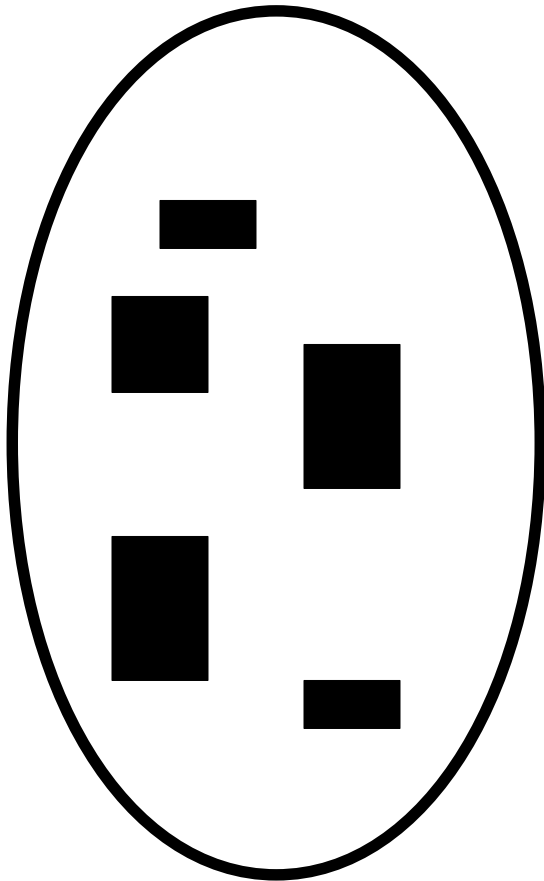
GRASP-PR for generalized quadratic assignment problem



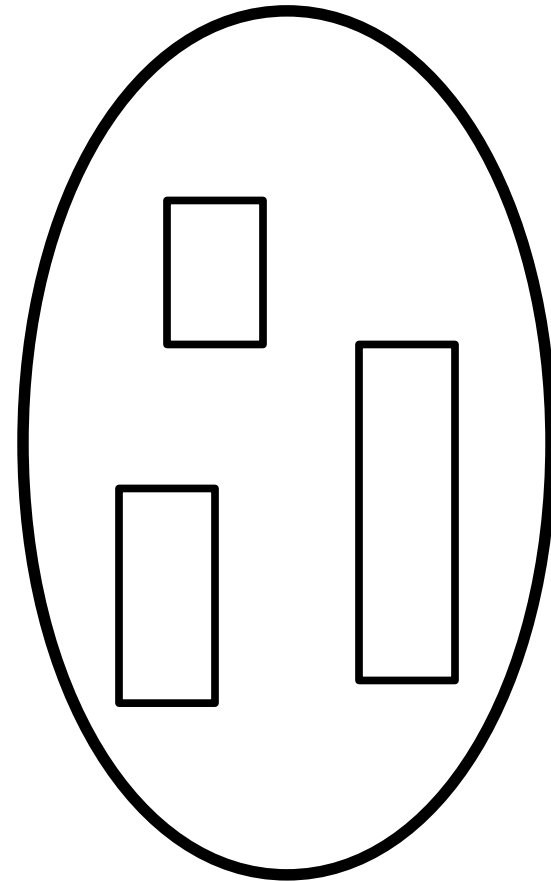
# Generalized quadratic assignment

- The GQAP is NP-hard.
- It is a 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



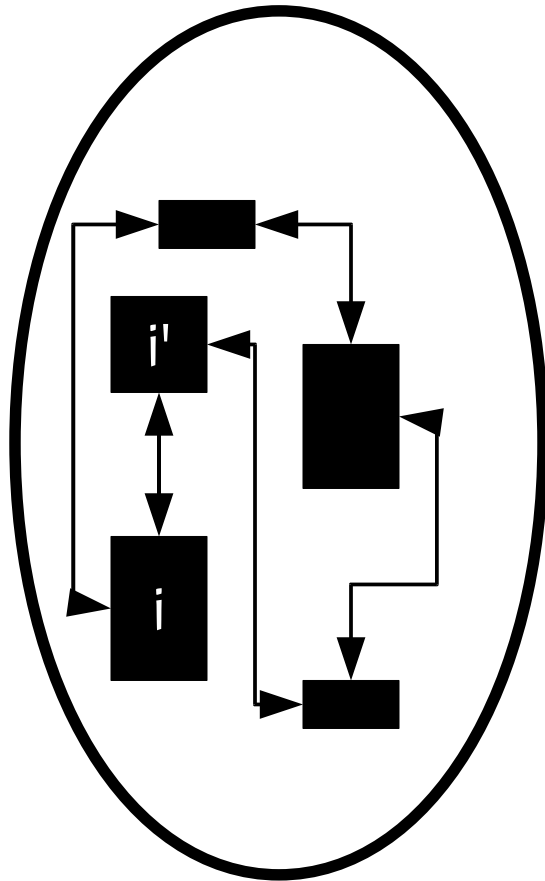
$M$ : set of  $m$  locations



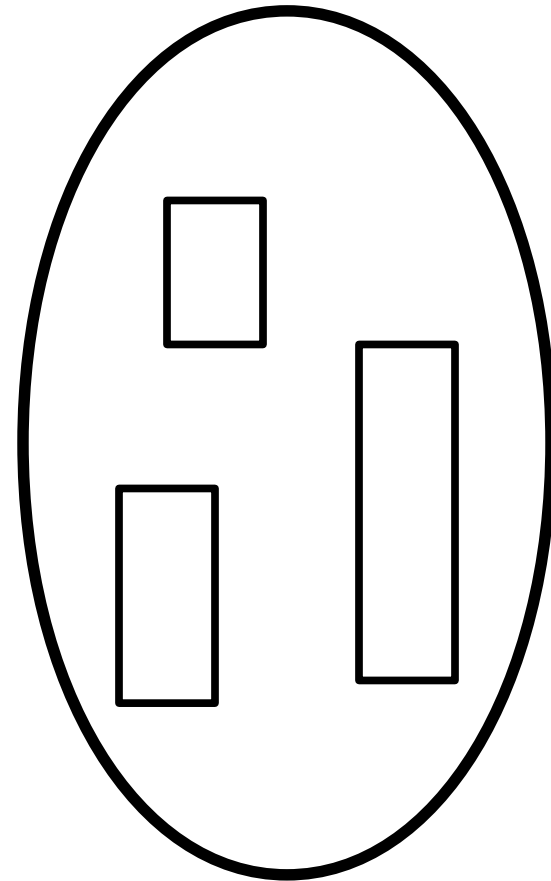
$d_i$  : capacity demanded by facility  $i \in N$

$Q_j$  : capacity of location  $j \in M$

N: set of n facilities

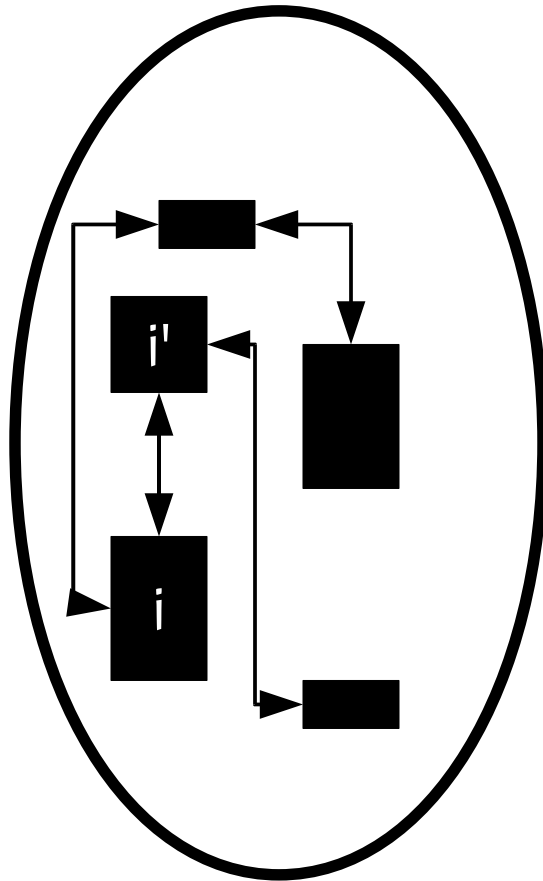


M: set of m locations

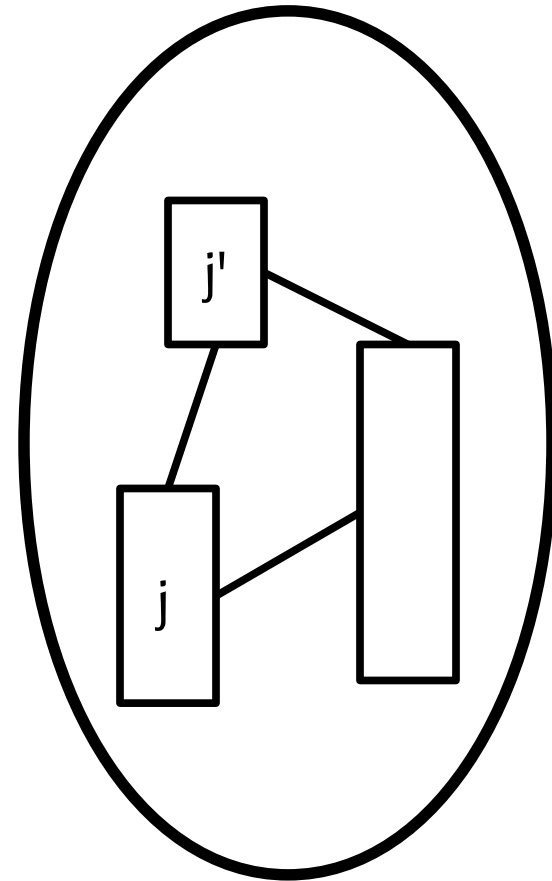


$A_{n \times n} = (a_{ij})$  : flow between facilities

N: set of n facilities



M: set of m locations

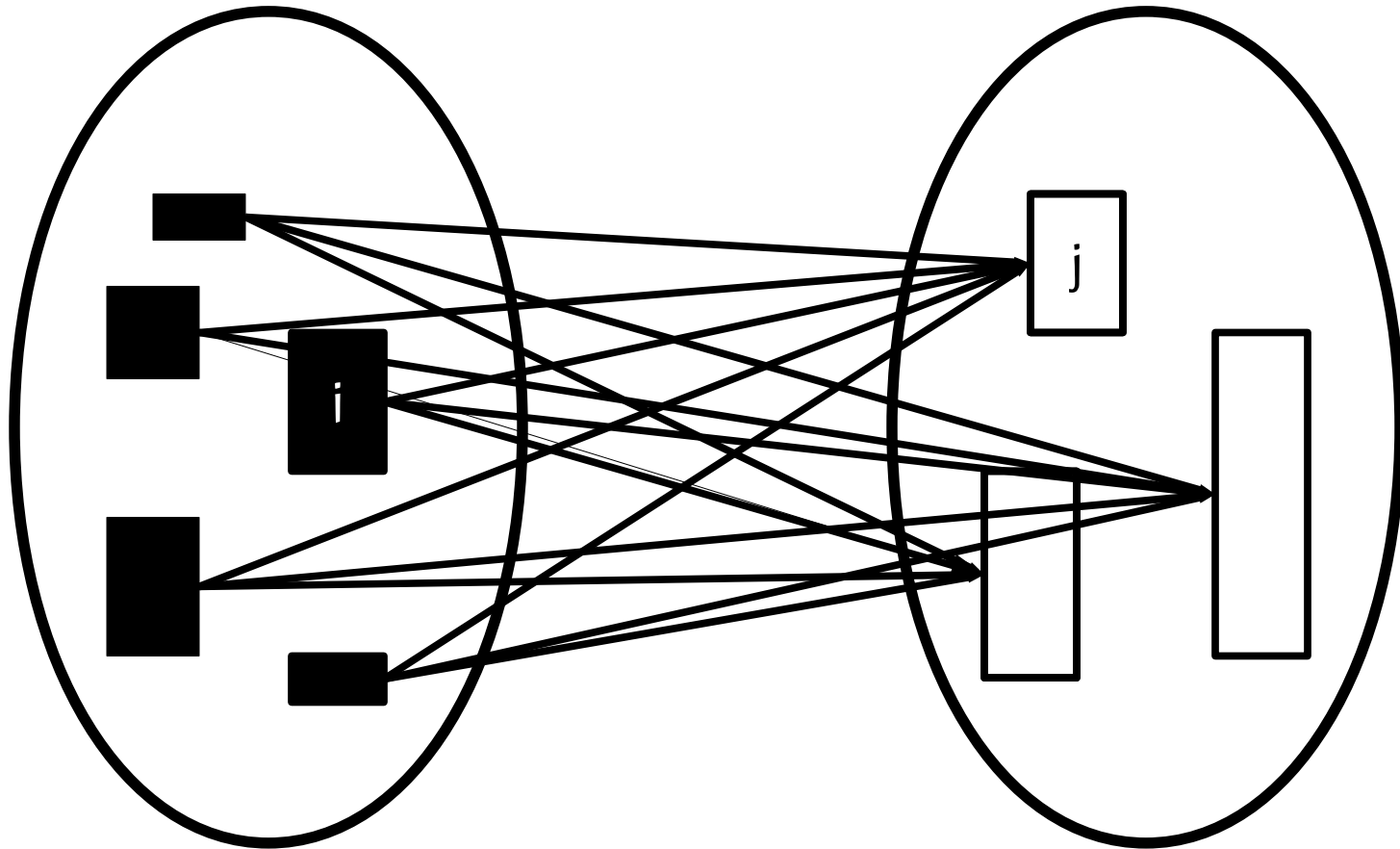


$A_{n \times n} = (a_{ii'})$  : flow between facilities

$B_{m \times m} = (b_{jj'})$  : distance between locations

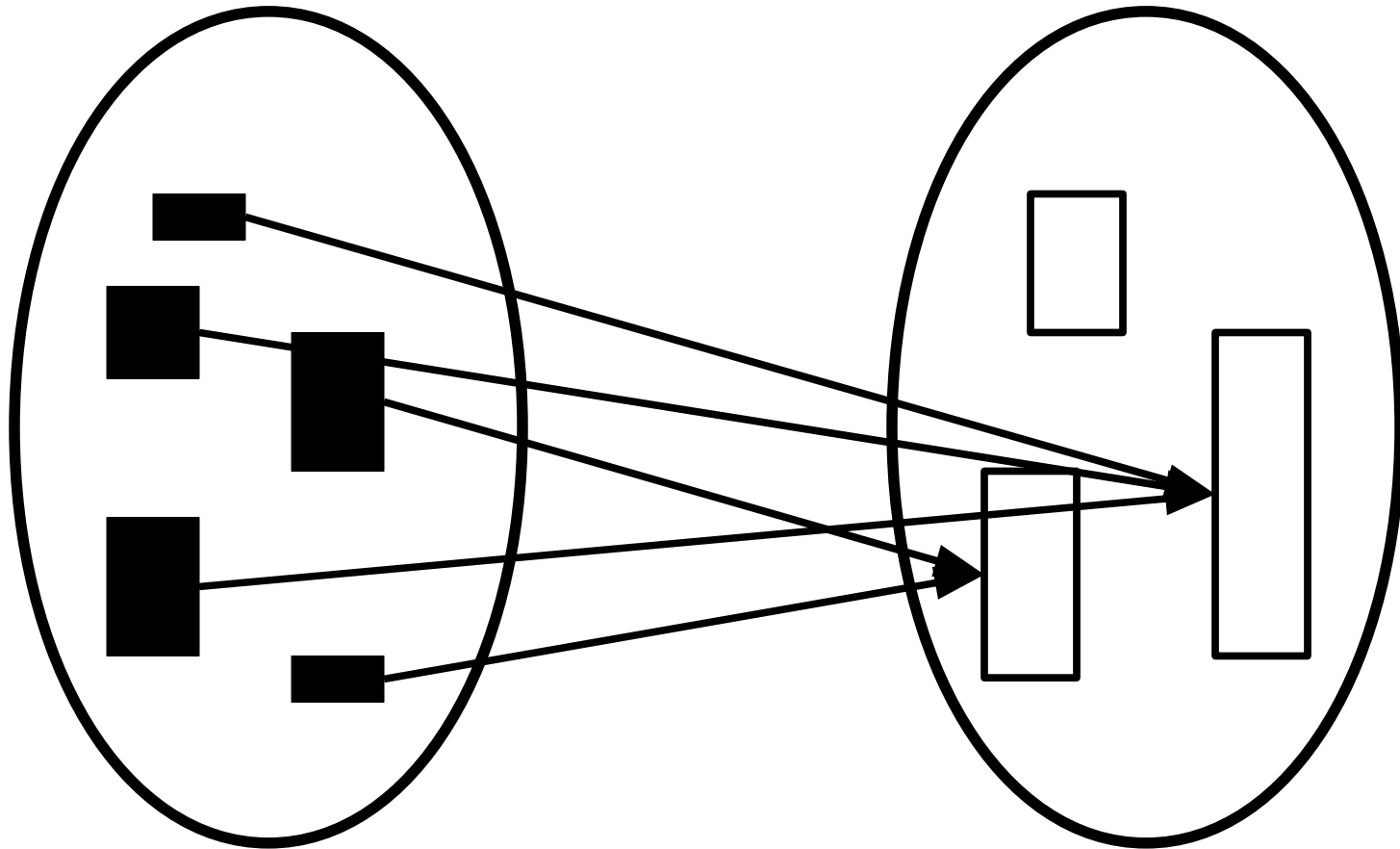
N: set of n facilities

M: set of m locations



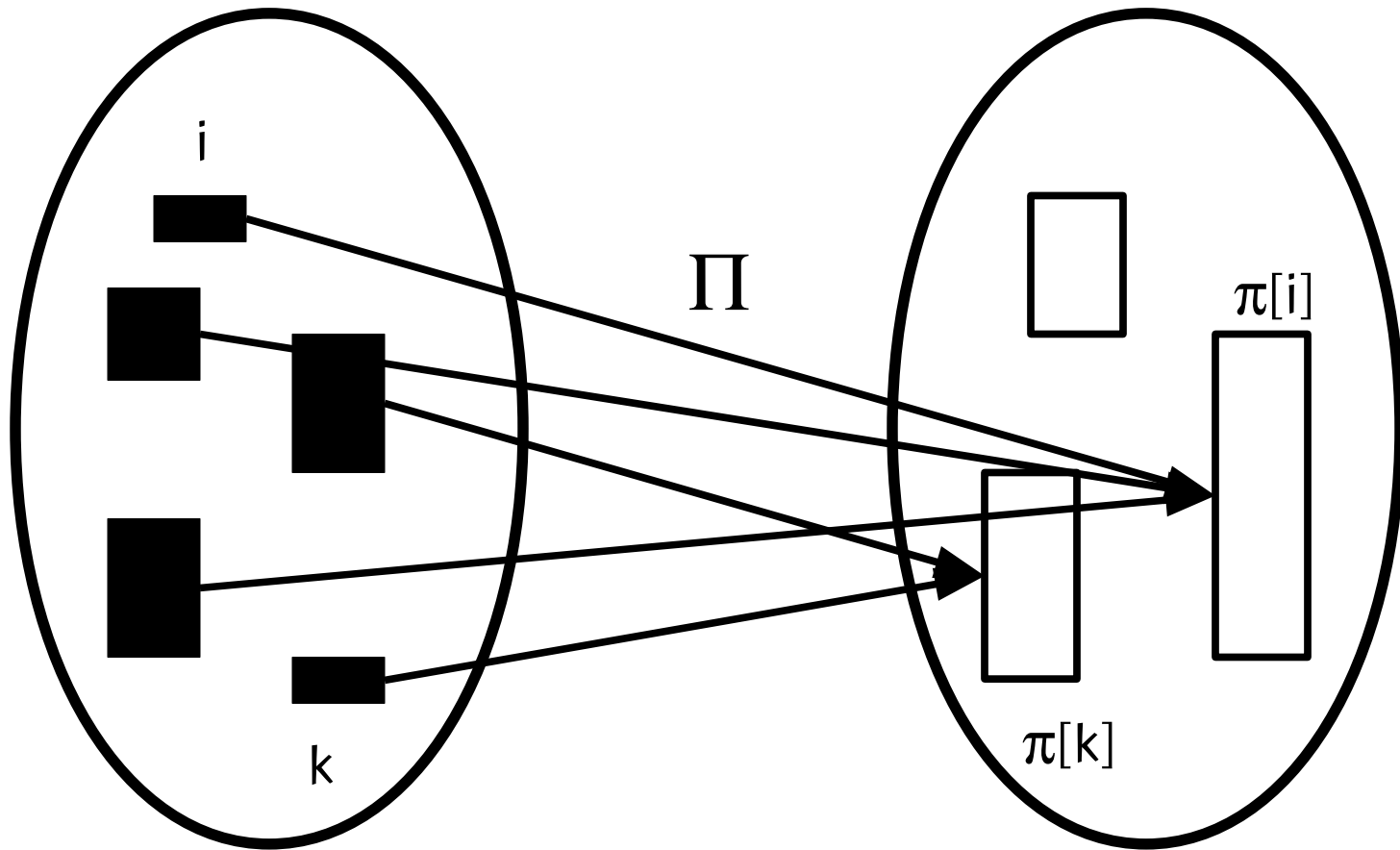
$C_{n \times m} = (c_{ij})$  : cost of assigning facility  $i \in N$  to location  $j \in M$

# 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



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

# PBX telephone migration scheduling

(A real-world GQAP problem)

# PBX telephone migration scheduling

- Phone migration occurs when an organization upgrades to a newer phone switch (PBX).
- All phones using the old PBX must be moved to the new PBX.
- Each phone belong to one or more groups of phones that interact and should to be moved together in same time period.

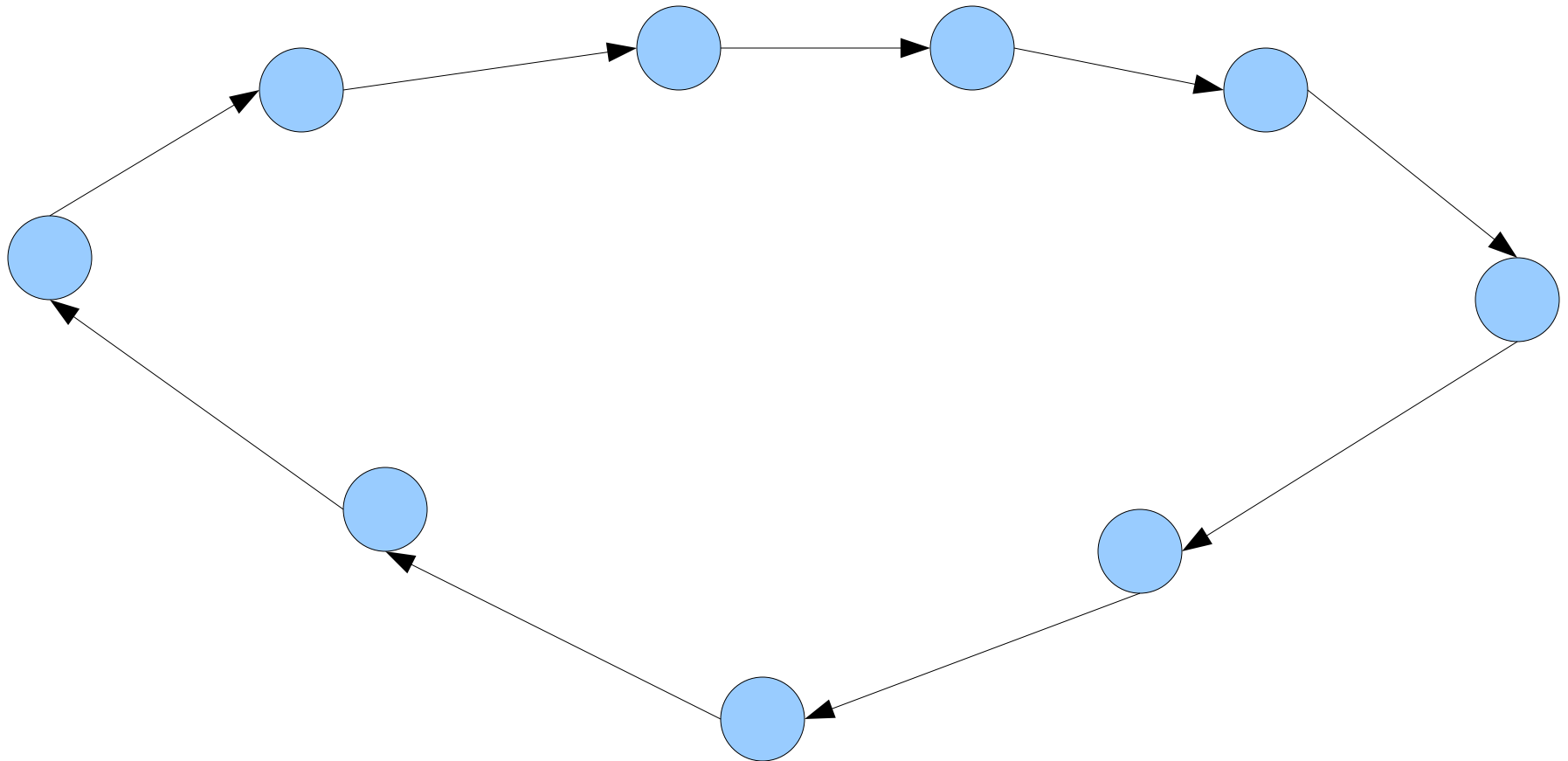
# PBX telephone migration scheduling

- Phone migration occurs when an organization upgrades to a newer phone switch (PBX).
- All phones using the old PBX must be moved to the new PBX.
- Each phone belong to one or more groups of phones that interact and should to be moved together in same time period.

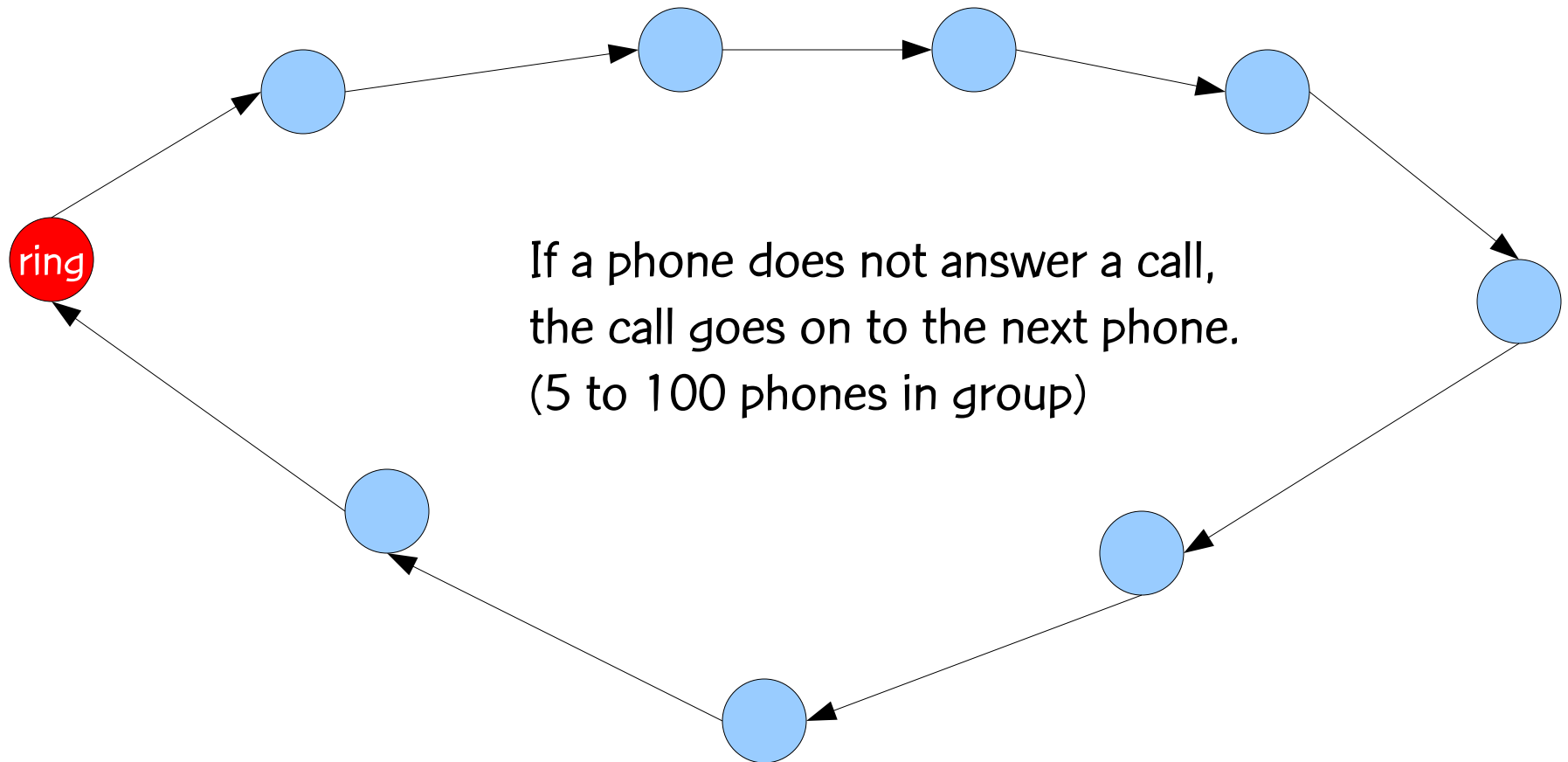
# PBX telephone migration scheduling

- Phone migration occurs when an organization upgrades to a newer phone switch (PBX).
- All phones using the old PBX must be moved to the new PBX.
- Each phone belongs to one or more groups of phones that interact and should to be moved together in same time period.

