GRASP with evolutionary path-relinking

VII Metaheuristics International Conference (MIC 2007)
Montréal, Canada ~ June 25-29, 2007

Mauricio G. C. Resende
AT&T Labs Research
Florham Park, New Jersey
mgcr@att.com

Joint work with Diogo Andrade (Rutgers U.)

Summary

- Path-relinking
- GRASP with path-relinking
- Evolutionary path-relinking
- GRASP with evolutionary path-relinking
- Experimental results
- Concluding remarks

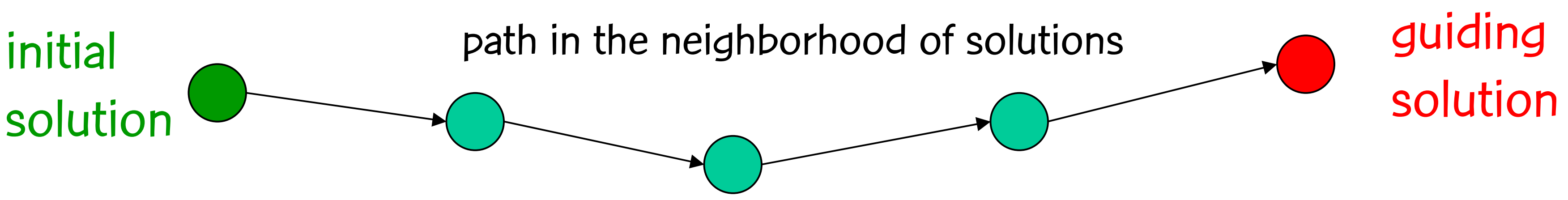
Path-relinking (PR)

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.

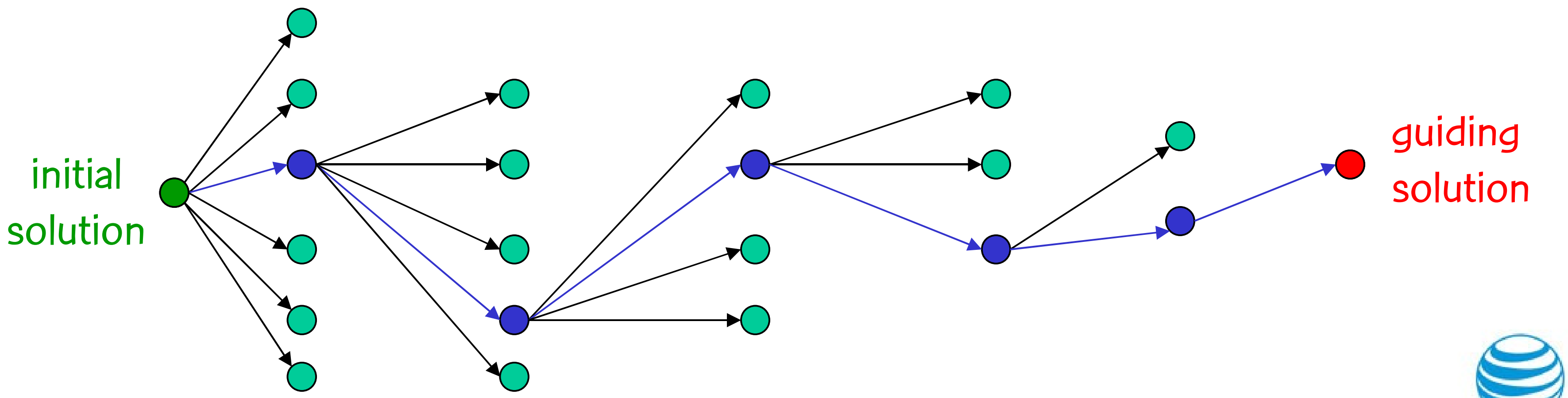
Path-relinking

- 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:



Path-relinking

Solutions x and y to be combined.