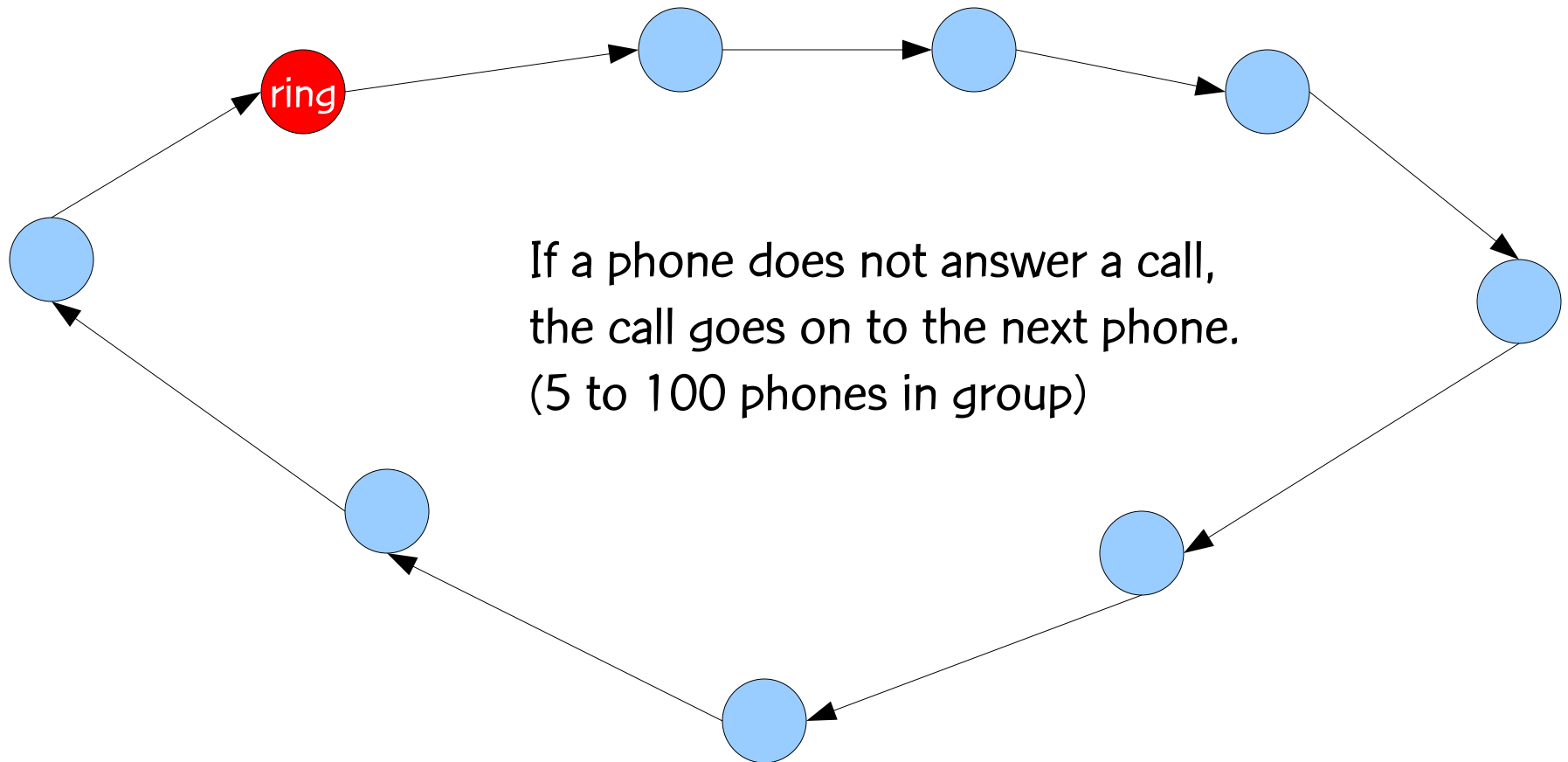
# Multi-line hunt group



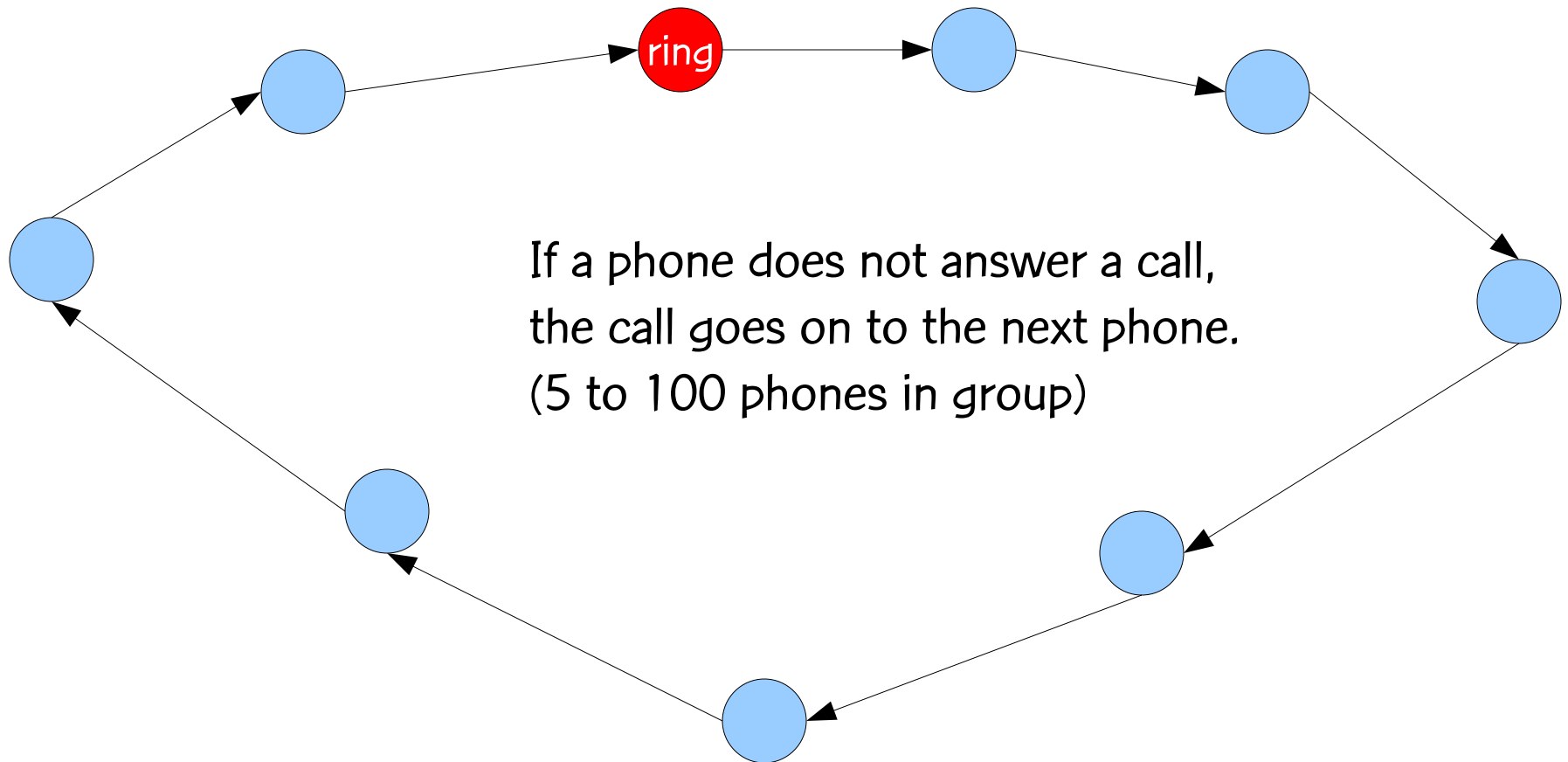
# Multi-line hunt group



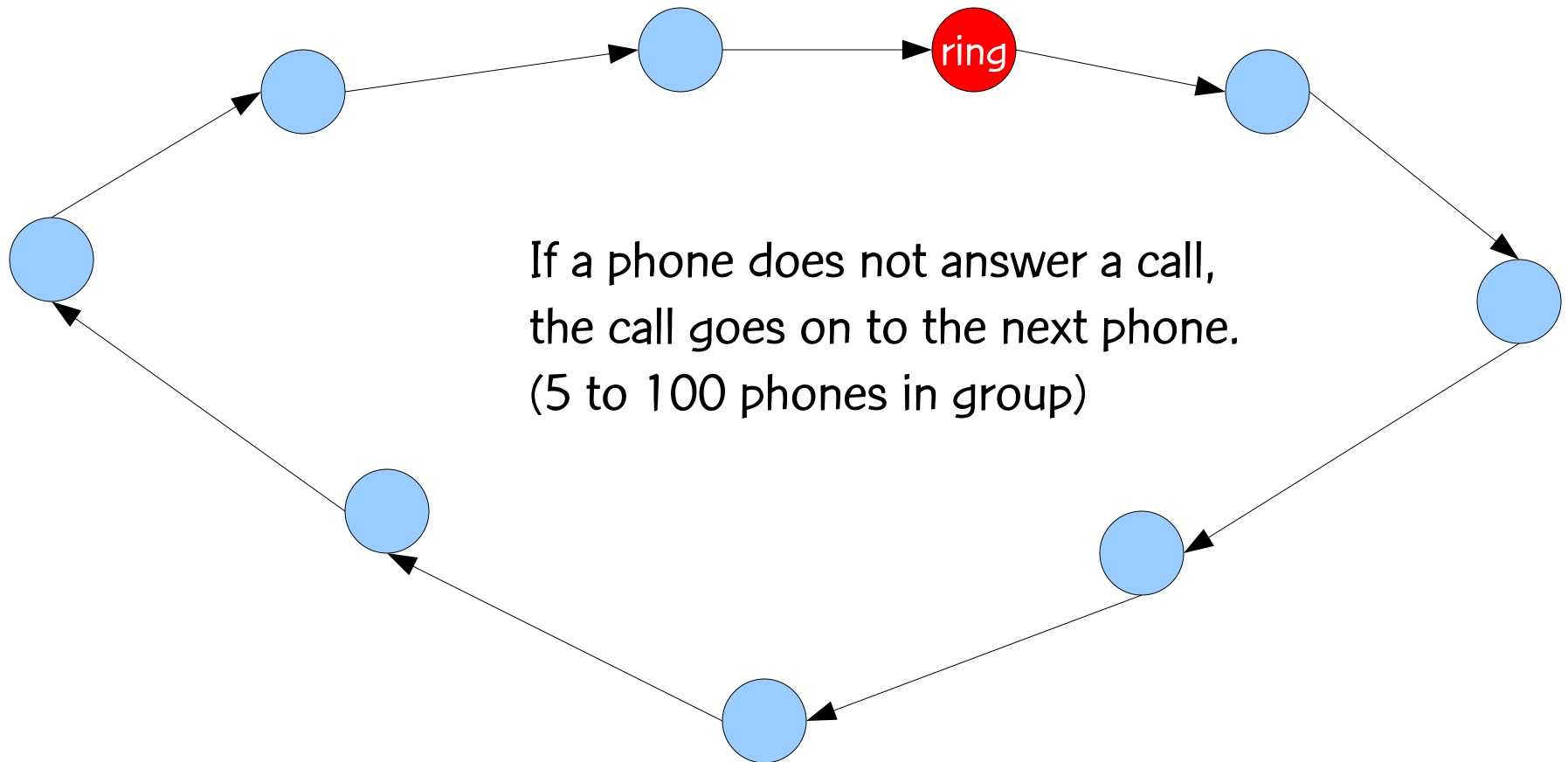
# Multi-line hunt group



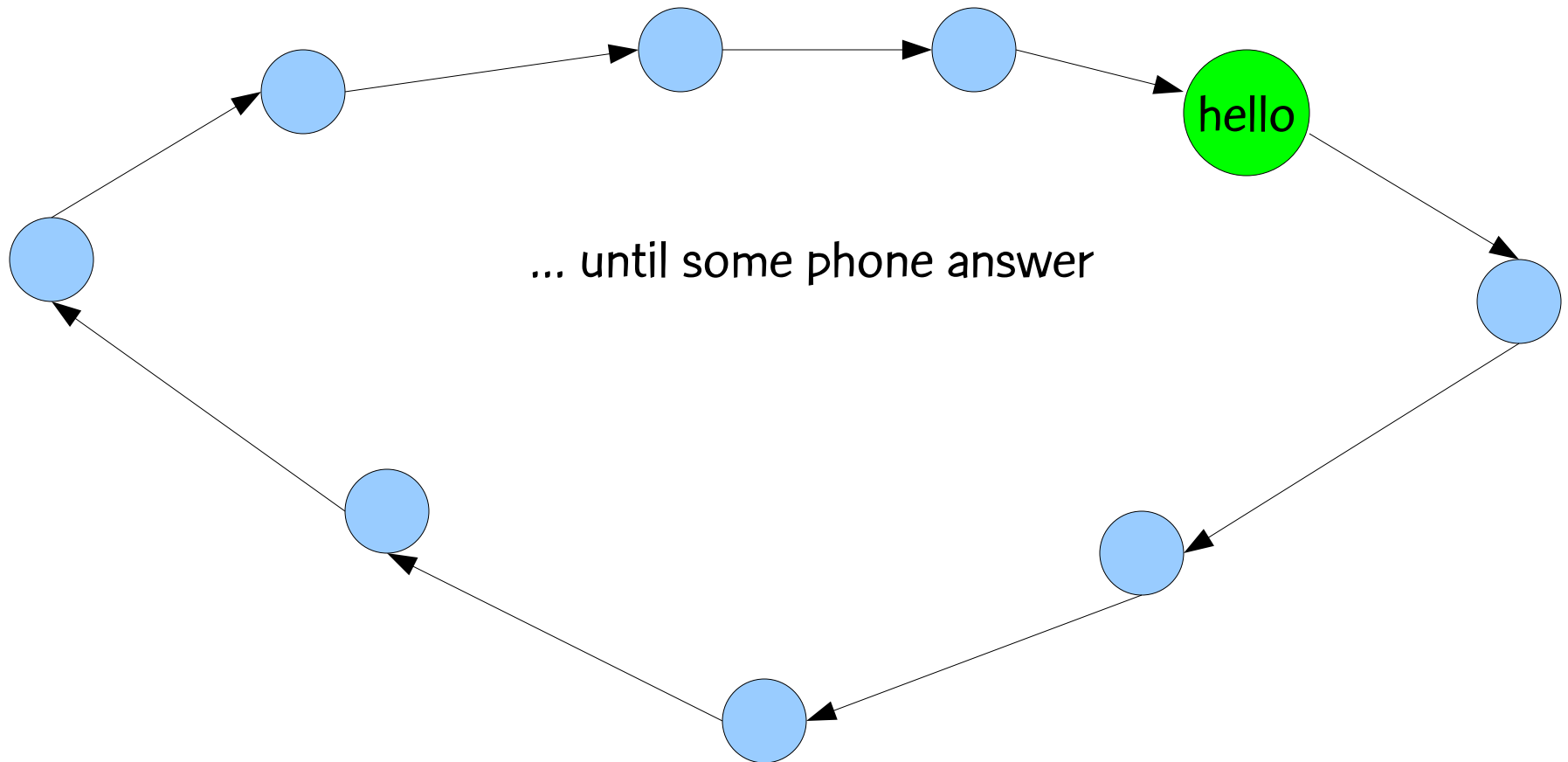
# Multi-line hunt group



# Multi-line hunt group

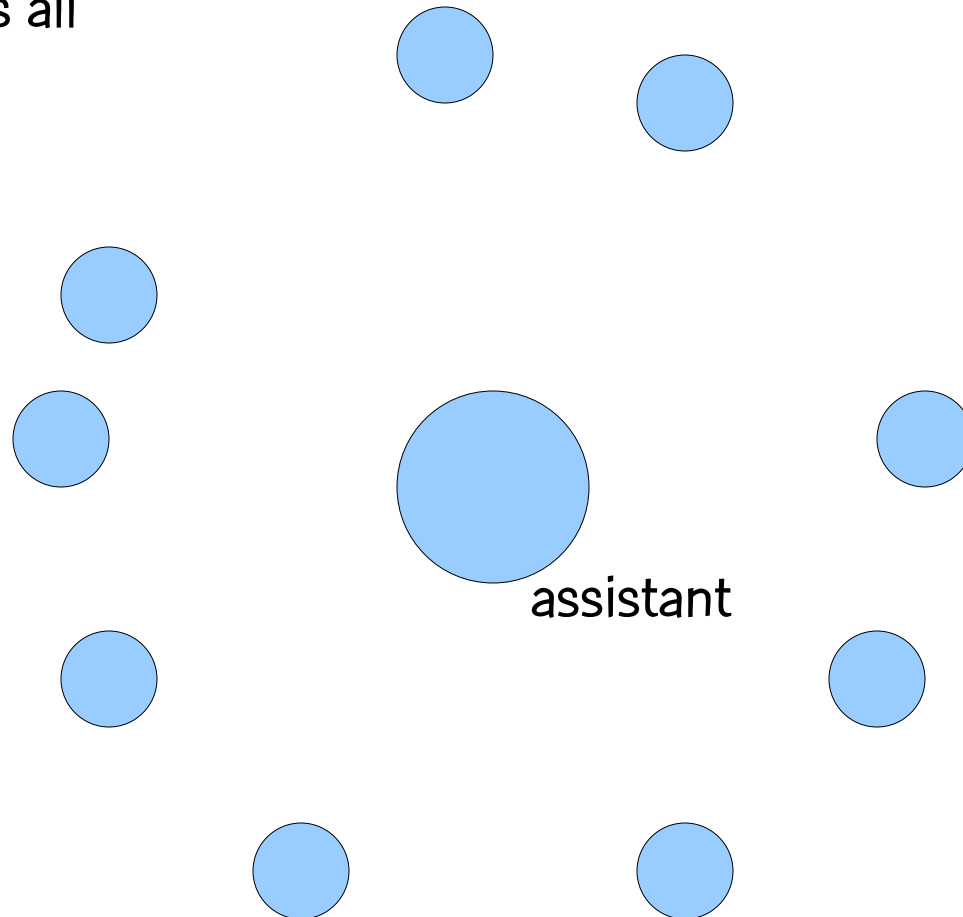


# Multi-line hunt group



# Shared TN

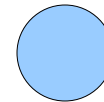
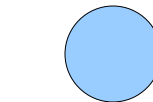
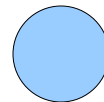
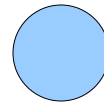
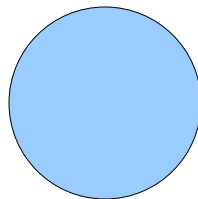
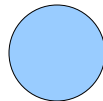
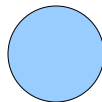
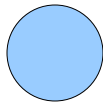
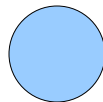
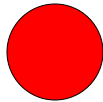
Assistant answers all  
calls to group.



# Shared TN

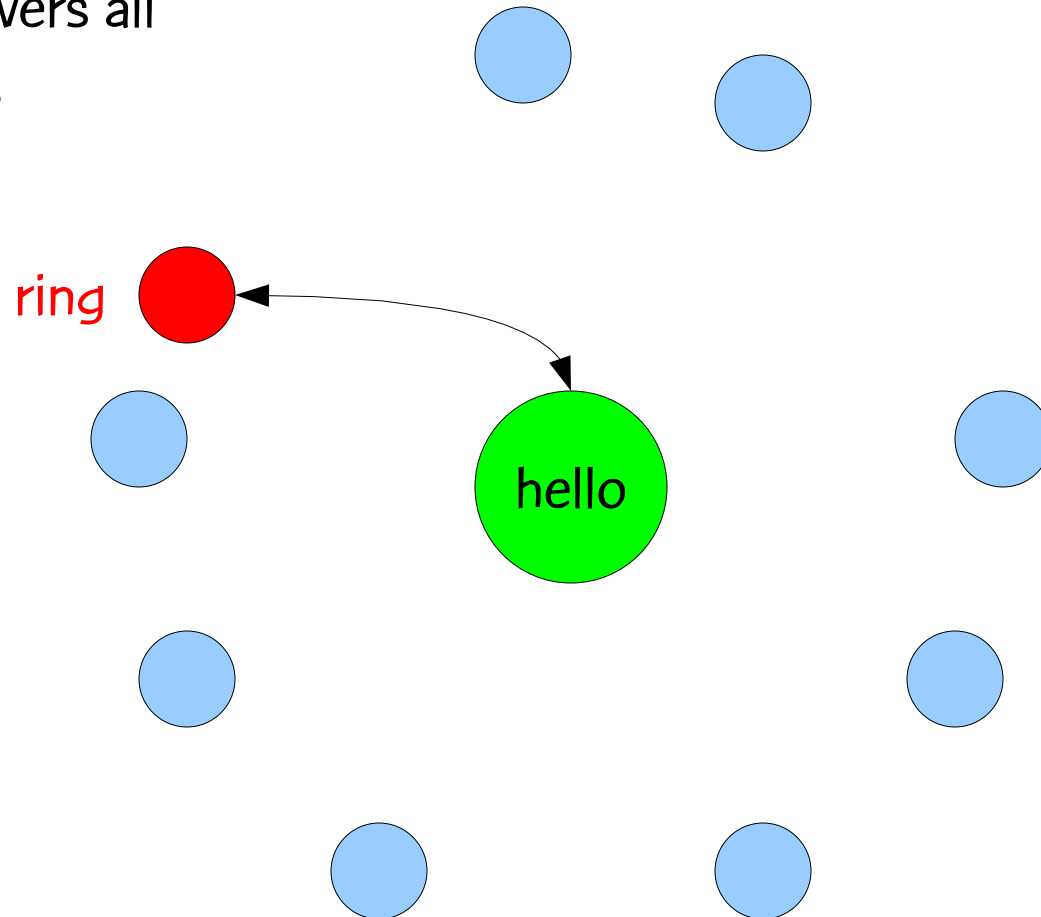
Assistant answers all  
calls to group.

ring



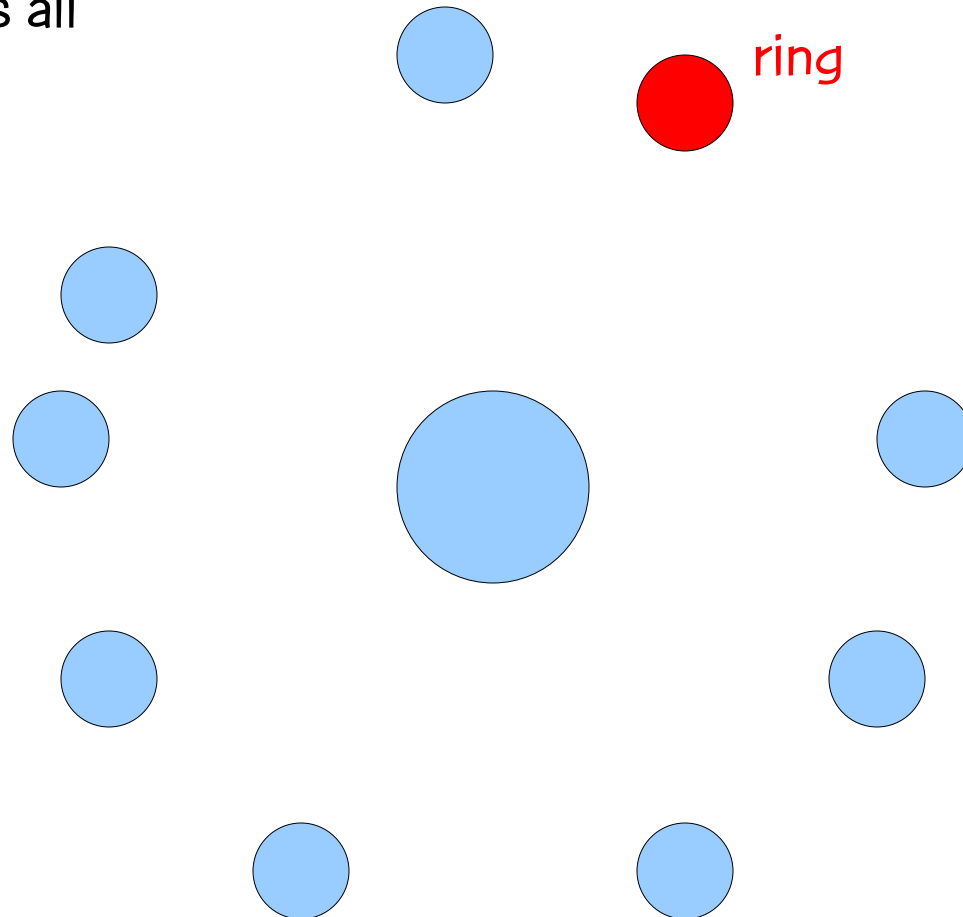
# Shared TN

Assistant answers all  
calls to group.



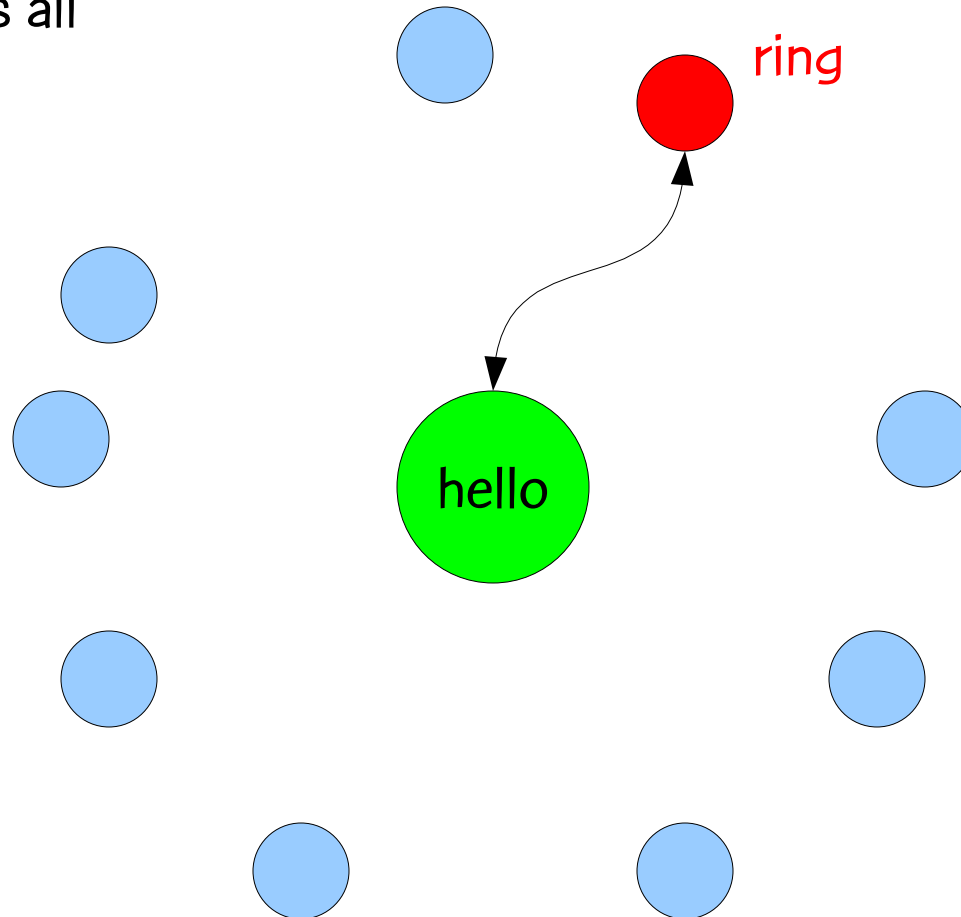
# Shared TN

Assistant answers all  
calls to group.



# Shared TN

Assistant answers all  
calls to group.



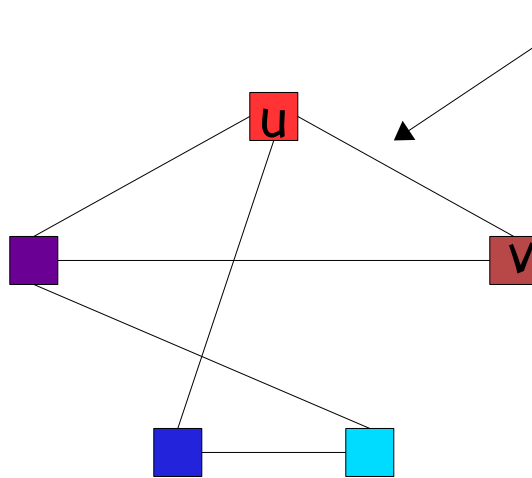
# PBX telephone migration scheduling

- Each phone belongs to one or more groups of phones that interact and should to be moved together in same time period.
- Given penalties for not moving a pair of phones together and a maximum number of phones that can be moved in a time period, find assignment of phones to periods such that total penalty is minimized.

# PBX telephone migration scheduling

- Each phone belong to one or more groups of phones that interact and should to be moved together in same time period.
- Given penalties for not moving a pair of phones together and a maximum number of phones that can be moved in a time period, find assignment of phones to periods such that total penalty is minimized.

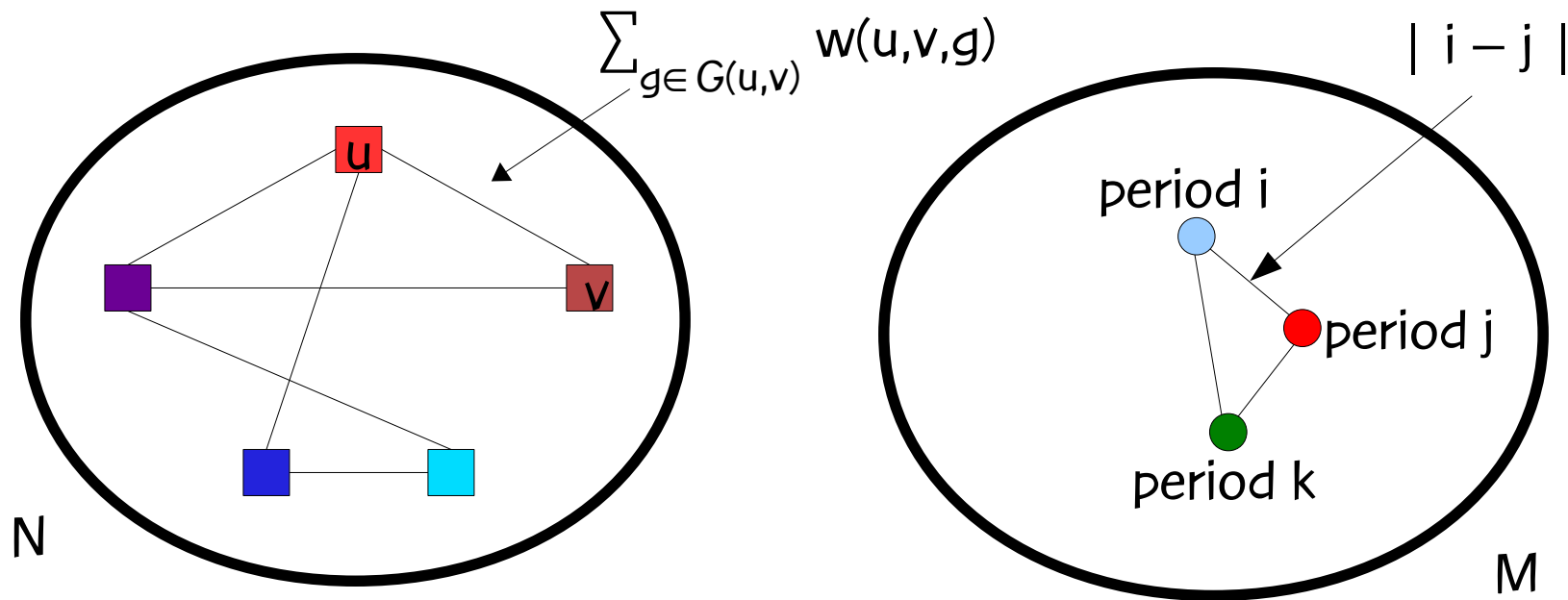
# PBX telephone migration scheduling



$$\sum_{g \in G(u,v)} w(u,v,g)$$

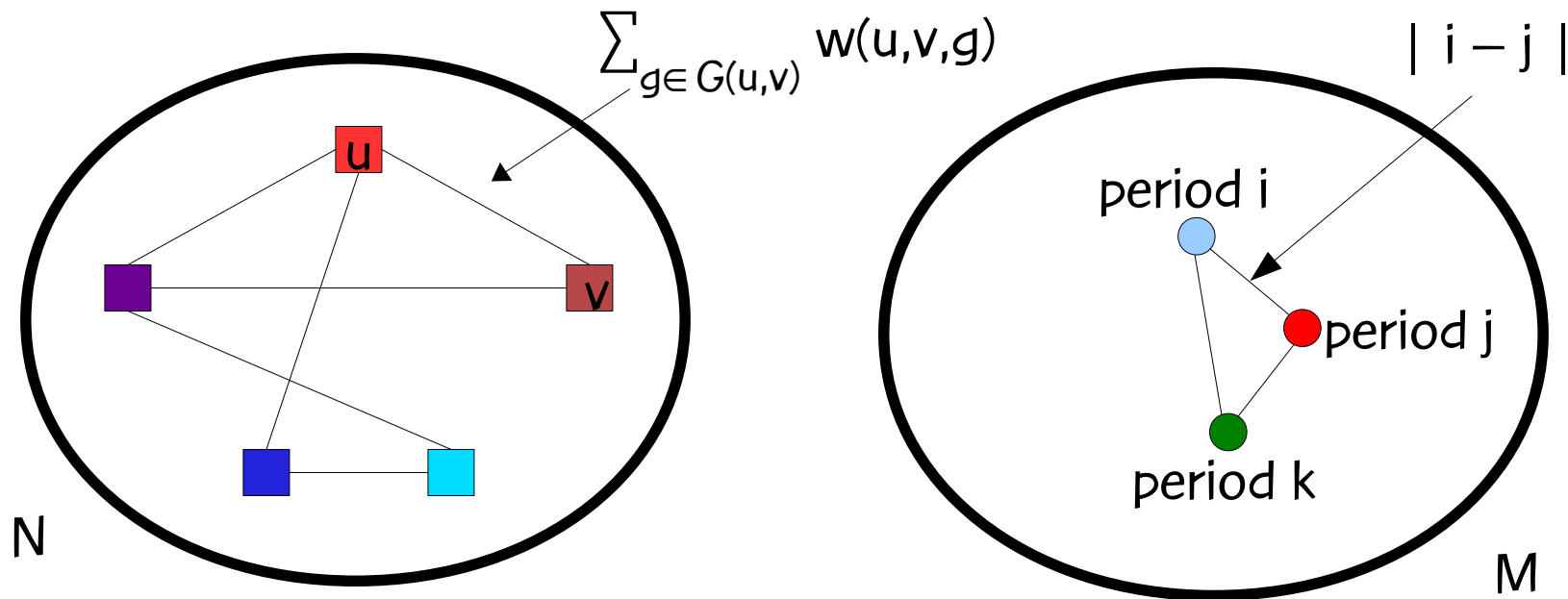
Each phone is a facility.  
Flow between phones is penalty  
associated with phone pair.

# PBX telephone migration scheduling as GQAP



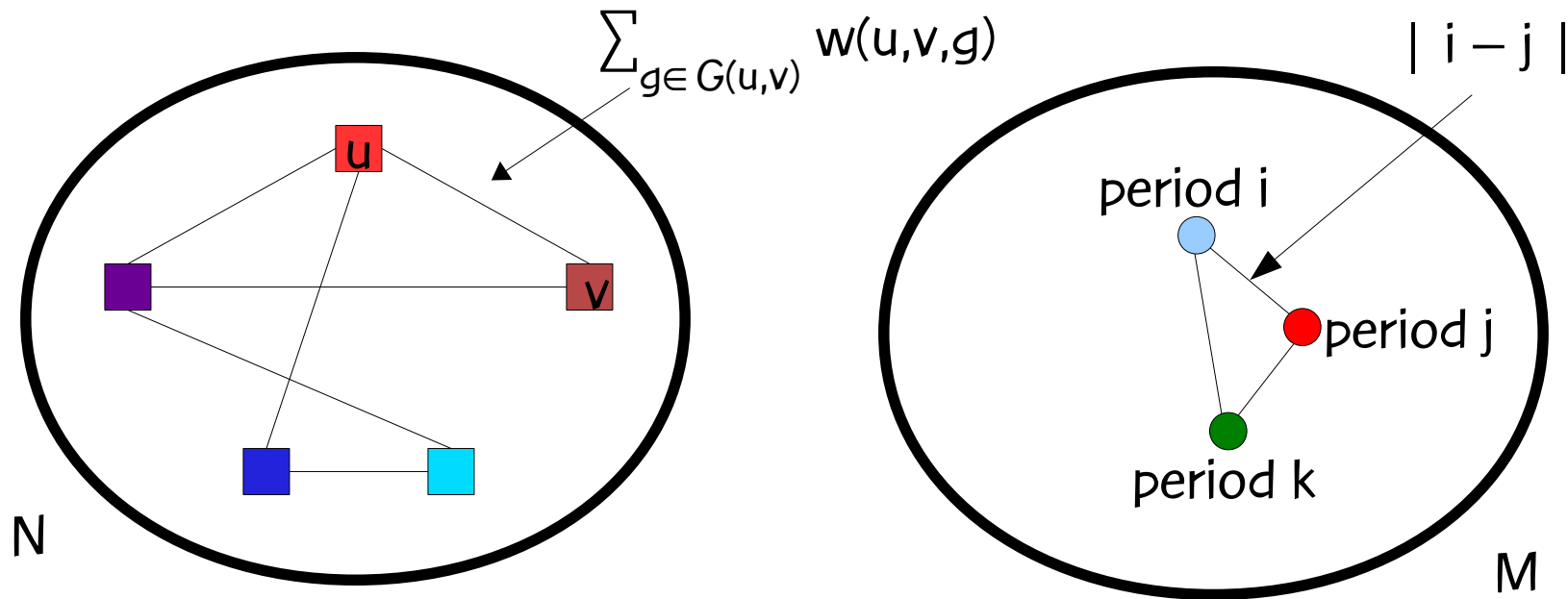
- Facilities (phones) and locations (periods);

# PBX telephone migration scheduling as GQAP



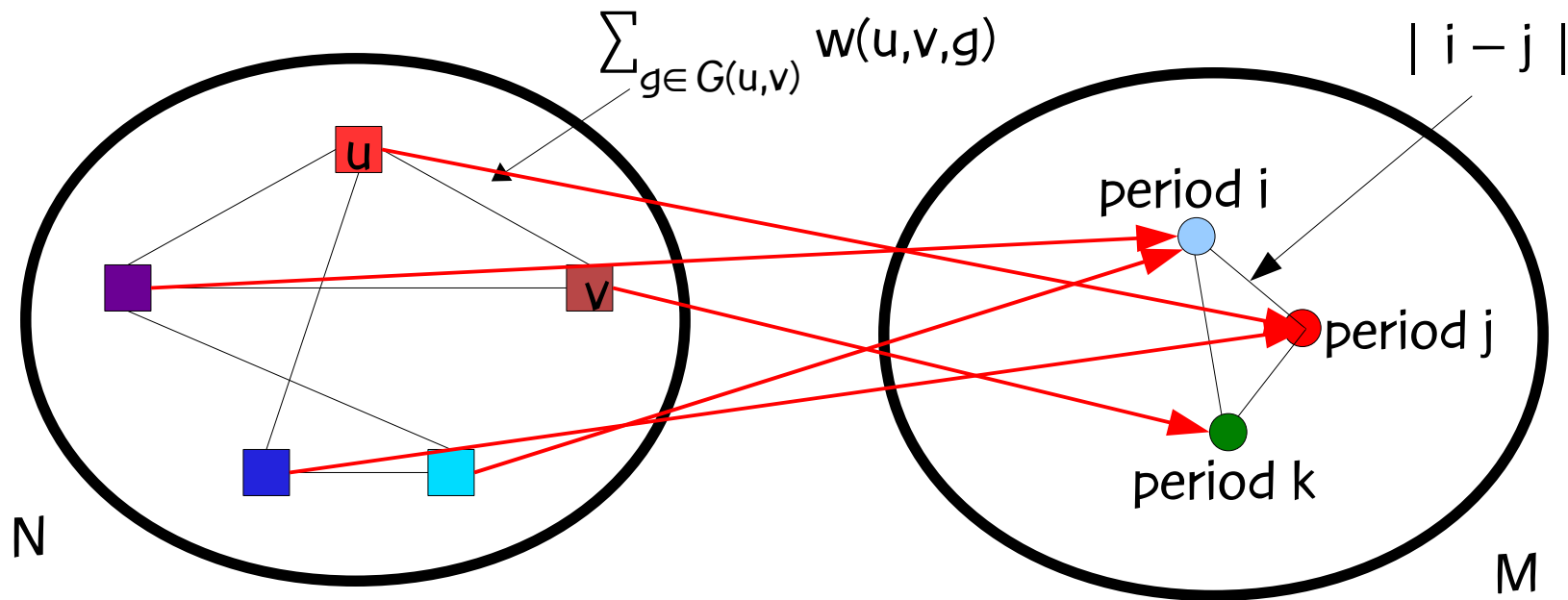
- Facilities (phones) and locations (periods);
- Flow (penalty) between facilities (phones)  $u$  and  $v = \sum_{g \in G(u,v)} w(u,v,g)$ ;

# PBX telephone migration scheduling as GQAP



- Facilities (phones) and locations (periods);
- Flow (penalty) between facilities (phones)  $u$  and  $v = \sum_{g \in G(u,v)} w(u,v,g)$ ;
- Distance (time interval) between locations (periods)  $i$  and  $j = |i - j|$ .

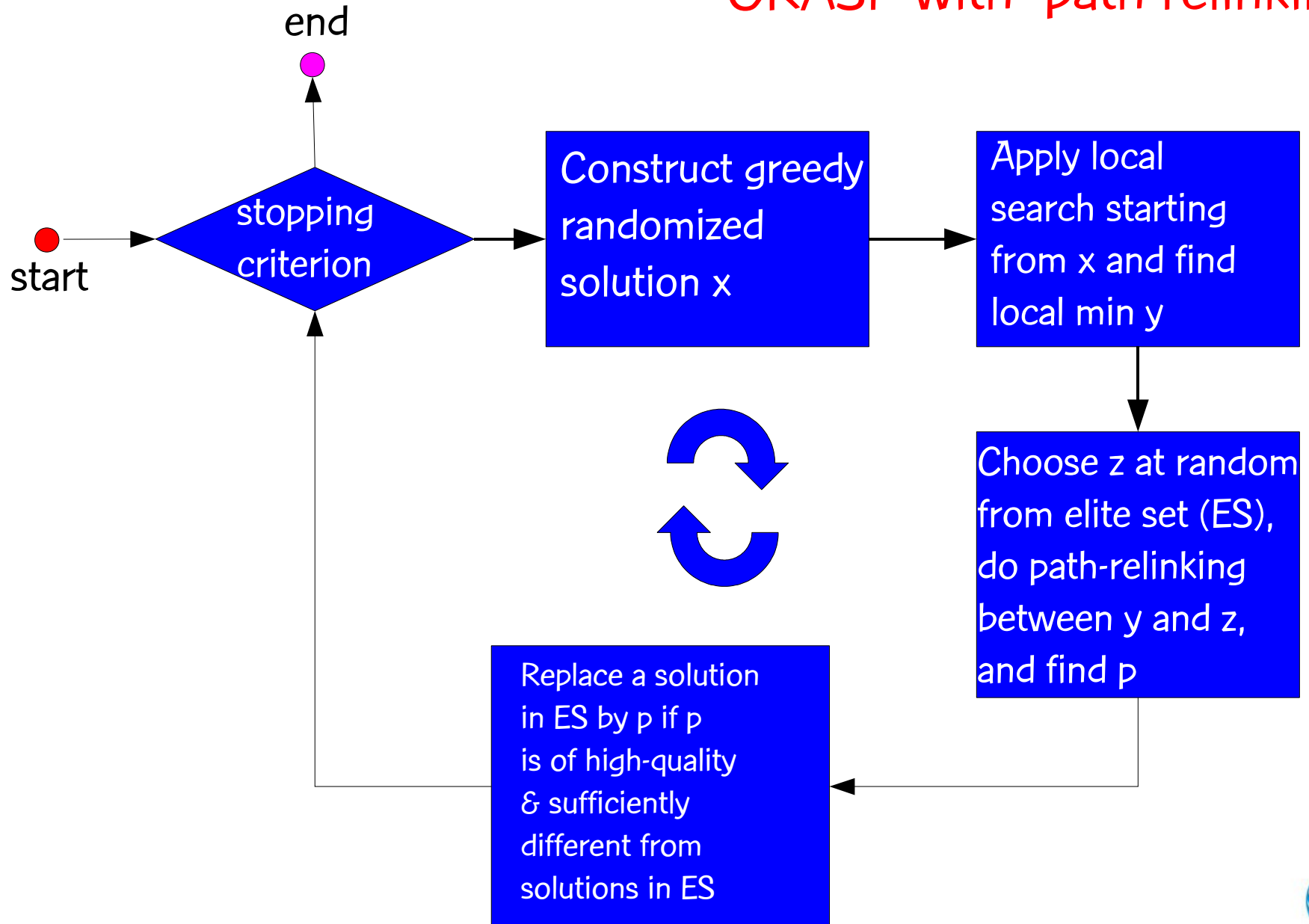
# PBX telephone migration scheduling as GQAP



Find assignment of phones to periods such that total penalty is minimized without violating capacity of each period.

# Solution method

# GRASP with path-relinking



# Components

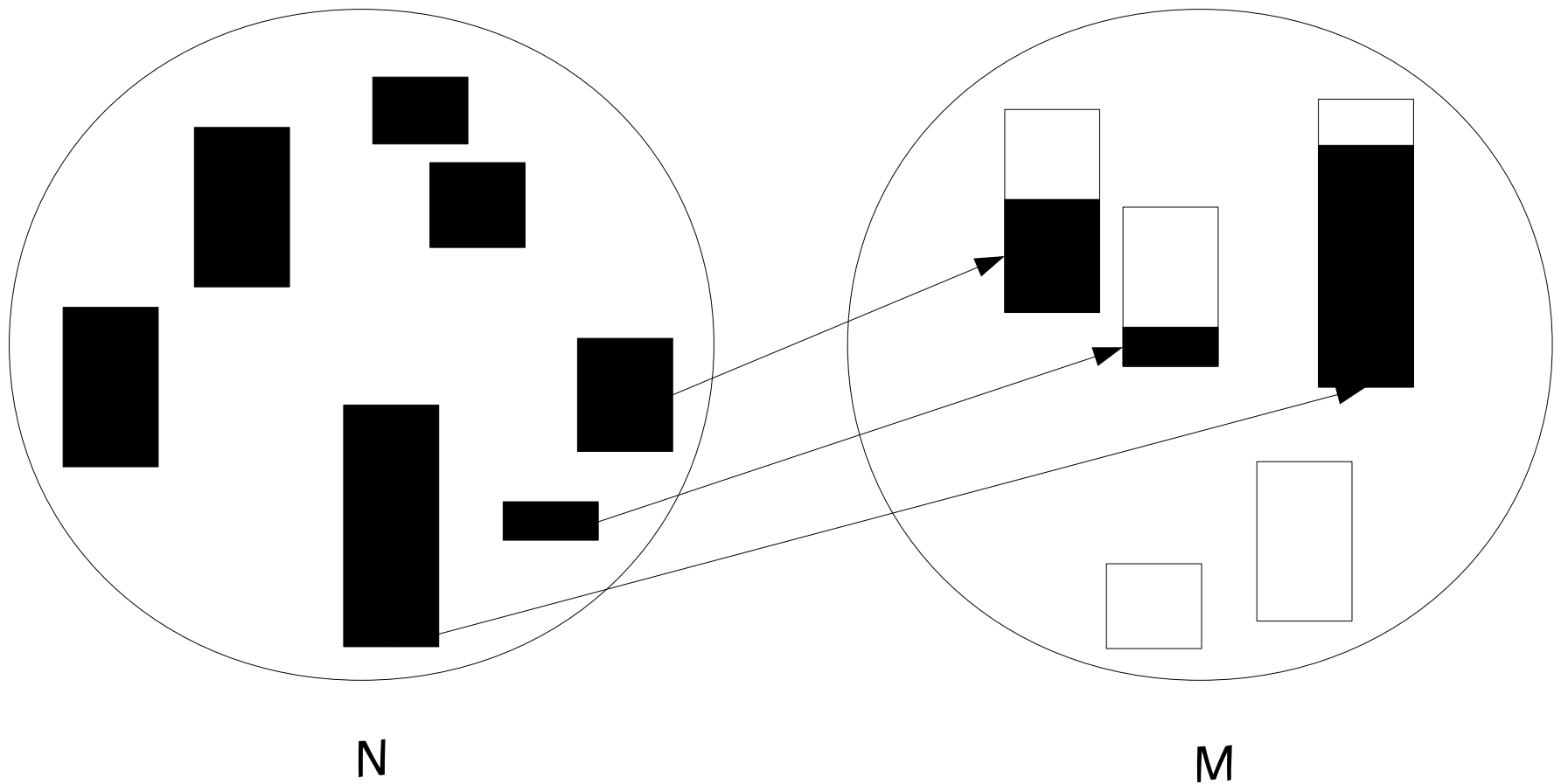
- Construction of greedy randomized solution
- Local search
- Path-relinking

# GRASP construction

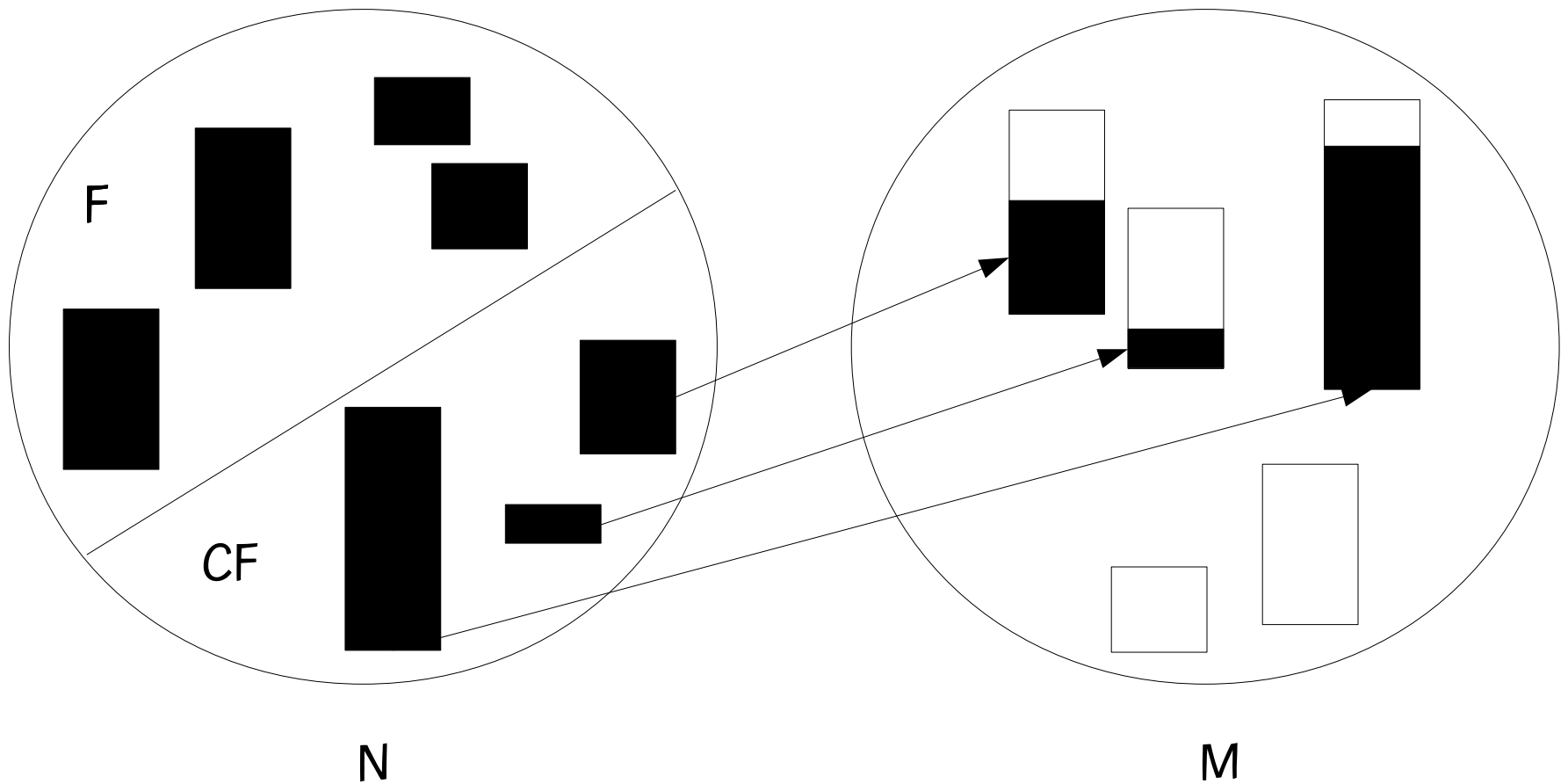
July 2009

GRASP-PR for generalized quadratic assignment problem

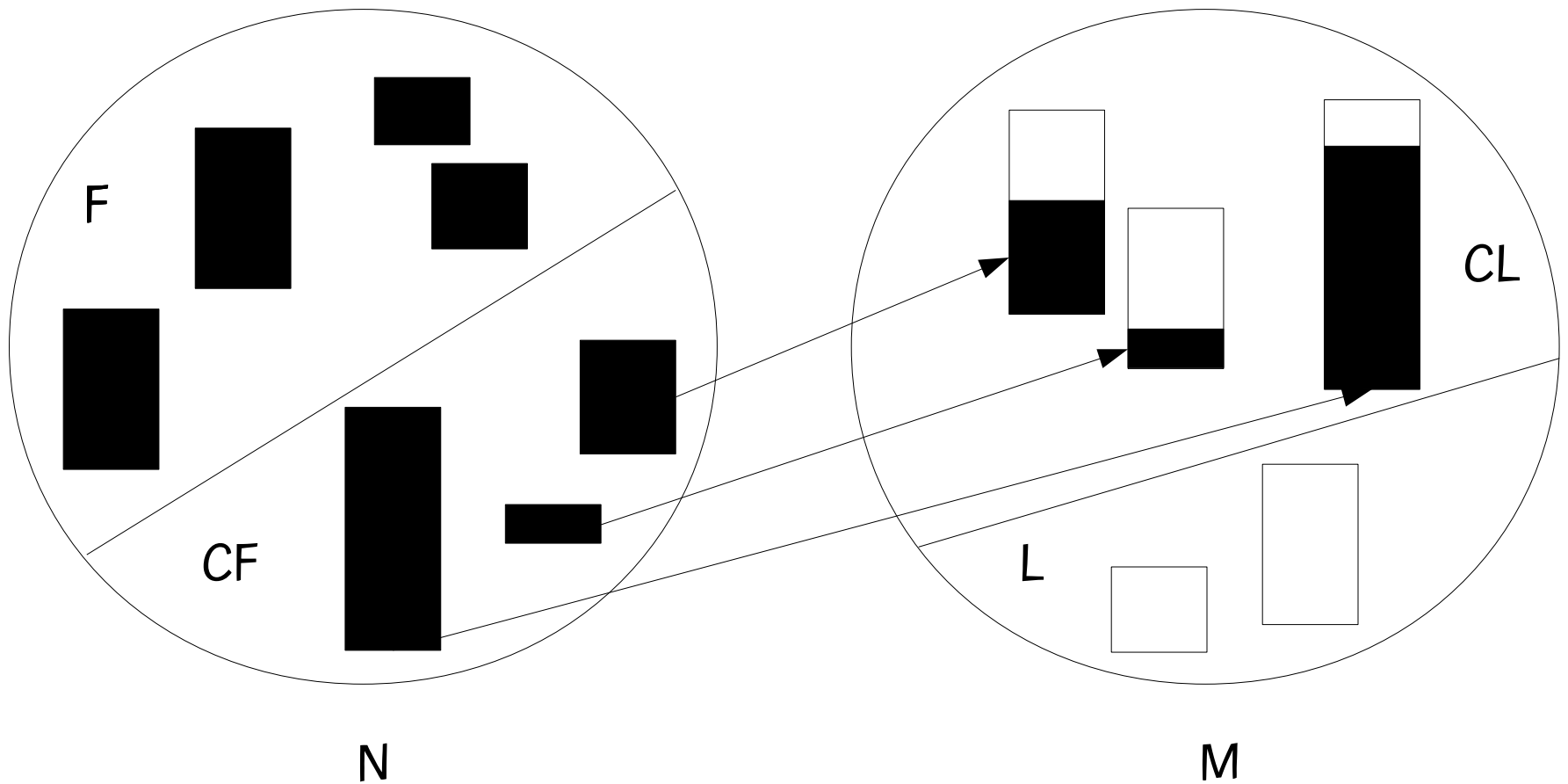




Suppose a number of assignments have already been made

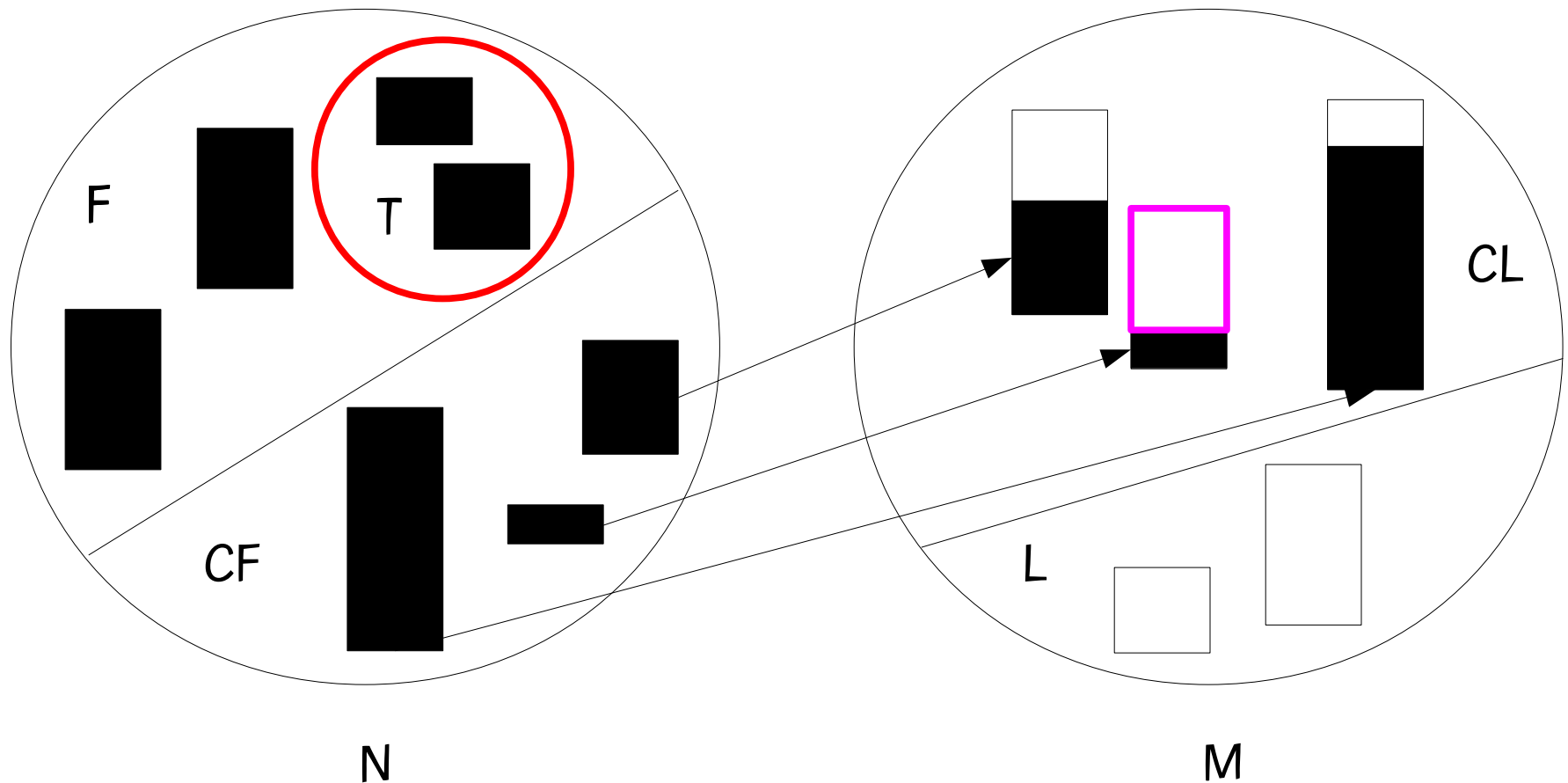


$N = F \cup CF$ , where  $CF$  is the set of assigned facilities and  $F$  the set of facilities not yet assigned to some location



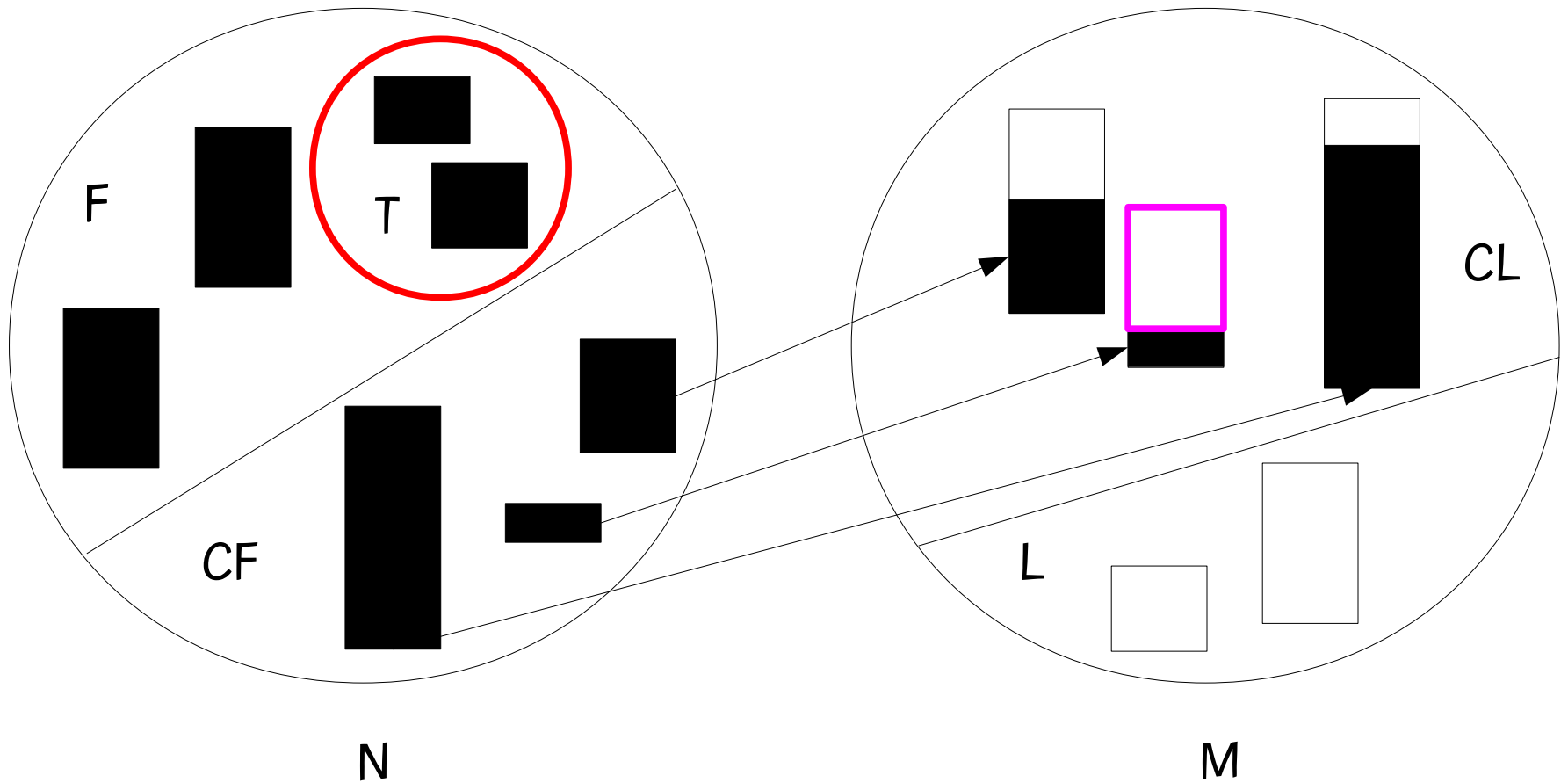
$M = L \cup CL$ , where  $CL$  is the set of previously chosen locations and  $L$  the set of unselected locations.

# Procedure to select a new location from set L



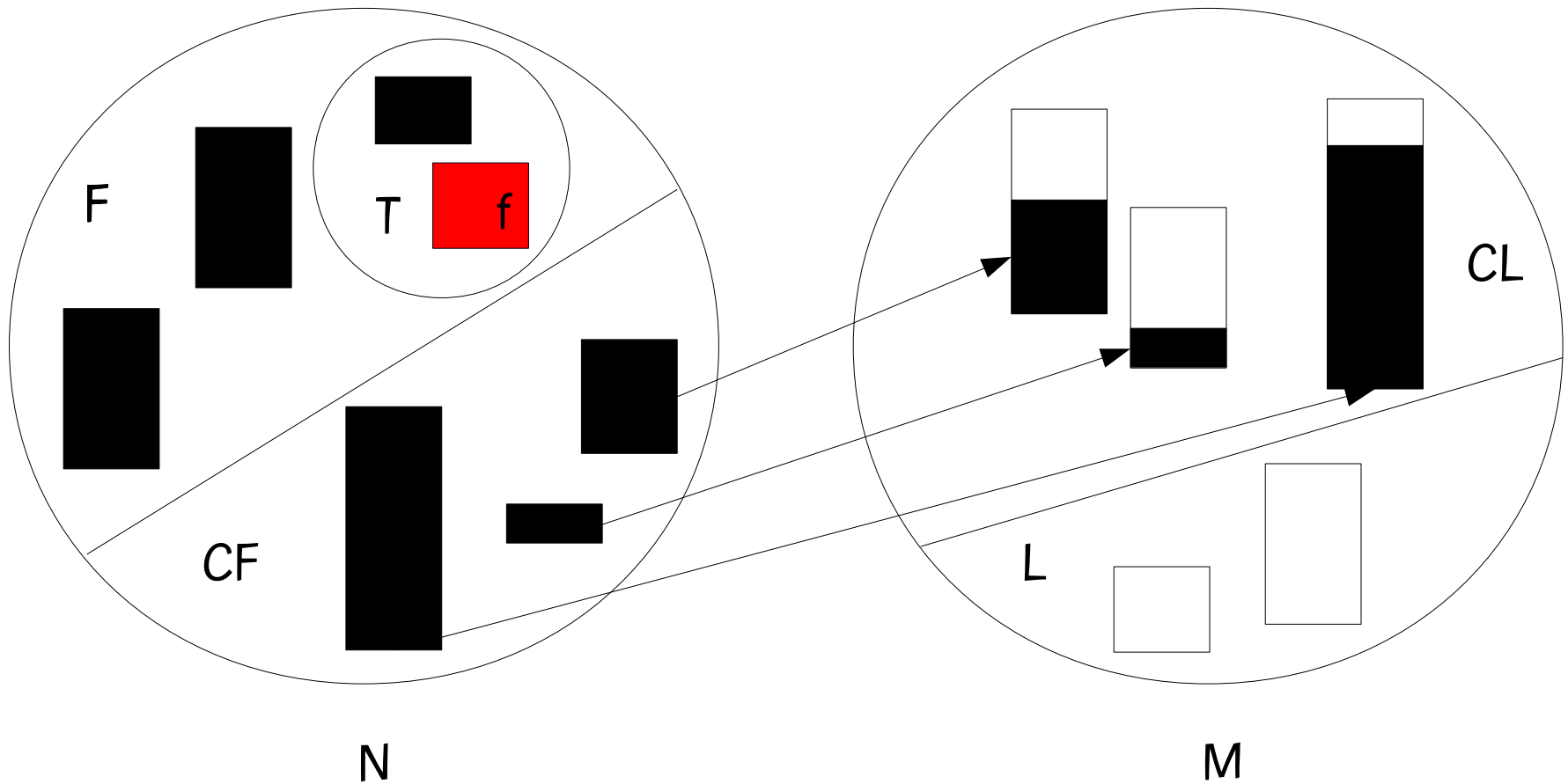
With probability  $1 - (|T|/|F|)$ , randomly select a new location  $I$  from  $L$ , where the **set T** consists of all unassigned facilities with demands less than or equal to the **maximum available capacity of locations in CL** and move location  $I$  to  $CL$

# Procedure to select a new location from set L



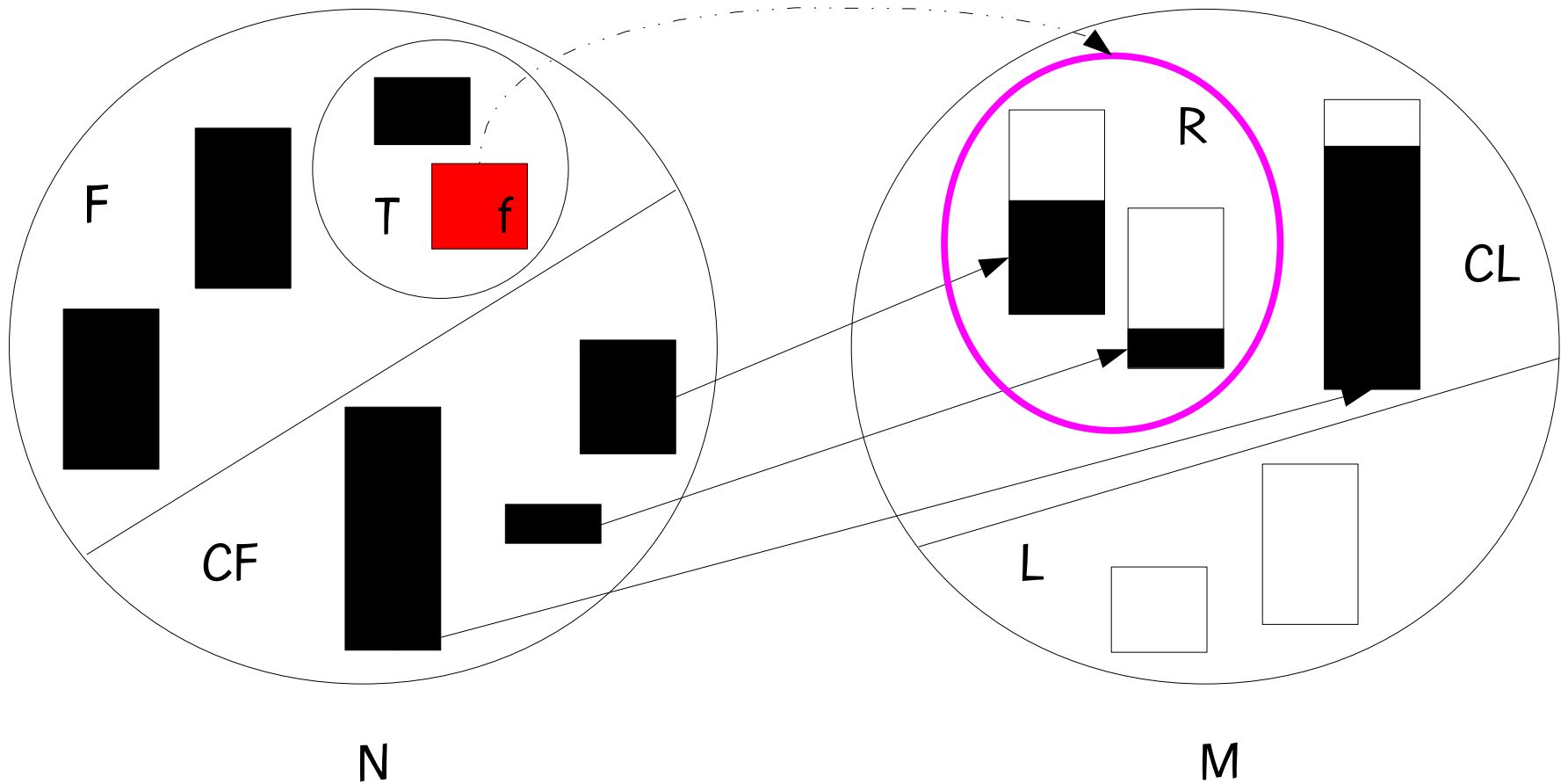
Favor locations in L that have **high available capacity** and that are **close to all locations in CL**

# Facility selection procedure



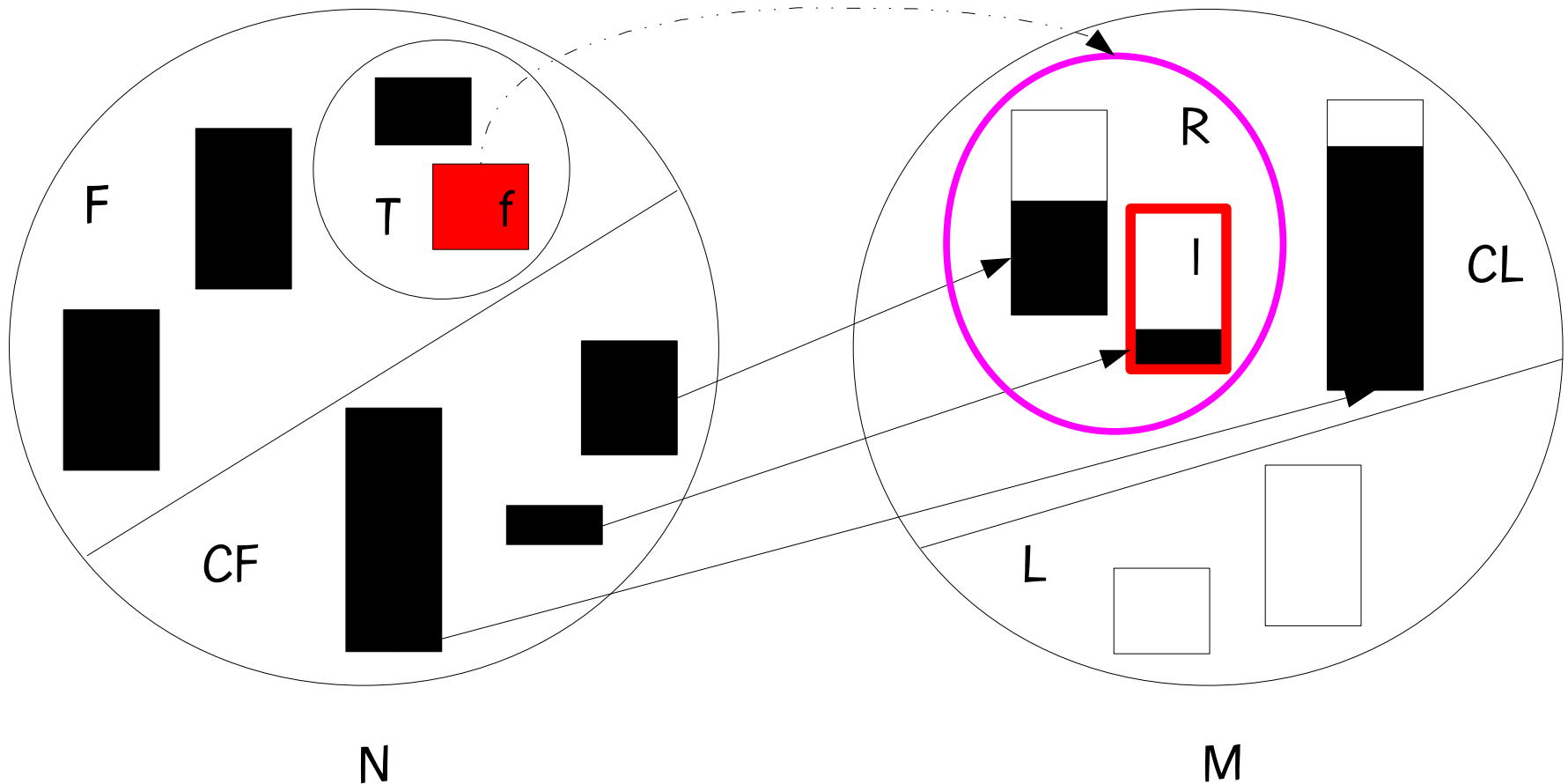
Randomly select a **facility  $f \in T$**  favoring facilities that have high demand and high flows to other facilities.

# Procedure to select a location from CL (step 1)



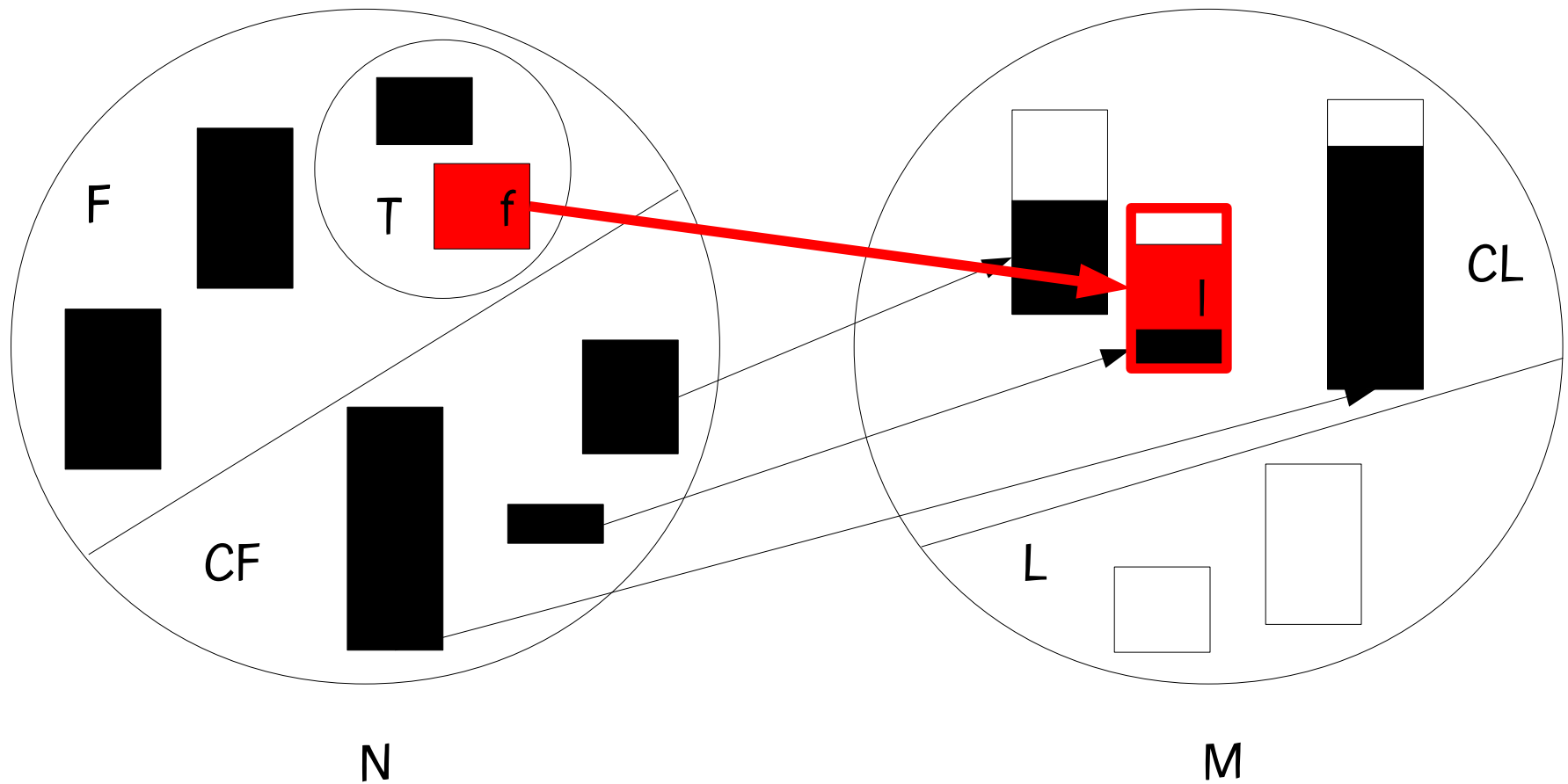
1. Let set  $R$  to be all locations in  $CL$  having slack greater than or equal to demand of facility  $f$ ;

# Procedure to select a location from CL (step 2)



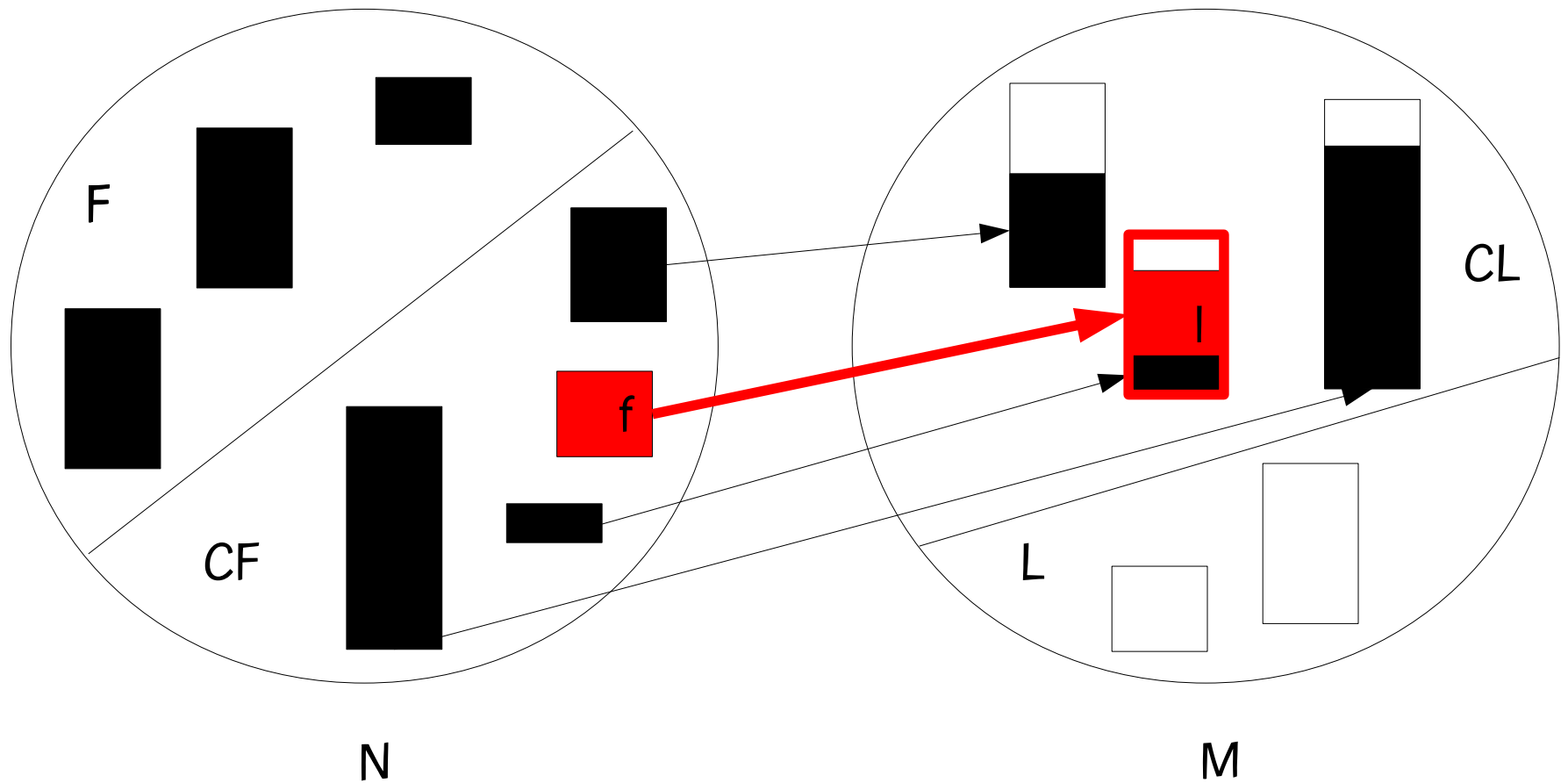
2. Randomly select a location  $I \in R$  favoring those having high available capacity and those close to high-capacity locations in CL;

# Assignment procedure



Assign facility **f** to location **l**

# Assignment procedure



Update sets  $F$ ,  $CF$ , and slack of location  $I$

# Considerations about the construction procedure

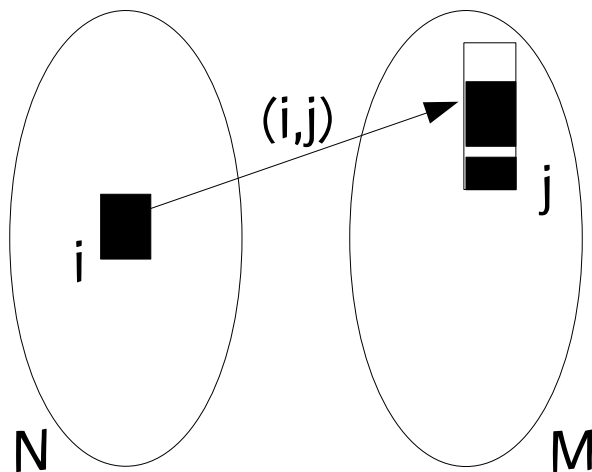
- The procedure is not guaranteed to produce a feasible solution.
- To address this difficulty, the construction procedure is repeated a maximum number of times or until all facilities are assigned (i.e. until  $F=\emptyset$ ).
- At start of construction, a location  $l$  from  $L$  is selected with probability proportional to its capacity. Location  $l$  is placed in  $CL$ .

# Local search

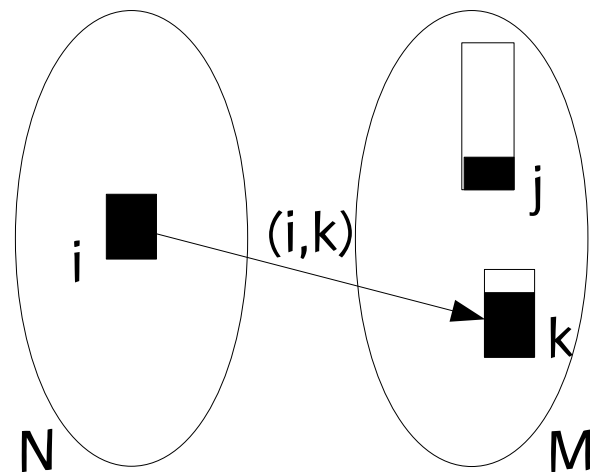
# Local search

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  $p$



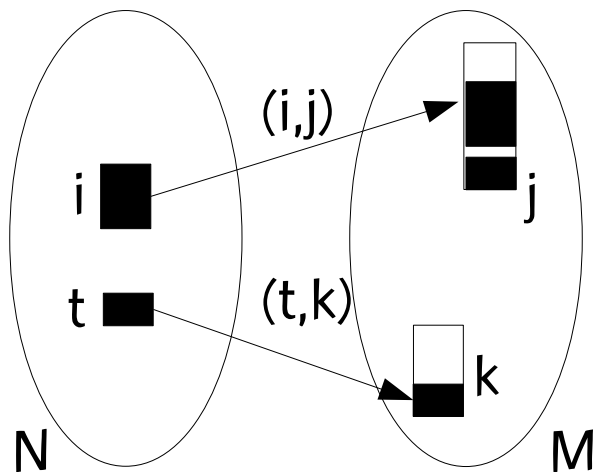
1-move neighbor of  $p$

# Local search

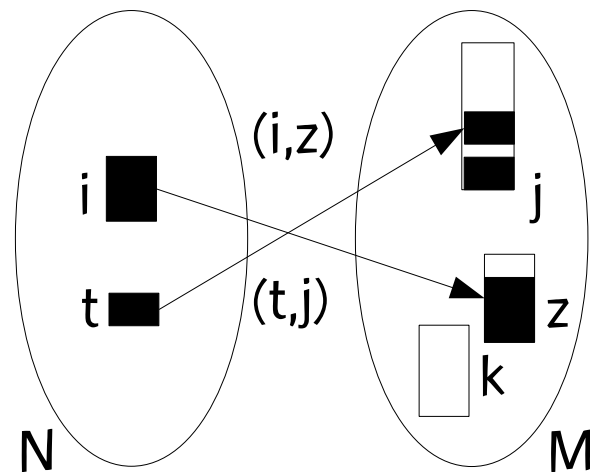
1-move and 2-move neighborhoods from solution  $p$  are used in our local search.

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

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

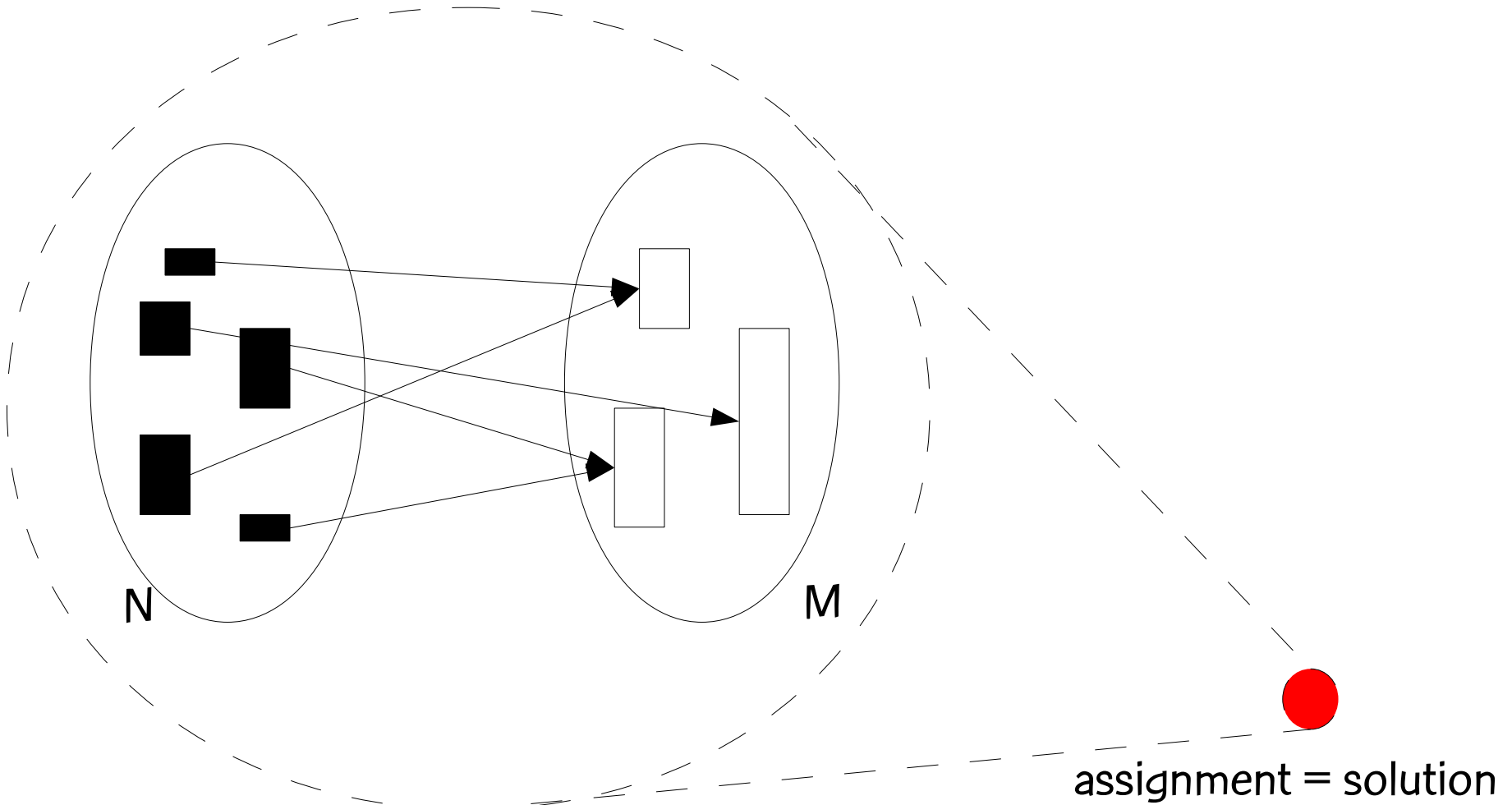


solution  $p$



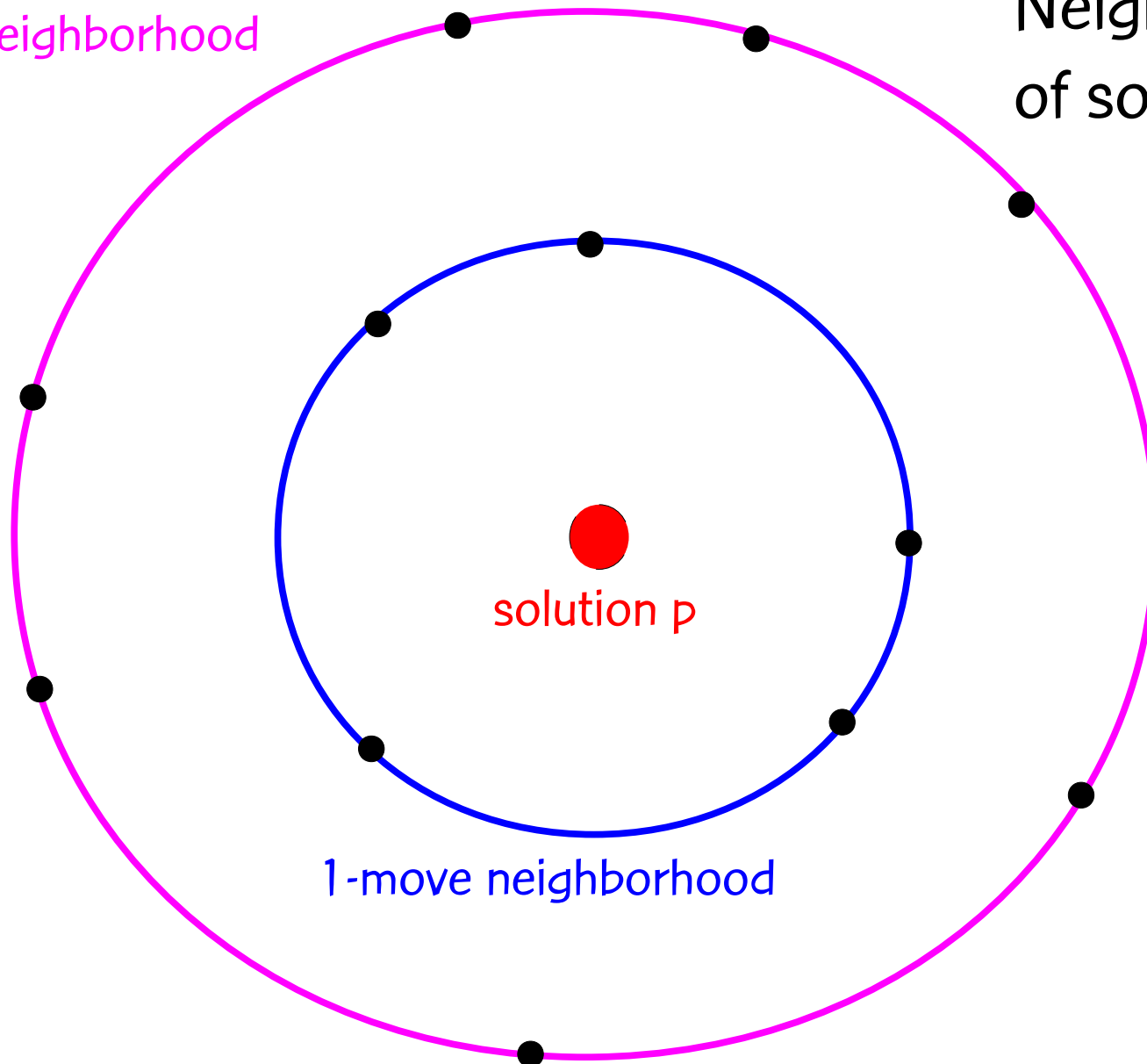
2-move neighbor of  $p$

# Assignment representation



2-move neighborhood

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

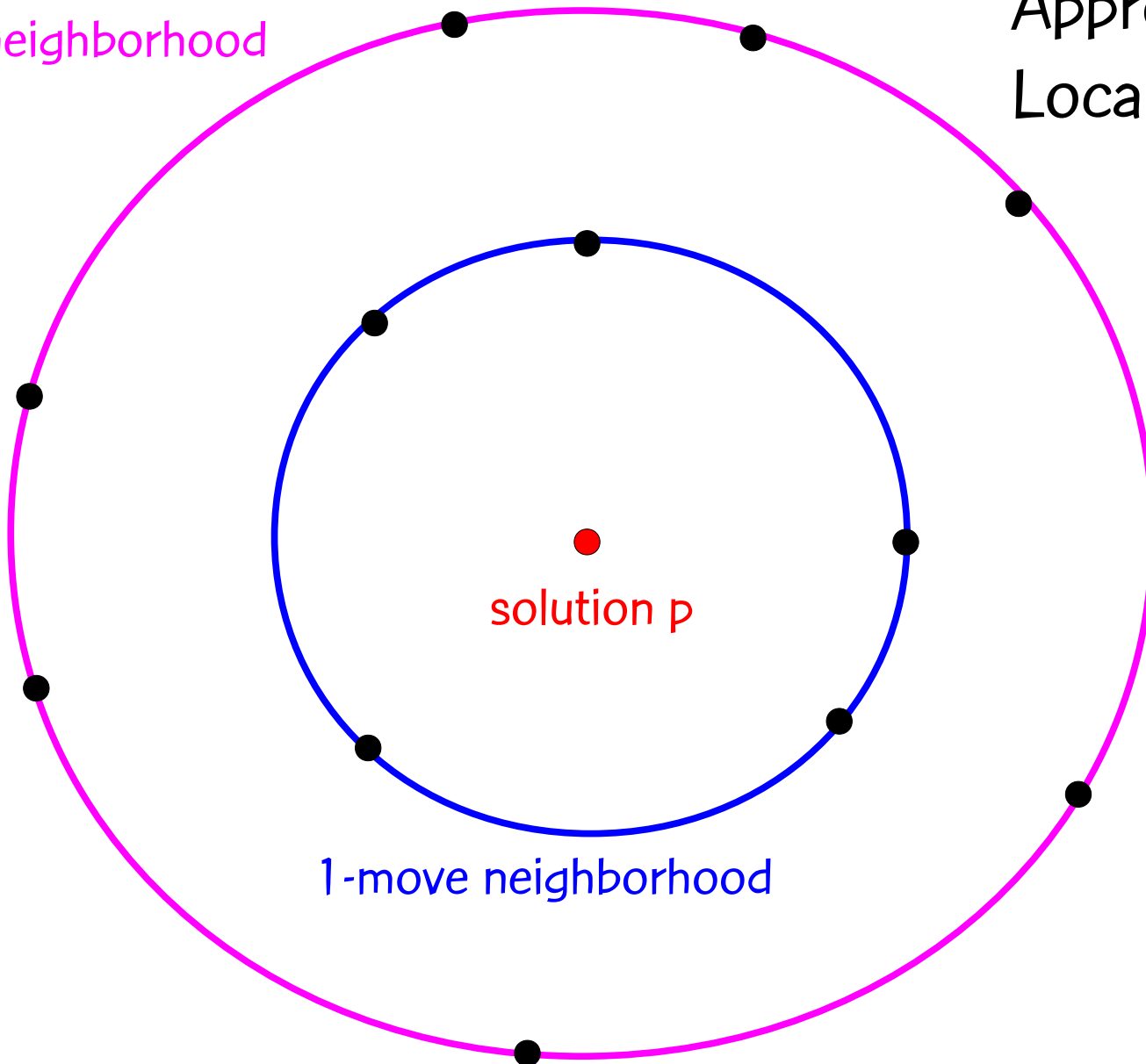
Neighborhoods can be very large for best improvement

Local search can take very long

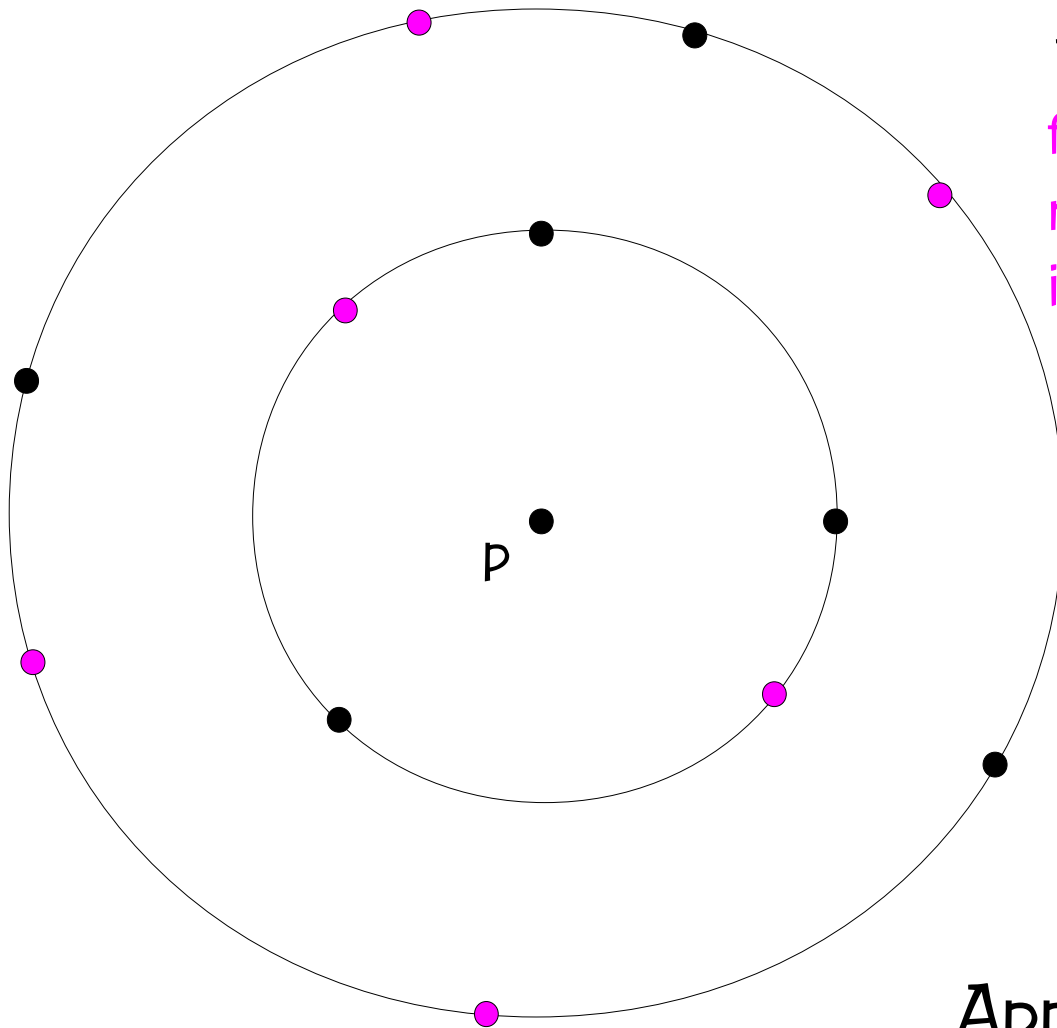
Tradeoff between best & first improvement: sample the neighborhood of solution  $p$ .

2-move neighborhood

Approximate  
Local Search

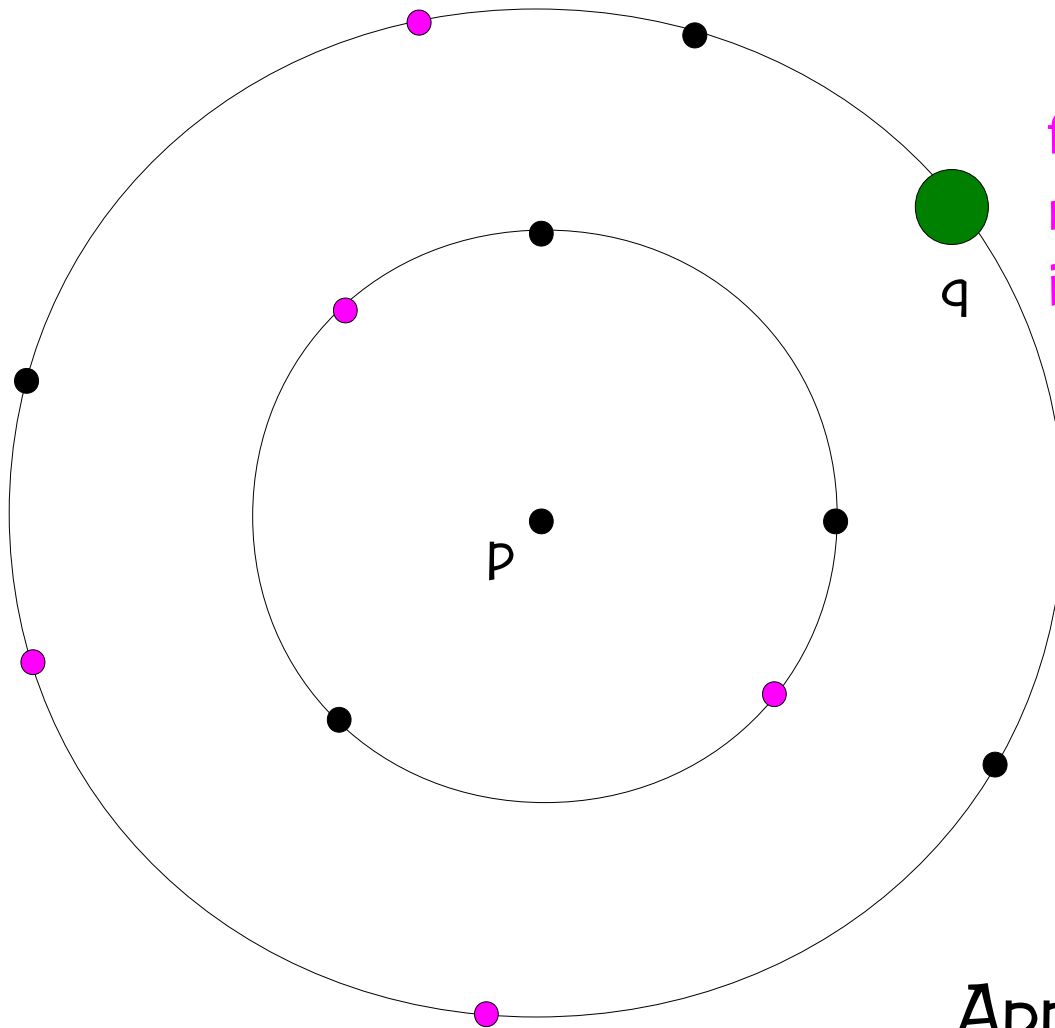


1-move neighborhood



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

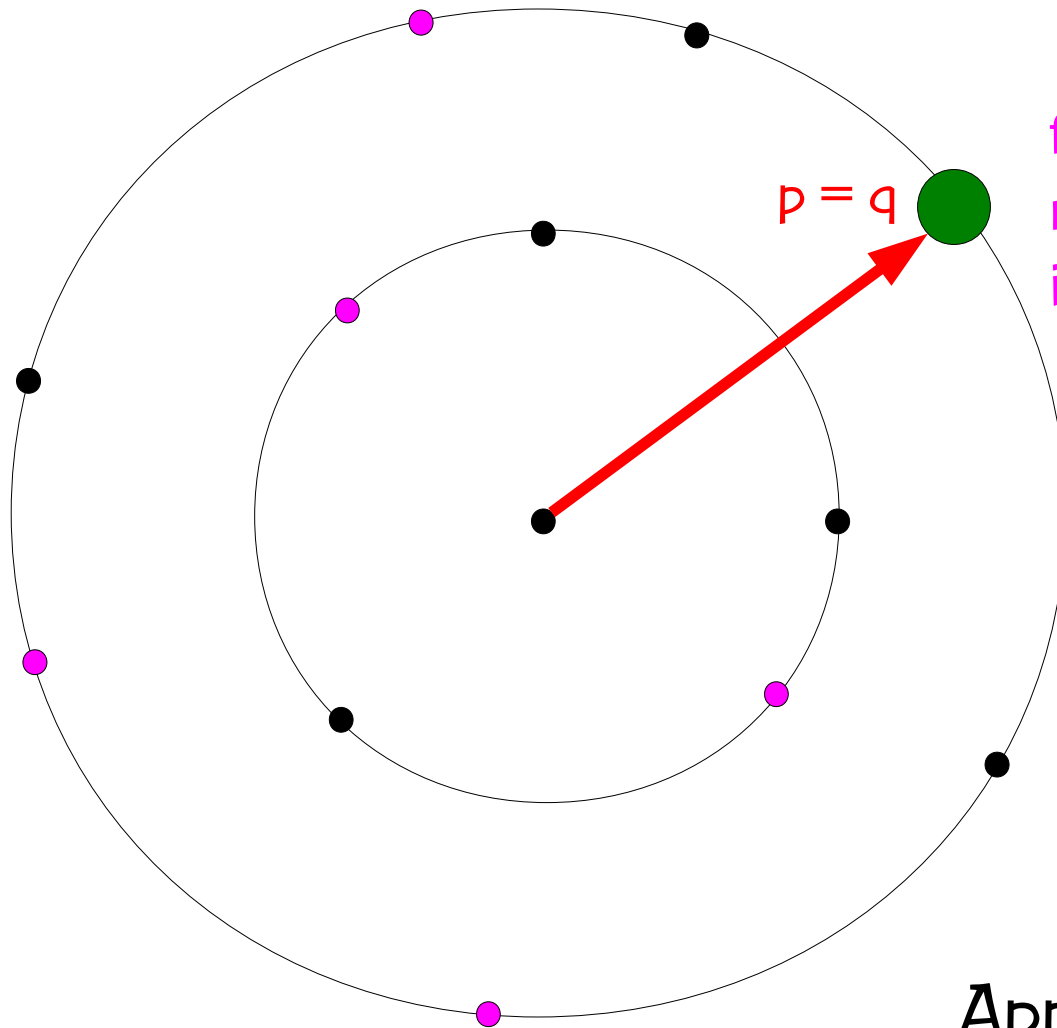
## Approximate Local Search



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



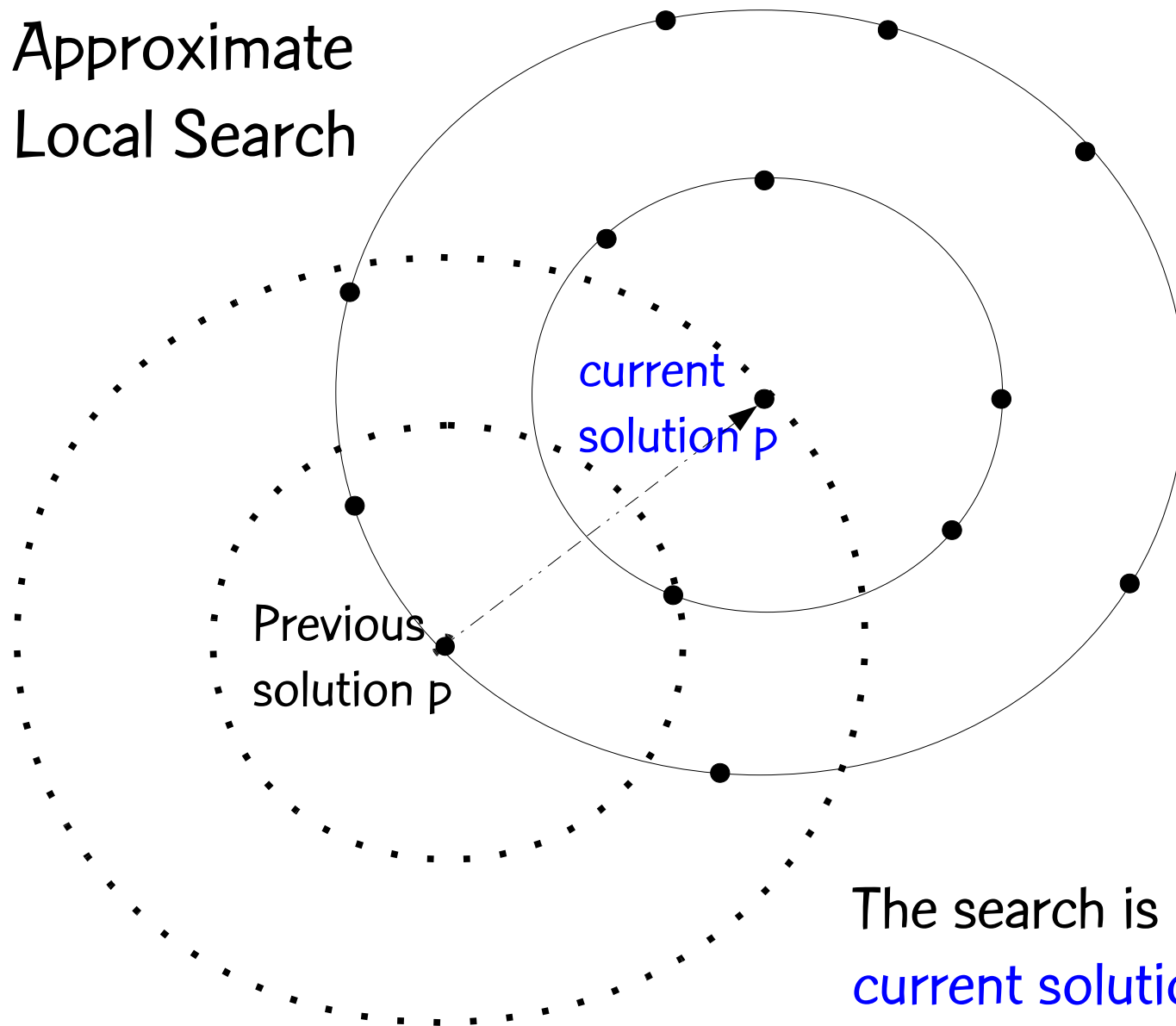
1. Sample  $k$  improving solutions from 1-move and 2-move neighborhood of  $p$  and place them in an elite set  $E$ .

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

3. Update  $p = q$

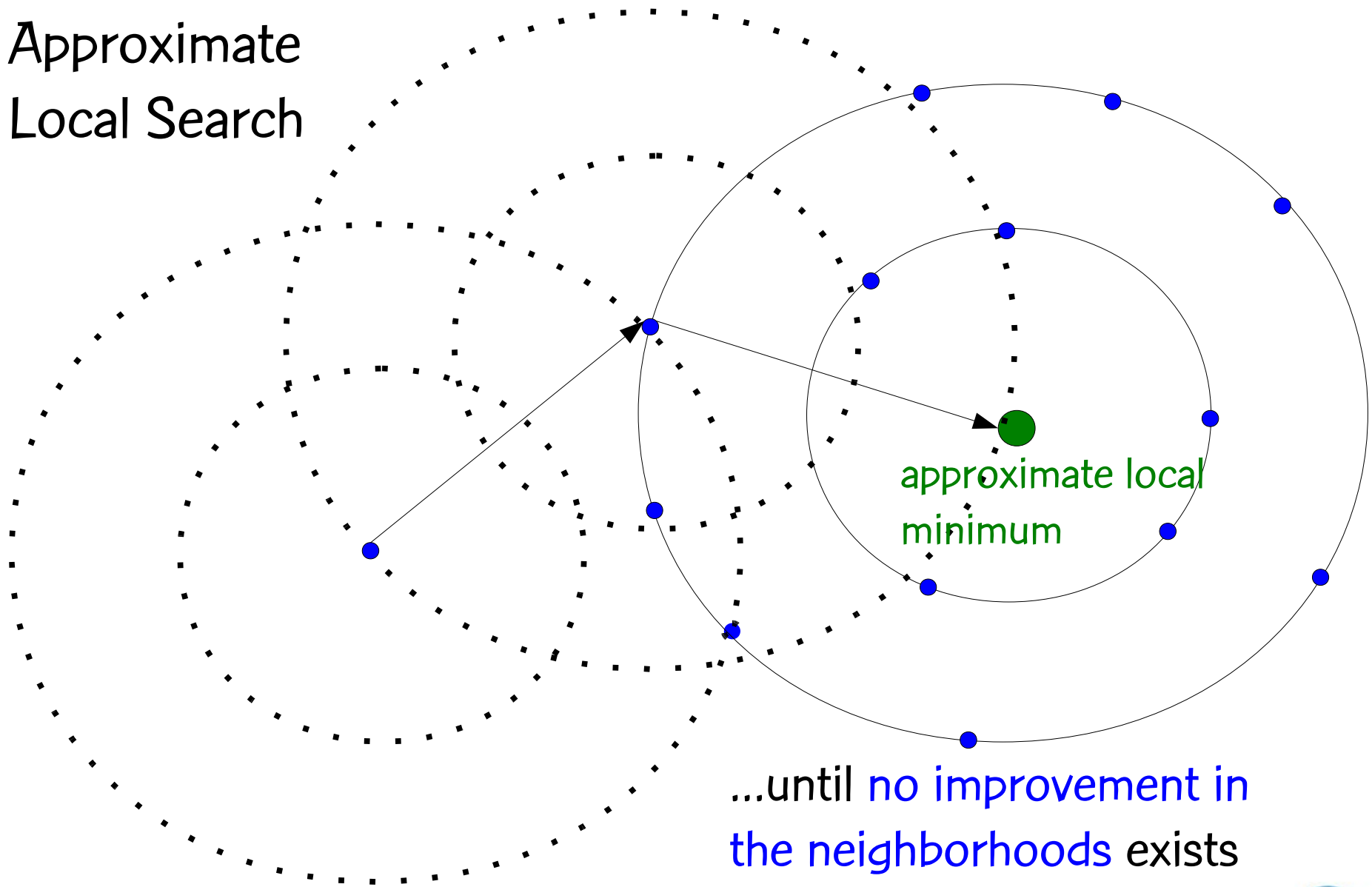
## Approximate Local Search

# Approximate Local Search



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

# Approximate Local Search



# Path-relinking

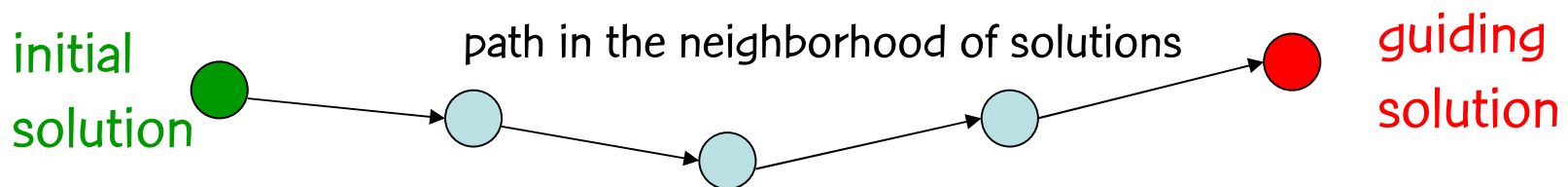
July 2009

GRASP-PR for generalized quadratic assignment problem



# 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:

initial  
solution ●

● guiding  
solution

# 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.

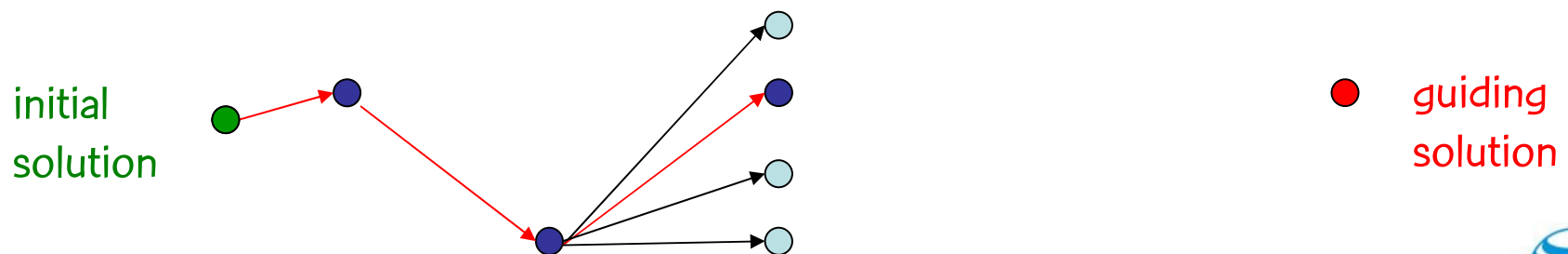
At each step, all moves that incorporate attributes of the guiding solution are evaluated and the best move is selected:



# Path-relinking

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

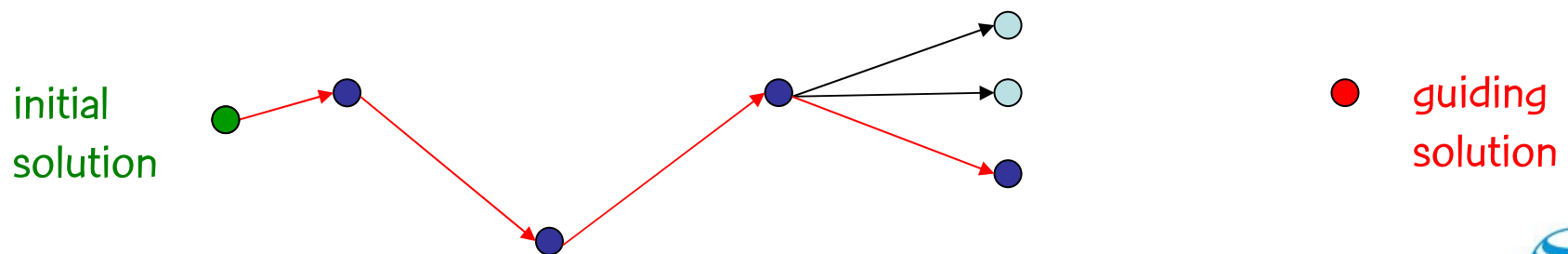
At each step, all moves that incorporate attributes of the guiding solution are evaluated and the best move is selected:



# Path-relinking

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

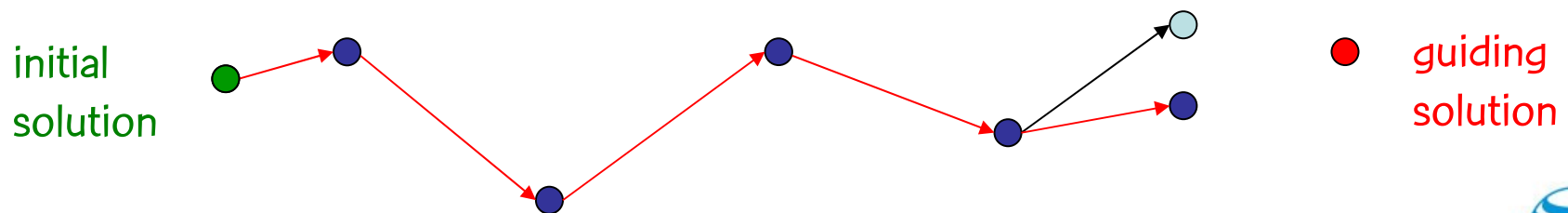
At each step, all moves that incorporate attributes of the guiding solution are evaluated and the best move is selected:



# Path-relinking

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

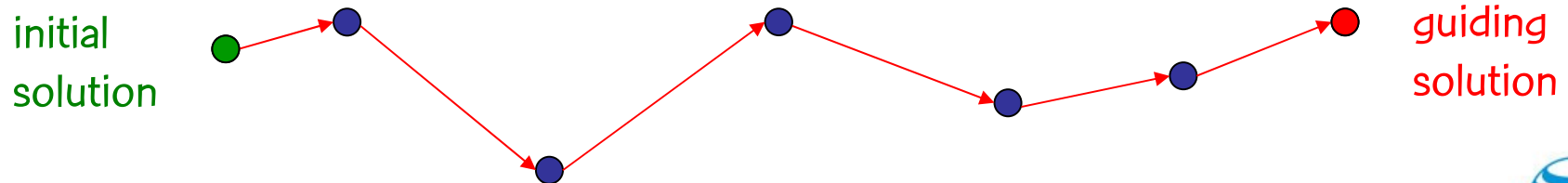
At each step, all moves that incorporate attributes of the guiding solution are evaluated and the best move is selected:



# Path-relinking

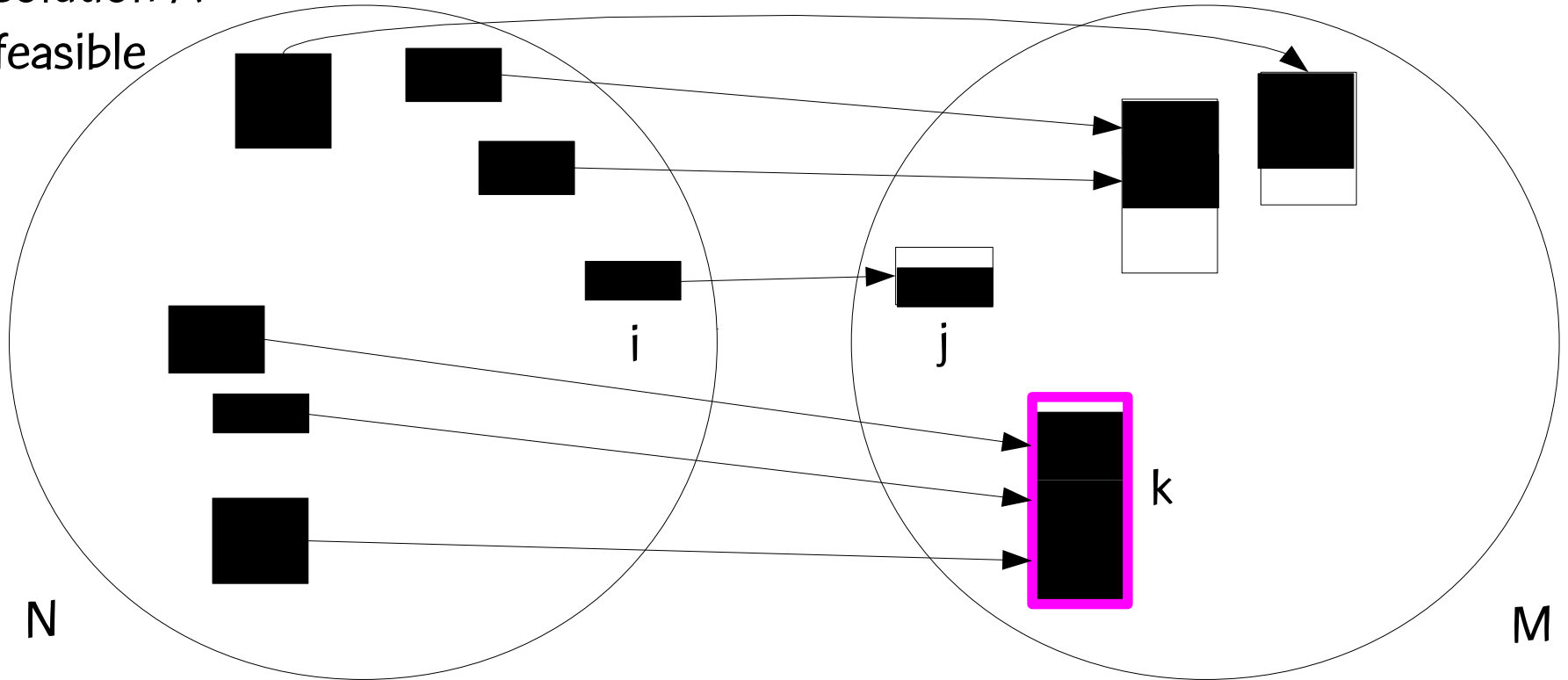
Path is generated by selecting moves that introduce in the initial solution attributes of the guiding solution.

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

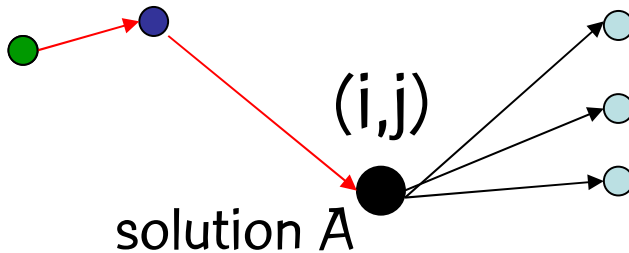


# Infeasibility in path-relinking for GQAP

solution A  
feasible



initial  
solution



● guiding  
solution  
 $(i,k)$



at&t

Your world. Delivered.

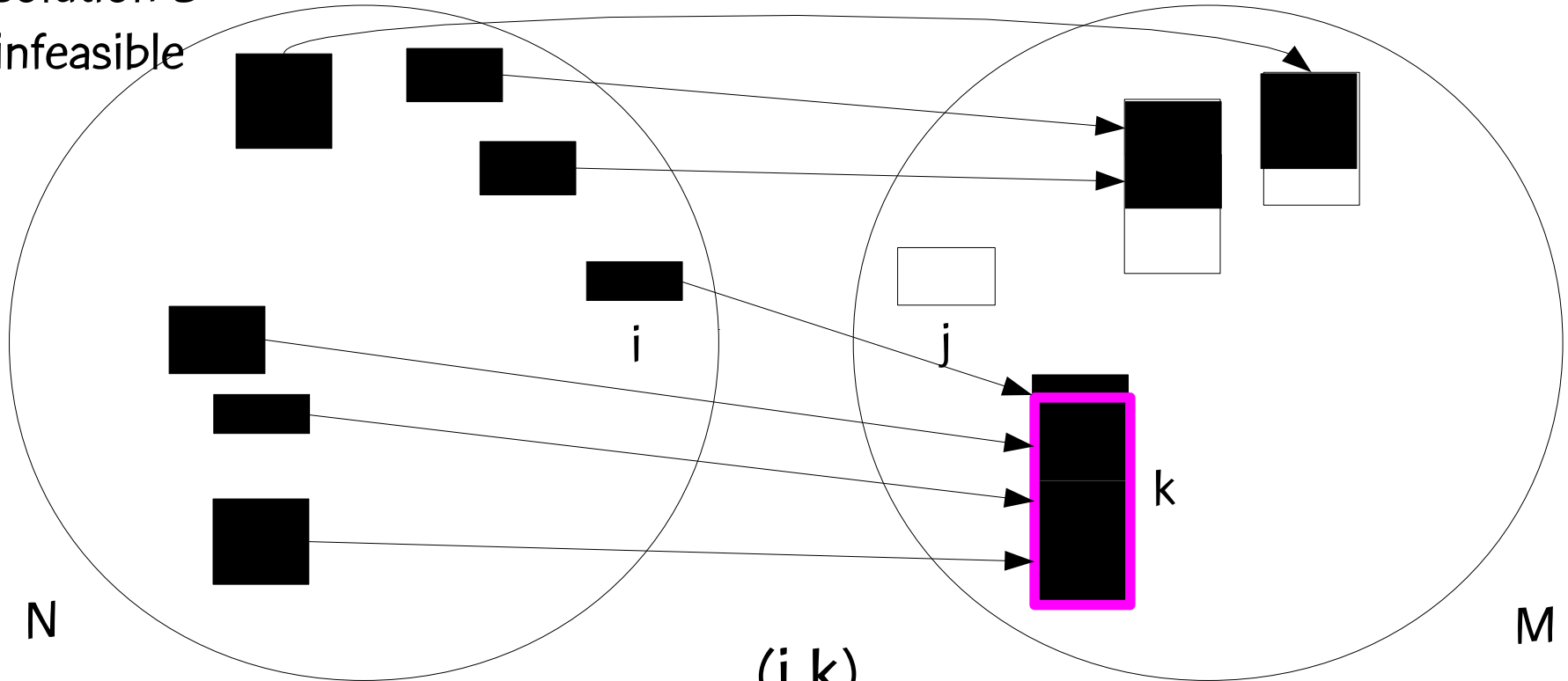
July 2009

GRASP-PR for generalized quadratic assignment problem

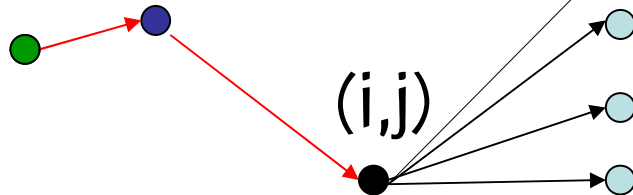
# Infeasibility in path relinking for GQAP

solution B

infeasible



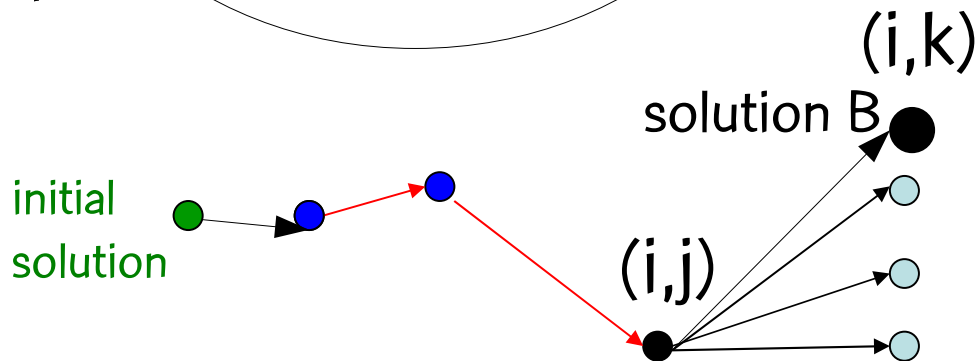
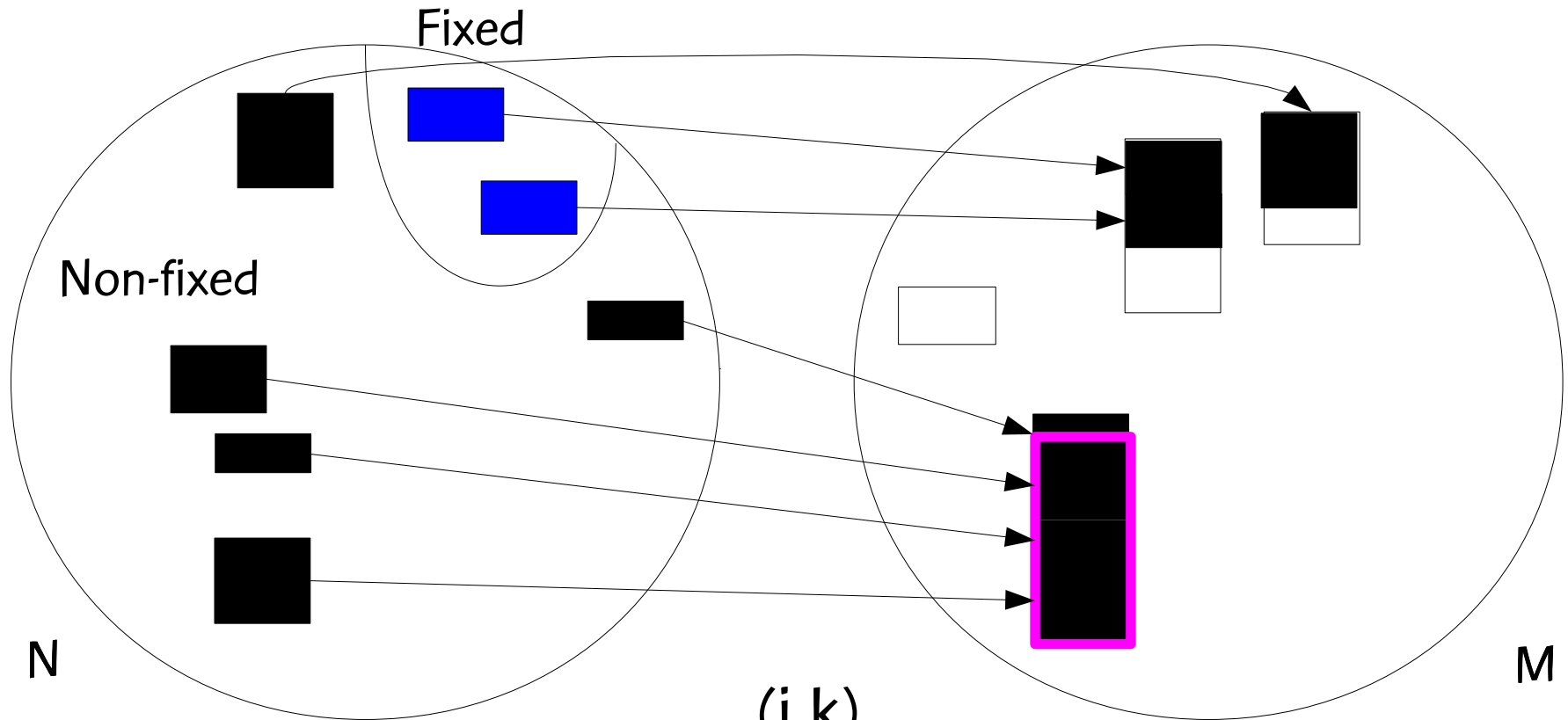
initial  
solution



● guiding  
solution  
(i,k)

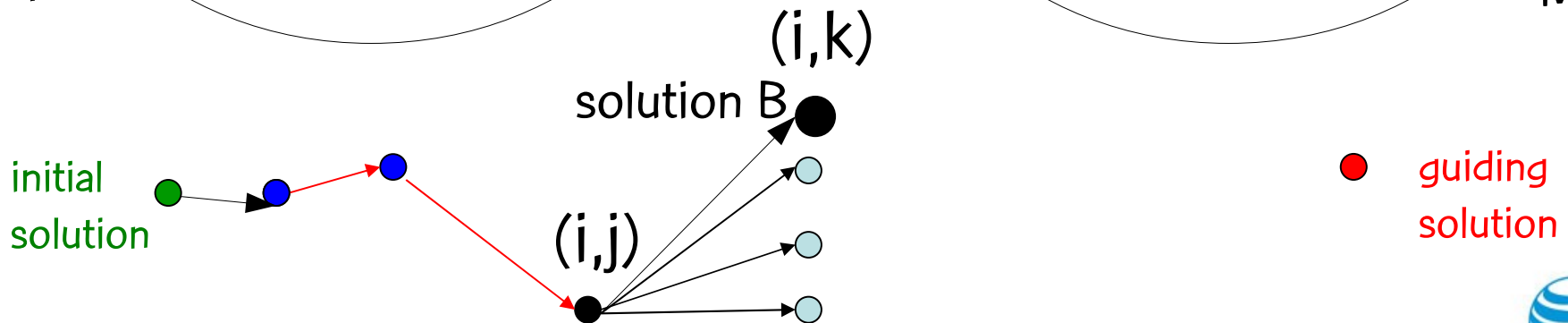
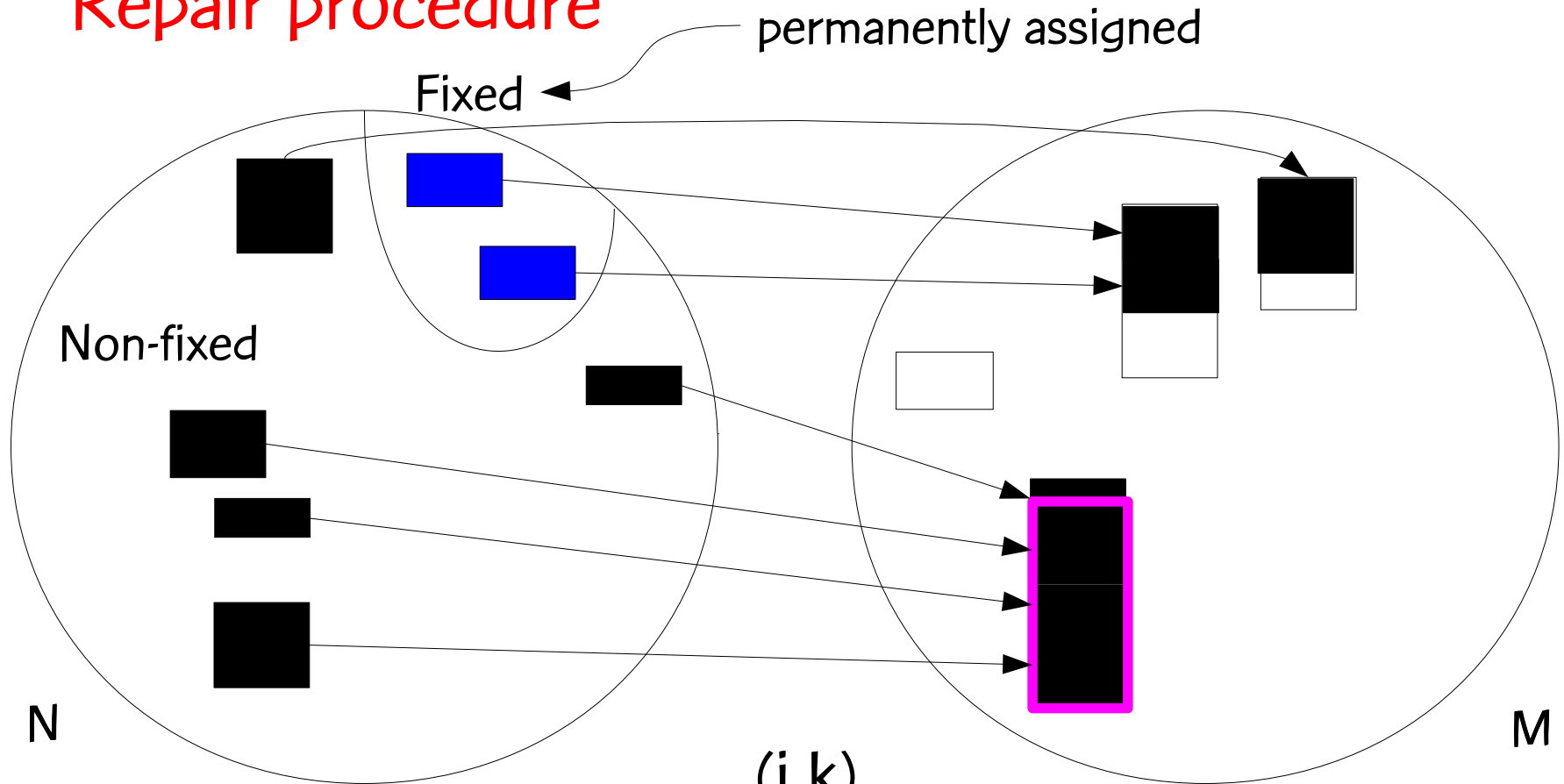


# Repair procedure



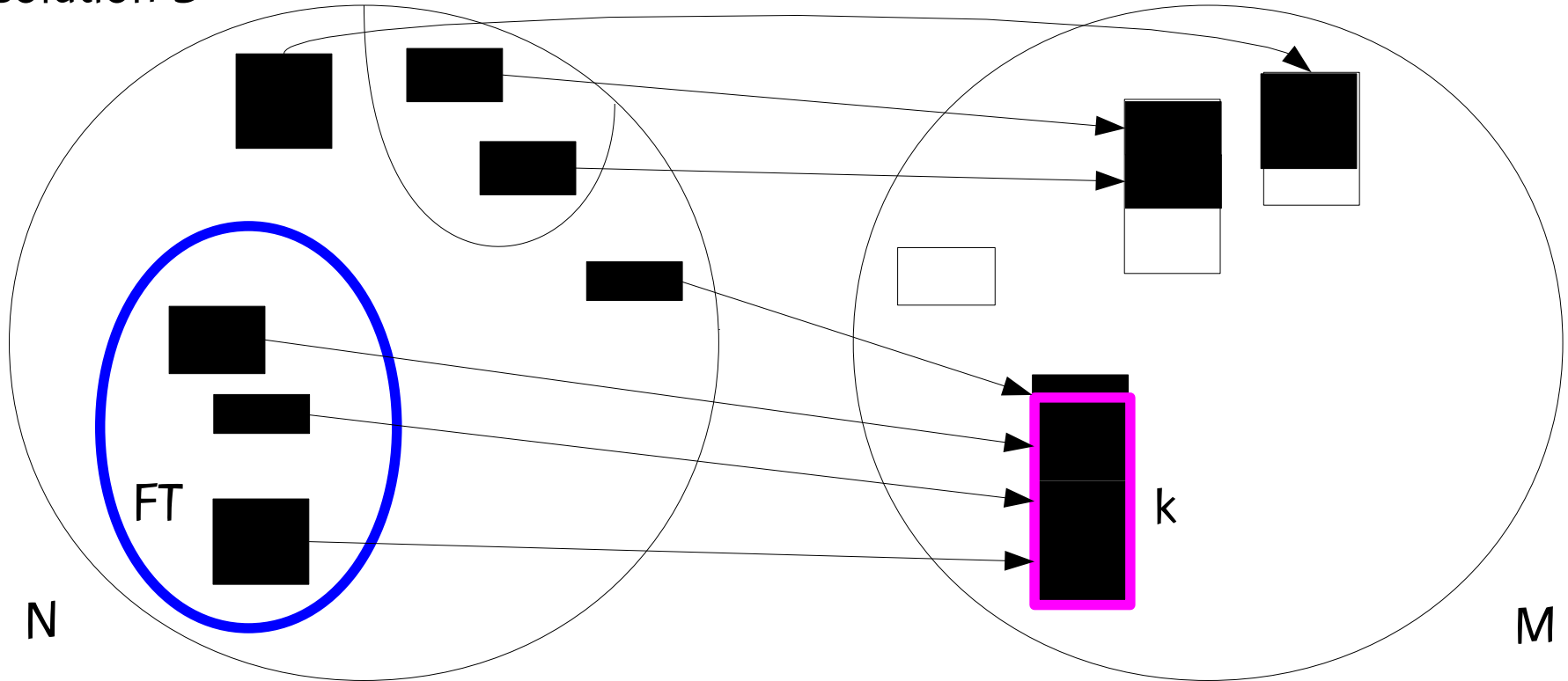
● guiding solution

# Repair procedure



# Repair procedure

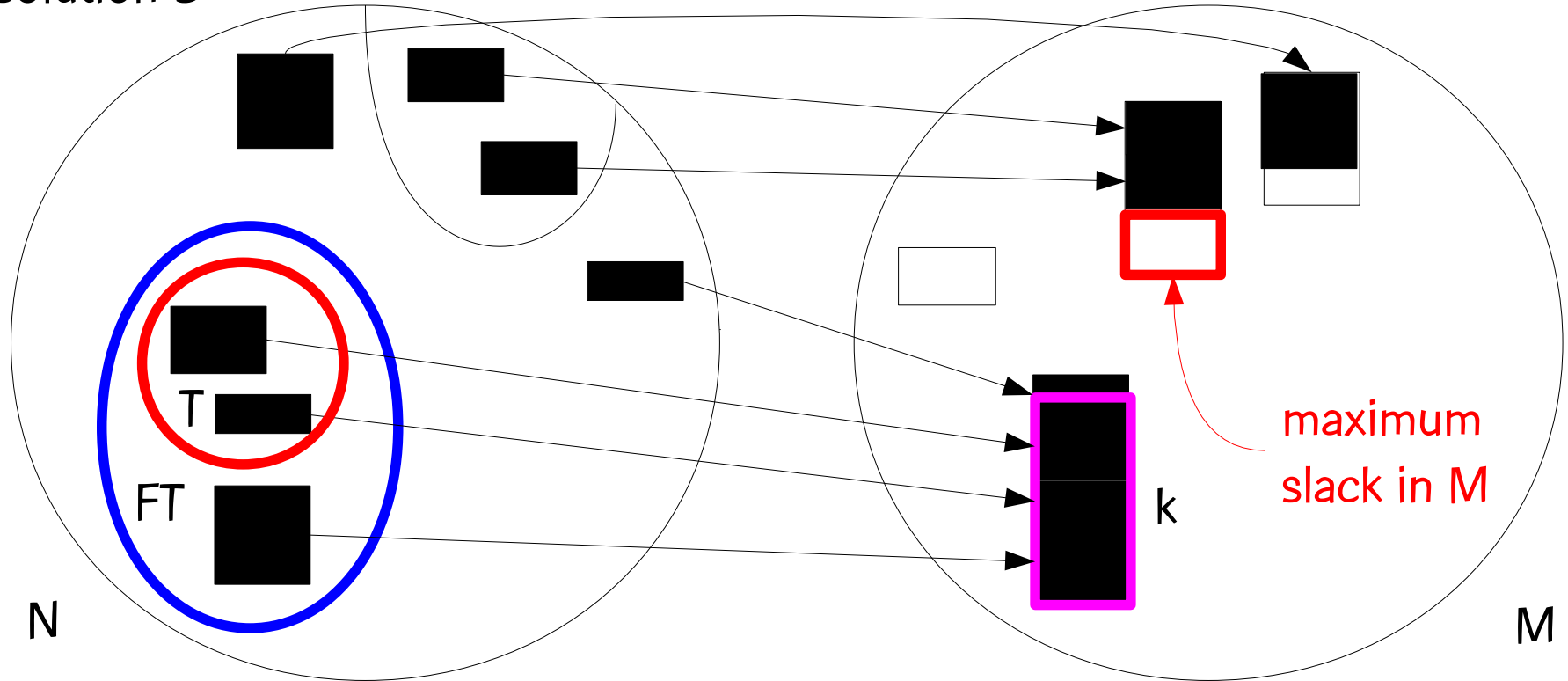
solution B



1. Set  $FT \subseteq \text{non-Fixed}$ : all facilities in solution B assigned to location k

# Repair procedure

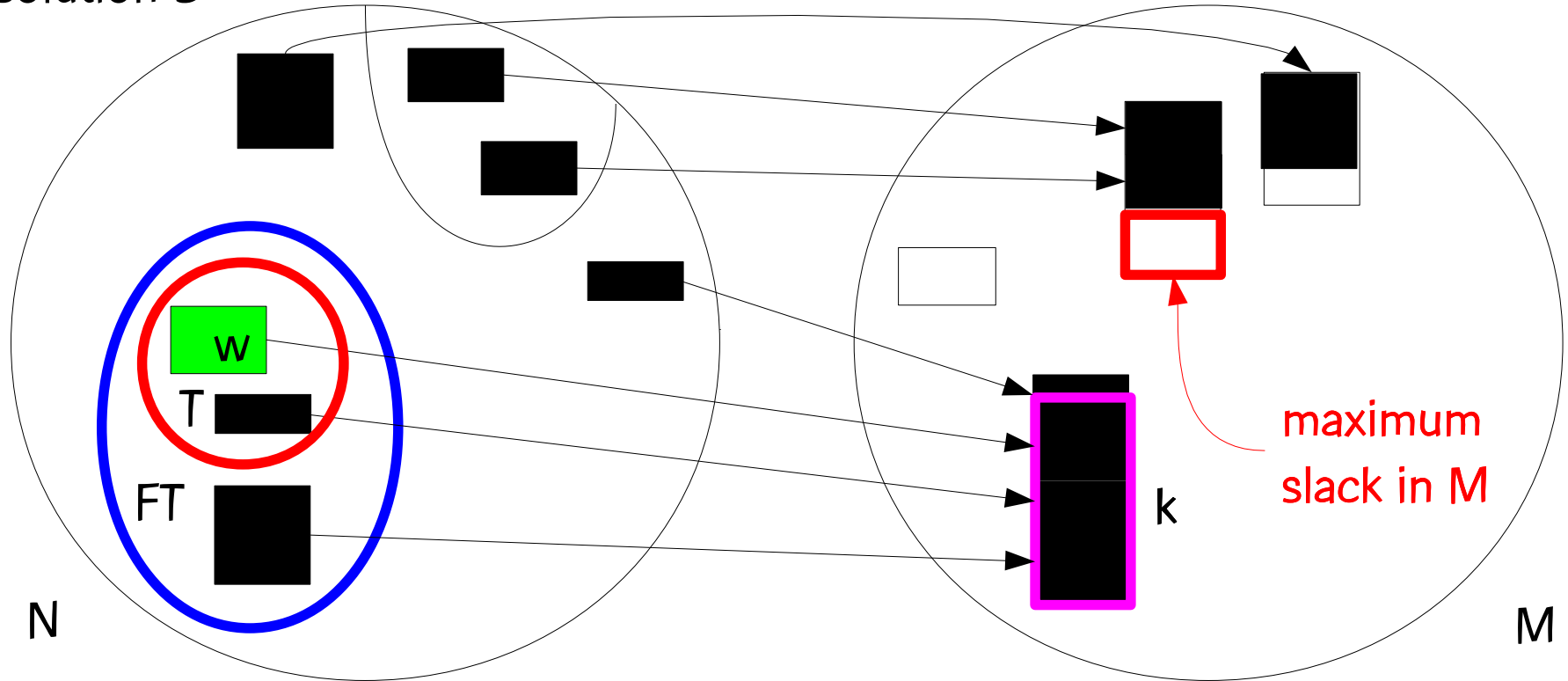
solution B



1. Set  $FT \subseteq \text{non-Fixed}$ : 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

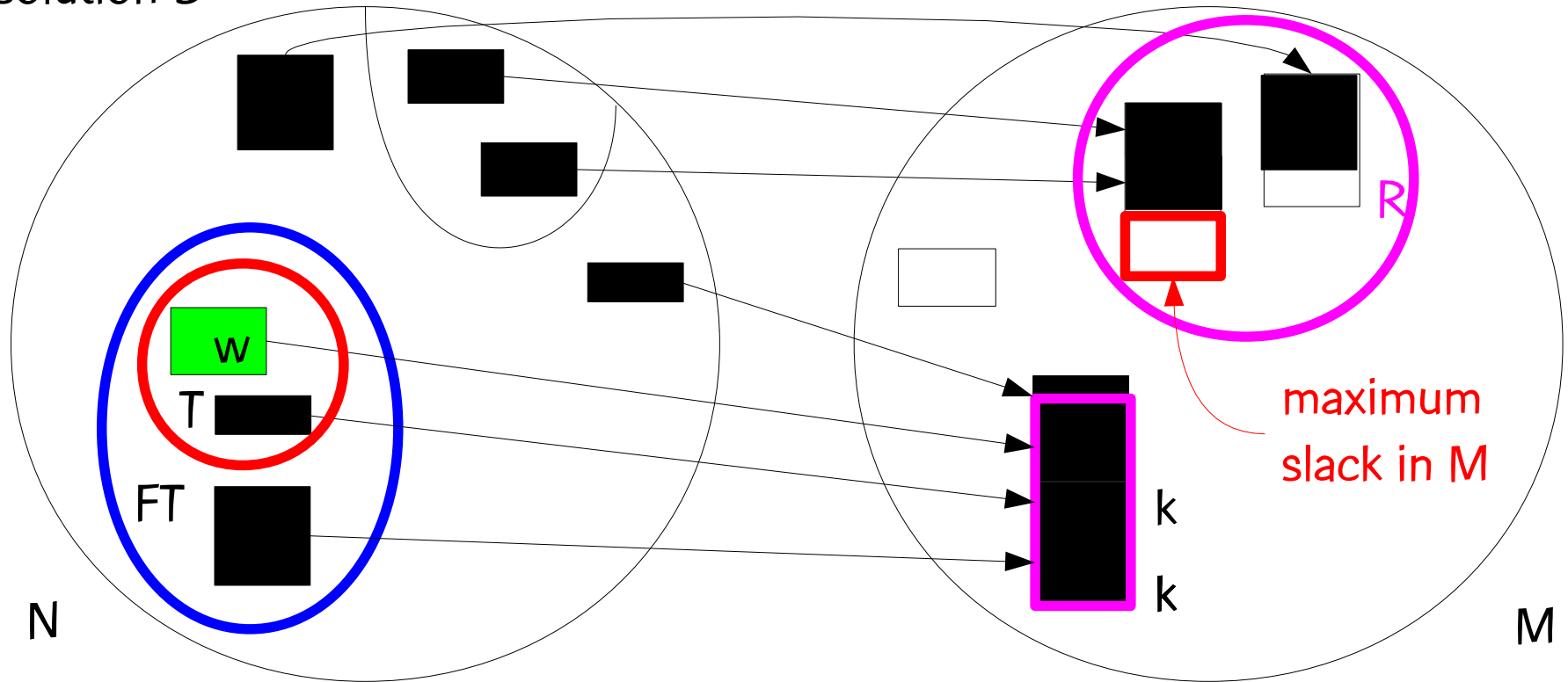
# Repair procedure

solution B



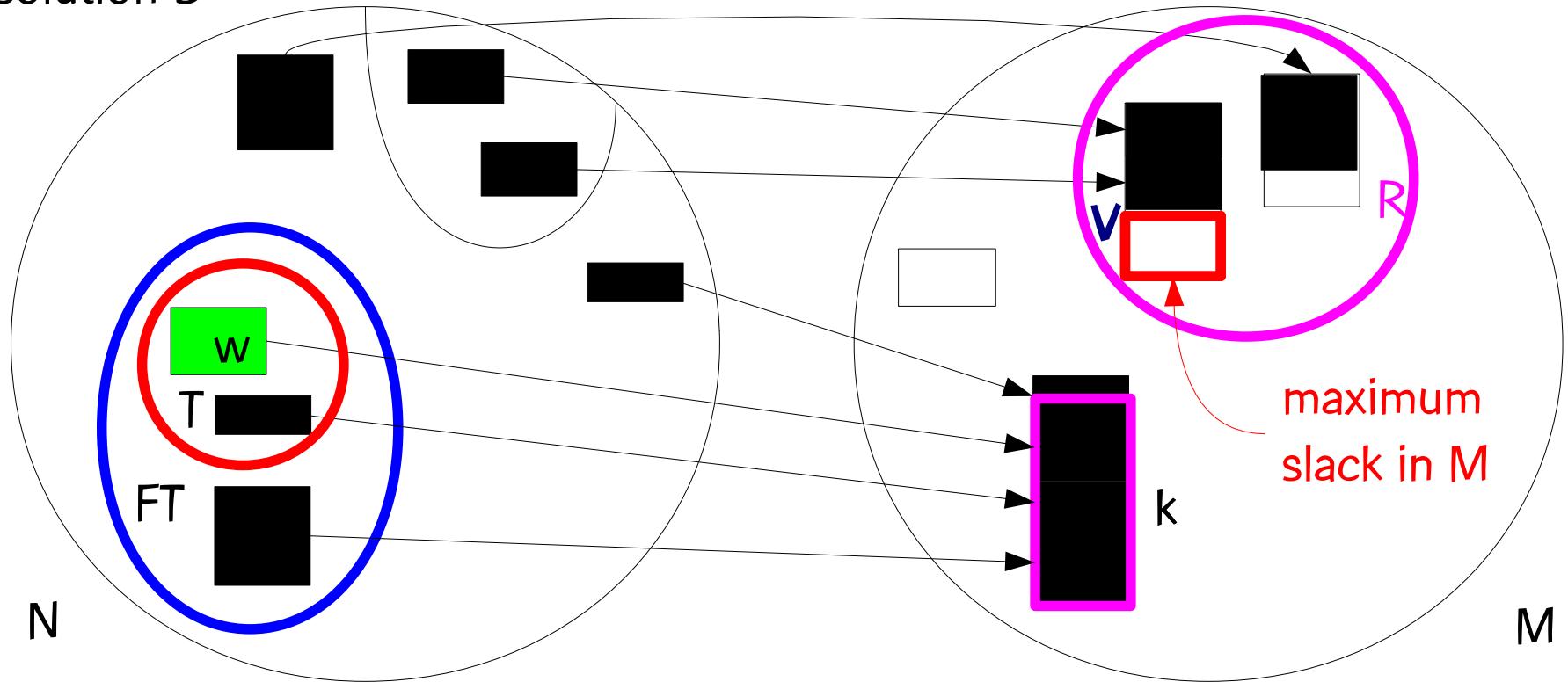
1. Set  $FT \subseteq$  non-Fixed: all facilities in solution B assigned to location k
2. Set  $T \subseteq FT$ : all facilities in B with demand  $\leq$  maximum slack in M
3. Randomly select a facility  $w \in T$  favoring those with higher demand

## solution B



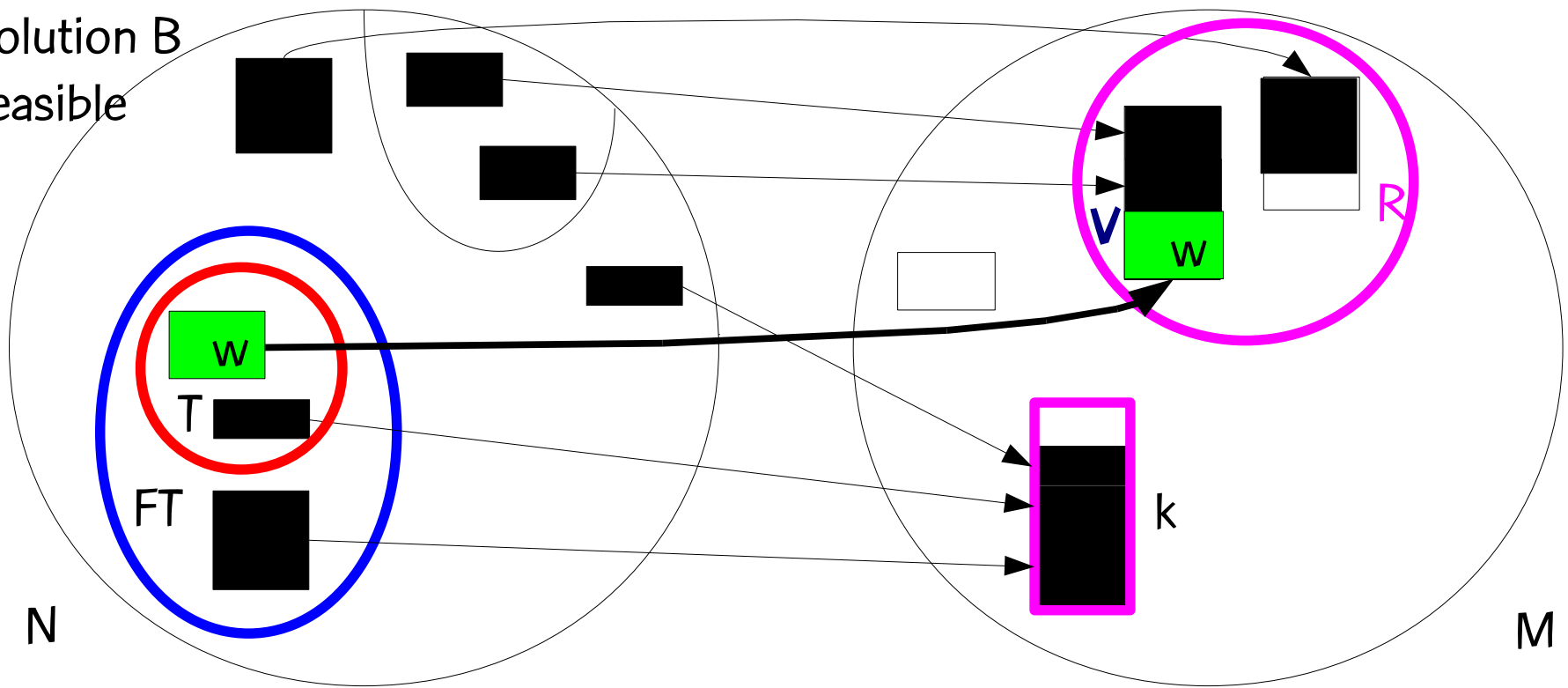
1. Set  $FT \subseteq$  non-Fixed: all facilities in solution B assigned to location k
2. Set  $T \subseteq FT$ : all facilities in B with demand  $\leq$  maximum slack in M
3. Randomly select a facility  $w \in T$  favoring those with higher demand
4. Set  $R \subseteq M$ : all locations having slack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

## solution B



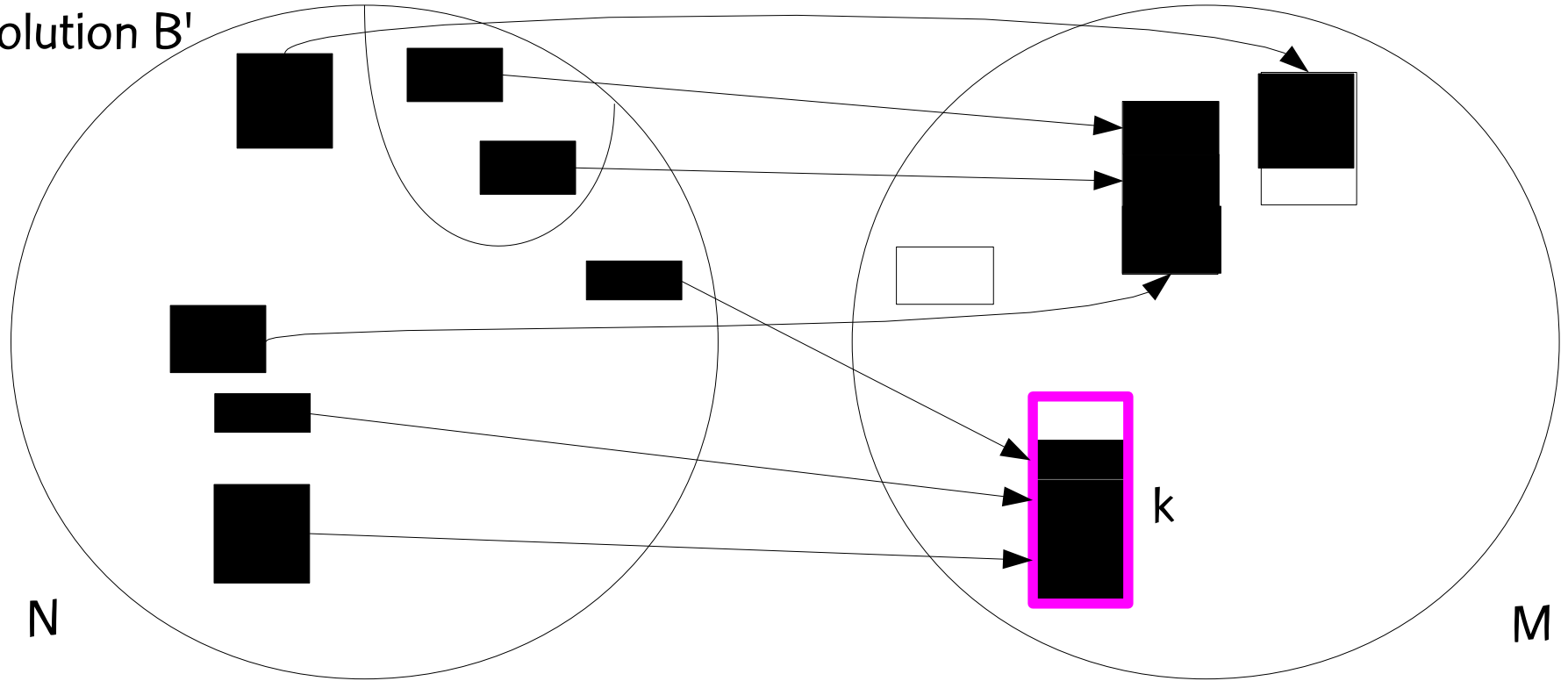
1. Set  $FT \subseteq$  non-Fixed: all facilities in solution B assigned to location k
2. Set  $T \subseteq FT$ : all facilities in B with demand  $\leq$  maximum slack in M
3. Randomly select a facility  $w \in T$  favoring those with higher demand
4. Set  $R \subseteq M$ : all locations having slack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

solution B  
feasible

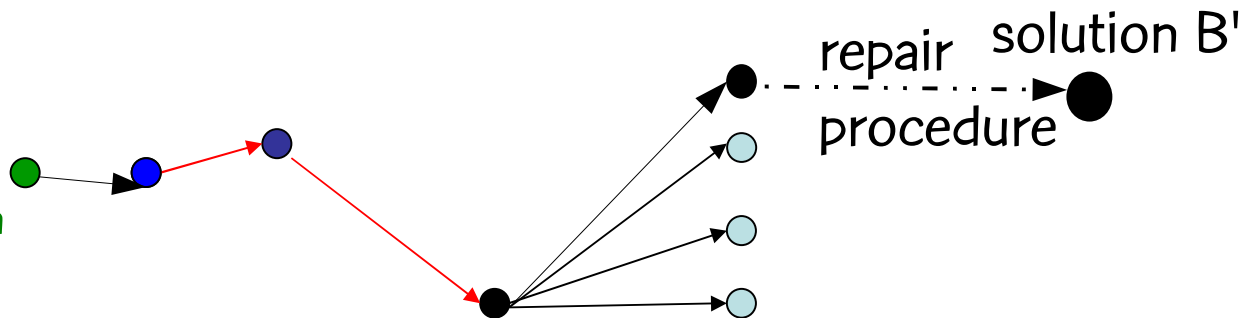


1. Set  $FT \subseteq$  non-Fixed: all facilities in solution B assigned to location k
2. Set  $T \subseteq FT$ : all facilities in B with demand  $\leq$  maximum slack in M
3. Randomly select a facility  $w \in T$  favoring those with higher demand
4. Set  $R \subseteq M$ : all locations having slack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

solution B'



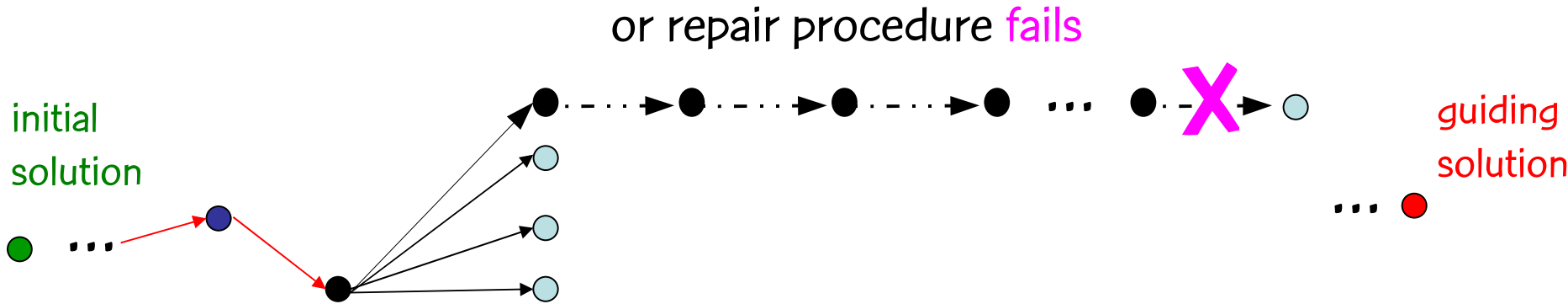
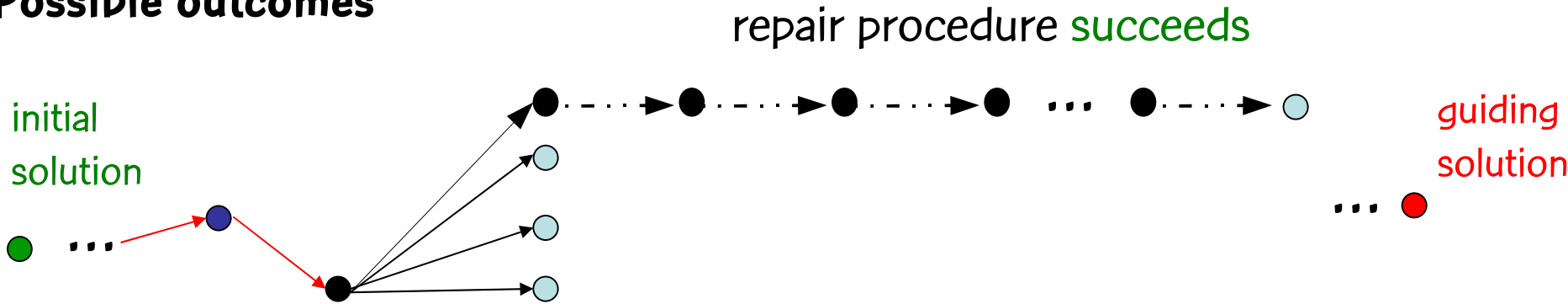
initial  
solution



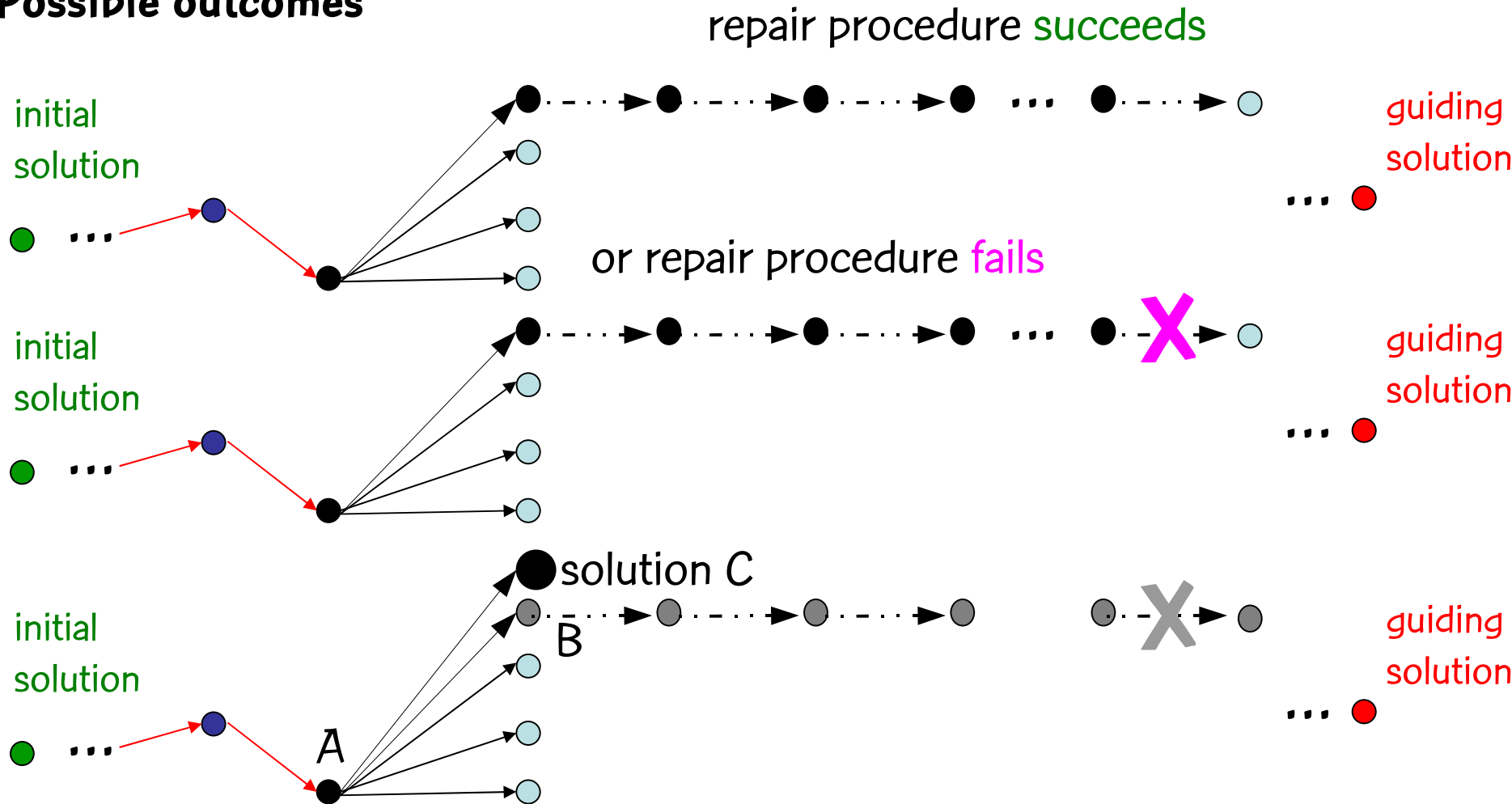
● guiding  
solution



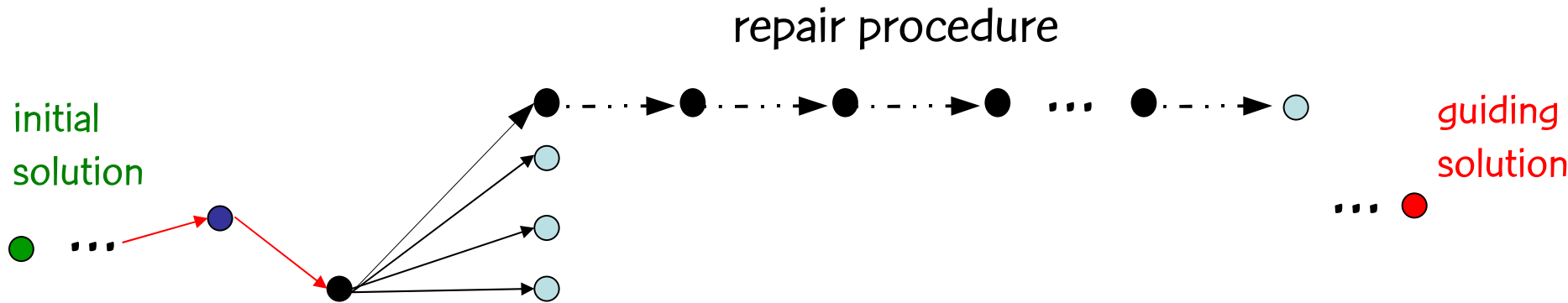
## Possible outcomes



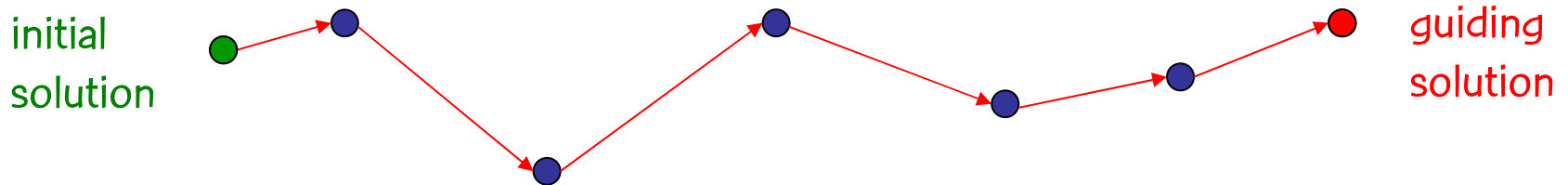
## Possible outcomes

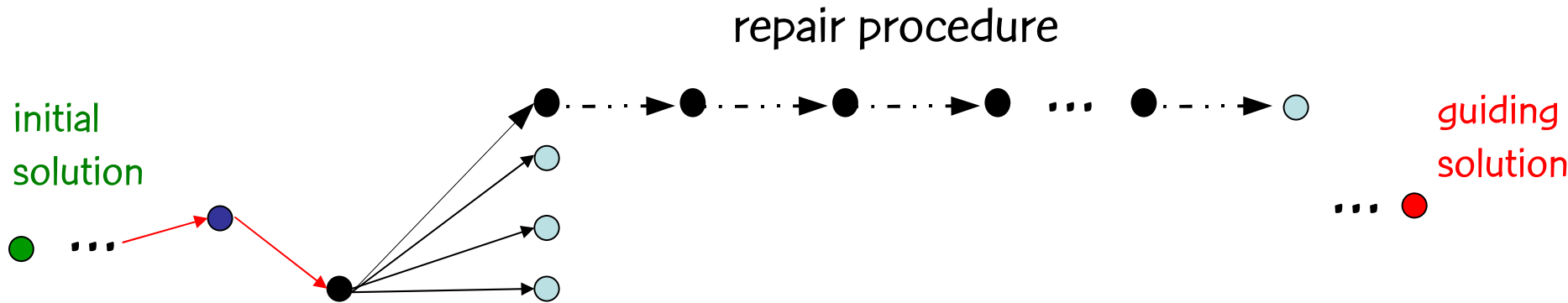


Repeat the repair procedure on solution B a maximum number of times. If a feasible solution is not found, discard B and move to solution C

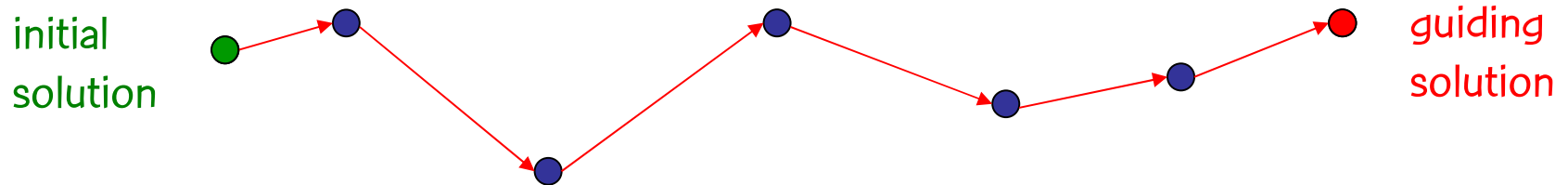


So, instead of a path with feasible solution in one single step ...

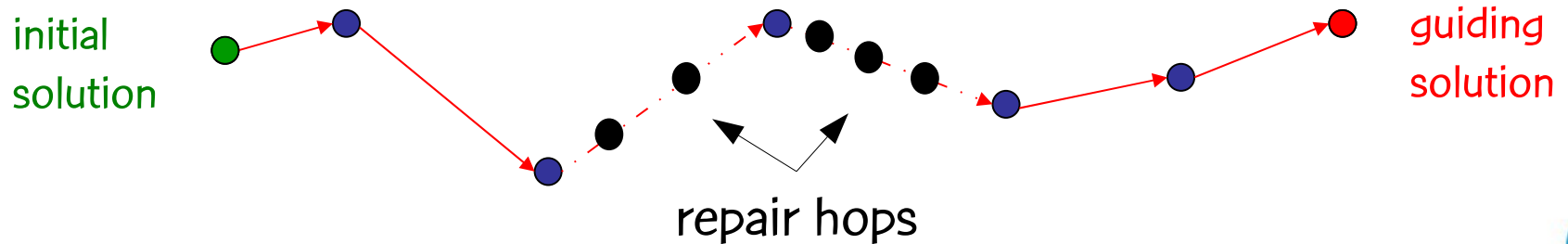




So, instead of a path with feasible solution in one single step ...



We have now a path with eventual intermediate repair hops



# Experimental results

# Test environment

Dell PE1950 computer with a dual quad core 2.66 GHz Intel Xeon processors an 16 GB of Memory

Red Hat Linux version 5.1.19.6

Java language, Javac compiler ver.1.6.0-05

Random-number generator: Mersenne Twister algorithm (Matsumoto and Nishimura, 1998) from the COLT library

# Test environment

## Instances:

From Elloumi et al. (2003), Lee and Ma (2005), and Cordeau et al. (2006):  
10 to 50 facilities and 3 to 20 locations.

## Experimental Design:

For each instance we made 200 independent runs of GRASP-PR. Each run stopped when a solution value as good as the best in the literature was found.

## Statistics:

Minimum, maximum, average times, and standard deviation.

Time for 95% of the runs to find solutions as good as the literature.

# Parameter tuning for GRASP-PR

Instance: 50-10-95 (Cordeau et al., 2006).

Strategies tested:

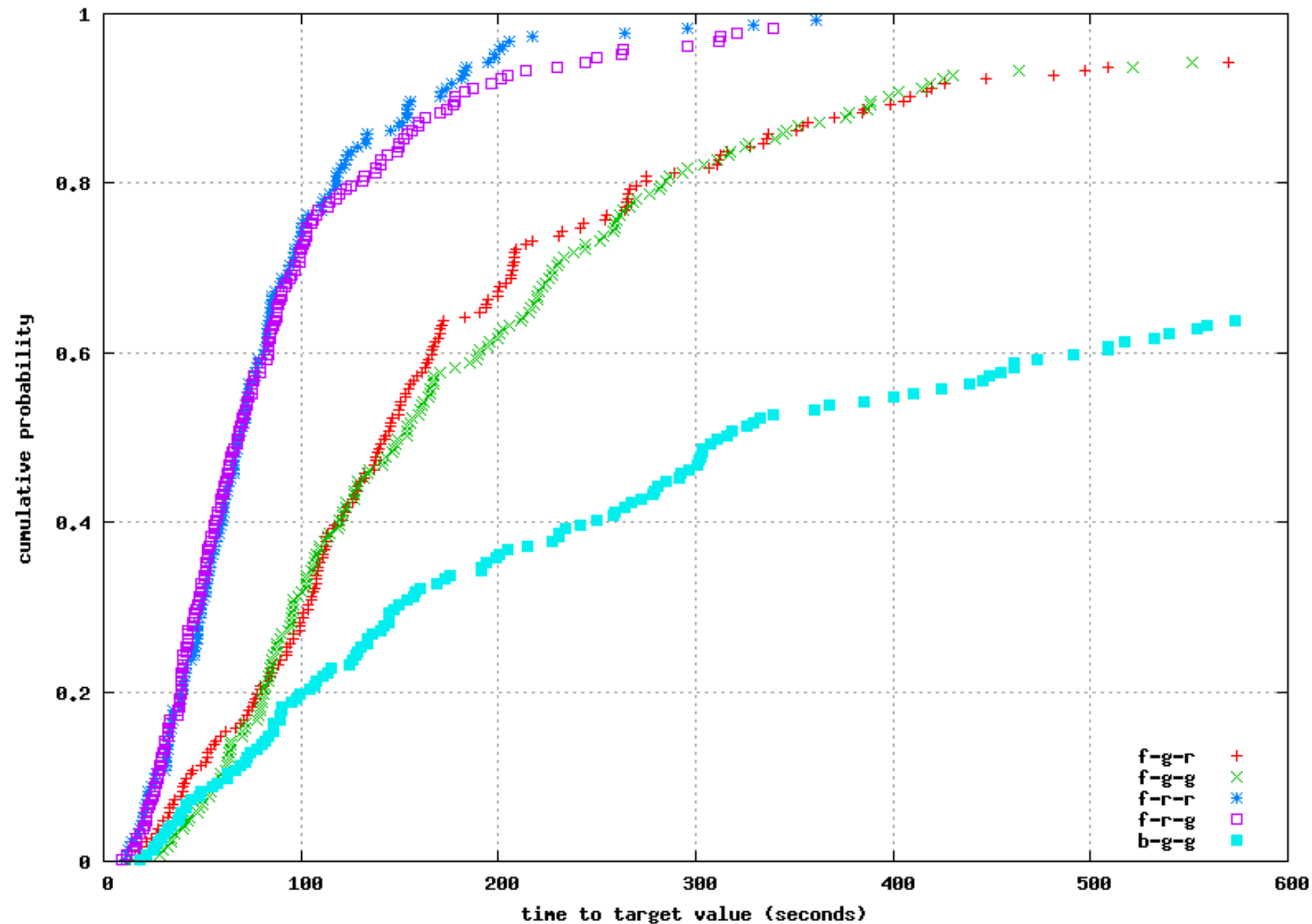
Path-relinking direction: forward (f) or backward (b);

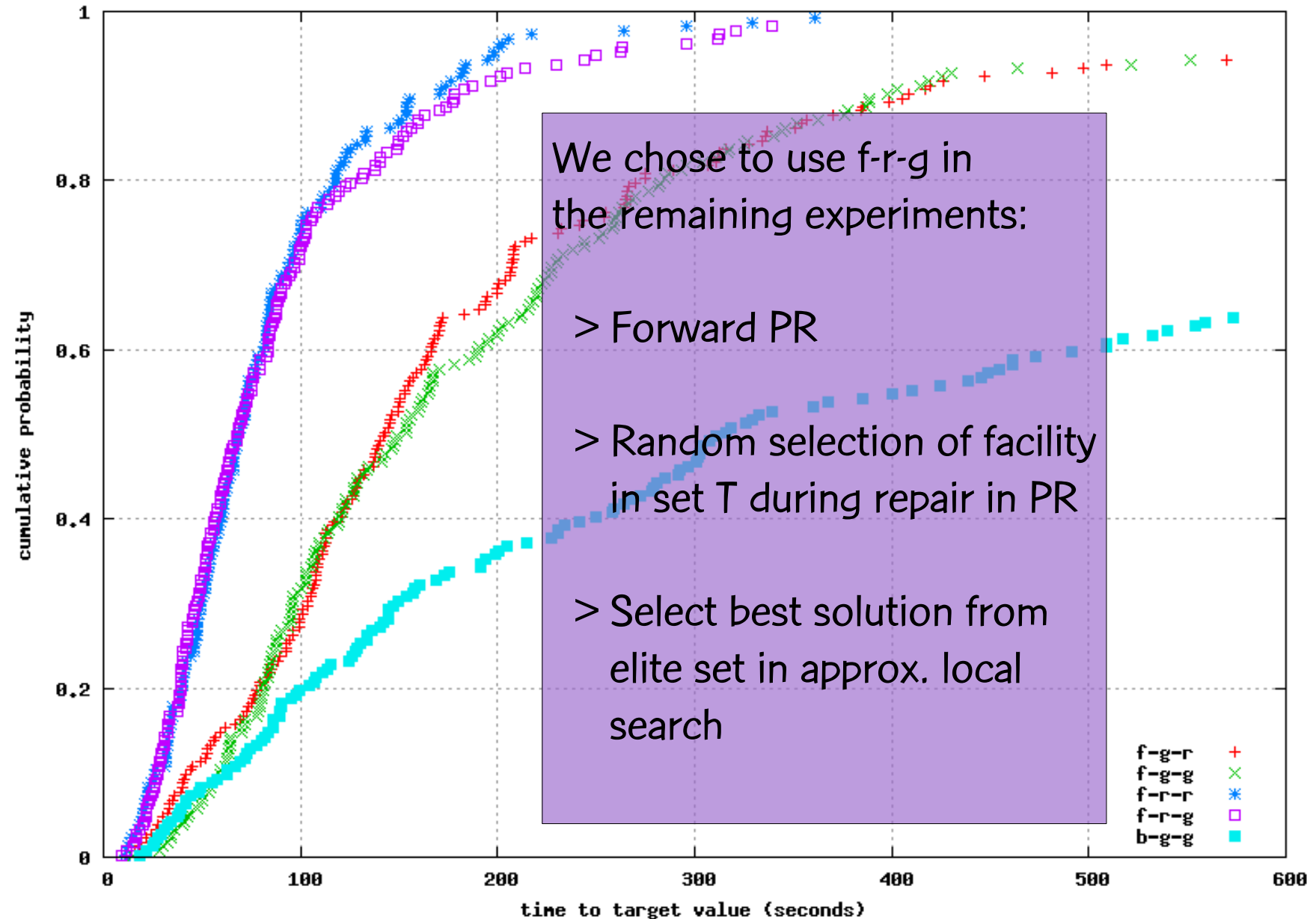
Criteria to select a facility from set T in the repairing procedure: randomly (r) or greedily (g)

Criteria to select a solution from elite set in the approximate local search: randomly (r) or greedily (g).

Combinations:  $2^3 = 8$

Parameter tuning: Instance 50-10-95





# Comparison with other algorithms

Elloumi et al. (2003)

Lee and Ma (2005)

Cordieu et al. (2006)

Hahn et al. (2007)

Pessoa et al. (2008)

# Comparison with Elloumi et al (2003):

Method(s): Three linearization methods (L1, L2, and L3), three semidefinite programming formulations (S0, S1, and S2) and a Lagrangian decomposition (D0).

Instances (Elloumi (1991) and Roupin (2004)): For each one of eight types [four configurations (A, B, C, and D) with two classes of instances], five instances with 10 facilities and three locations, and five instances with 20 facilities and five locations. Total of 80 instances

Comparison: GRASP-PR achieved the target values on all instances, with an AVERAGE performance improvement varying between a factor of 7.3 and over 5000 in relation to the BEST average time of the methods of Elloumi et al (2003)

# Comparison with Lee and Ma (2005):

Method(s): Three linearization methods (F-Y, K-B, and L3), based on the work of Frieze and Yadegar (1983), Kaufman and Broeckx (1978), and Padberg and Rijal (1996) and a branch and bound method (B&B) based on the work of Burkard (1991).

Instances: Suite of test problems with 10 to 16 facilities and 3 to 8 locations. Total of 25 instances.

Comparison: GRASP-PR found the target value on all 200 runs for each of the instances, with an AVERAGE performance improvement varying between a factor of 11.2 and 1004.6 in relation to the BEST average time of the methods of Lee and Ma (2005)

# Comparison with Cordeau et al. (2006):

Method: memetic algorithm.

Instances: problems with 20 to 50 facilities and 6 to 20 locations.

Total = 21 instances

Comparison: GRASP-PR found the target value on all 200 runs for each of the instances, with an AVERAGE performance improvement varying between a factor of 1.5 and 59.2 in relation to the BEST average time of the memetic algorithm, except for instances 30-20-95, 35-15-95, and 50-10-75.

However, for the last two instances the FASTEST GRASP-PR running times were FAR LESS than those of the memetic algorithm.

For instance 30-20-95, the GRASP-PR heuristic found the best solution found by the memetic algorithm but in 44 hours and 47 minutes.



# Comparison with Hahn et al. (2007):

Method(s): Level-1 reformulation-linearization technique (RLT) dual ascent procedure in a branch-and-bound scheme.

Instances: Four instances from Elloumi et al. (2003), three instances from Lee and Ma (2005), and one instance from Cordeau et al. (2006). Total of eight instances.

Comparison: GRASP-PR found the target value on all 200 runs for each of the instances, with an AVERAGE performance improvement varying between a factor of 8.8 and over 69,000 w.r.t. the BEST average time of the method of Hahn et al. (2007).

# Comparison with Pessoa et al. (2008):

**Method:** Combination of Hahn et al. (2007) dual ascent procedure with the general-purpose volume algorithm of Barahona and Anbil (2000).

**Instances:** Four instances from Elloumi et al. (2003), three instances from Lee and Ma (2005), and 12 instances from Cordeau et al. (2006). Total of 24 instances.

**Comparison:** GRASP-PR found the target value on all 200 runs for each of the instances, with an AVERAGE performance improvement varying between a factor of 132.7 and over 100,000 w.r.t. the BEST average time of the method of Pessoa et al. (2008), except for instance 30-20-95.

# Concluding remarks

# Concluding remarks

Reviewed generalized quadratic assignment problem

Described several heuristics:

- > Greedy
- > Randomized greedy
- > Local search
- > Path-relinking
- > GRASP
- > GRASP with path-relinking

## Coauthors



Ricardo M.A. Silva  
Fed. U. of Lavras,  
Brazil. Visiting scholar  
at AT&T Research (2008-2010)



Geraldo R. Mateus  
Federal University of  
Minas Gerais, Brazil

# The End

Slides and full paper can be downloaded from  
<http://mauricioresende.com>