$\Delta(x,y)$: symmetric difference between x and y

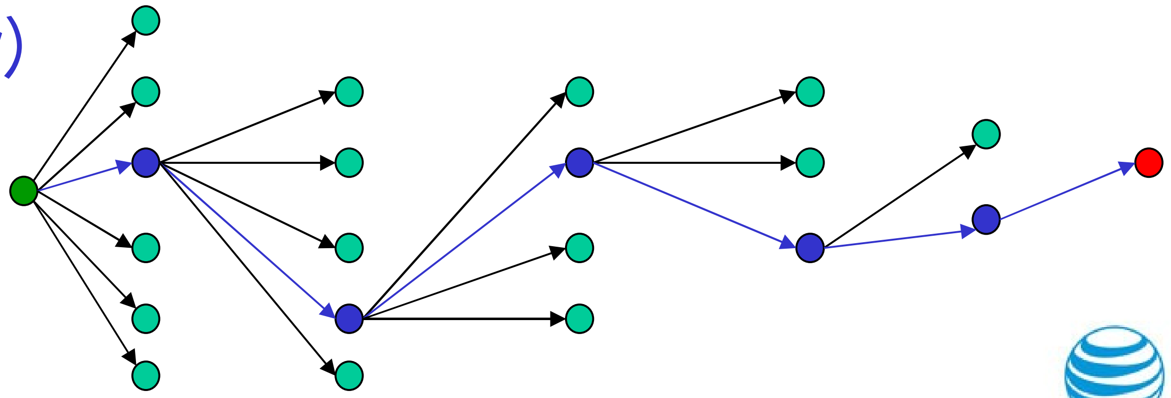
while ($|\Delta(x,y)| > 0$) {

1: evaluate corresponding moves in $\Delta(x,y)$

2: make best move

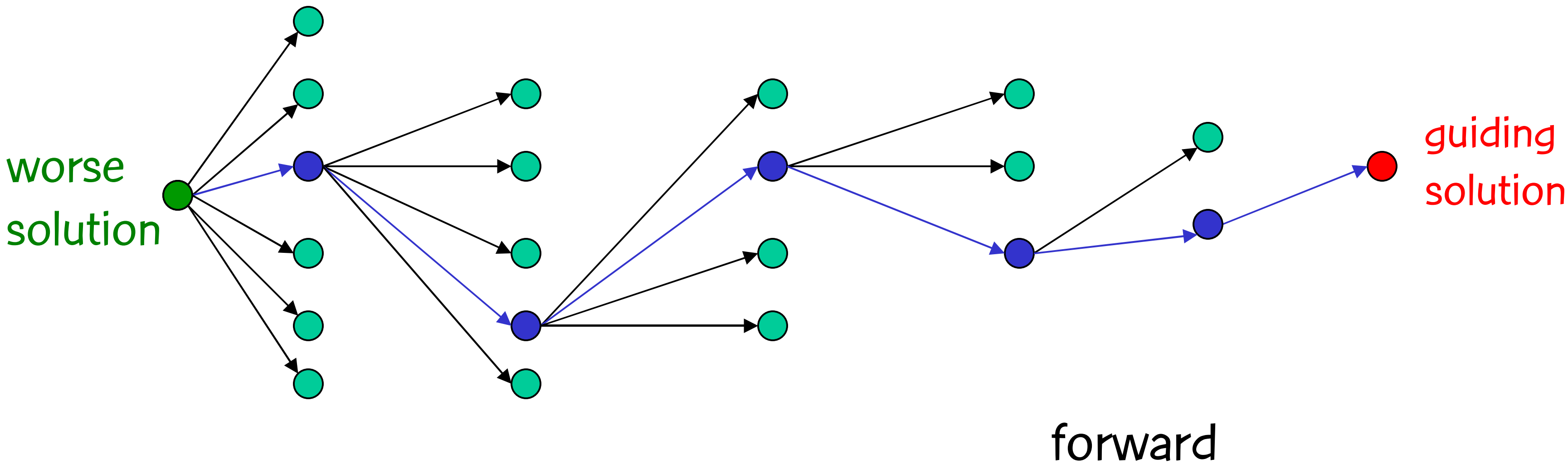
3: update $\Delta(x,y)$

}



Forward path-relinking

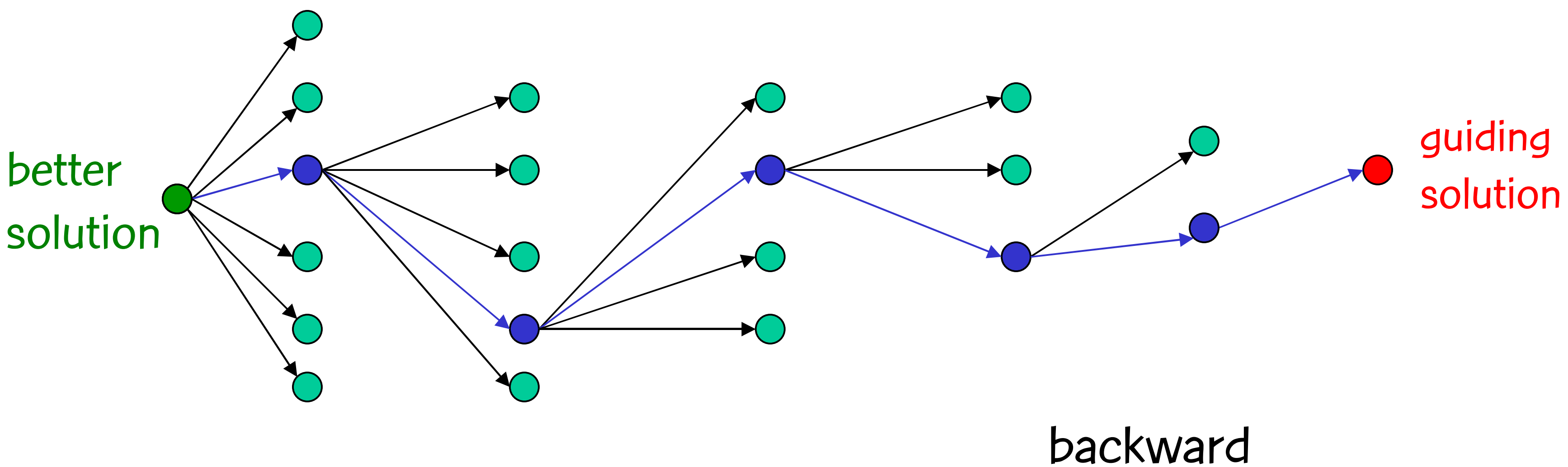
- Variants: trade-offs between computation time and solution quality
  - Forward PR adopts as initial solution the worse of the two input solutions and uses the better solution as the guide.



at&t
Your world. Delivered.

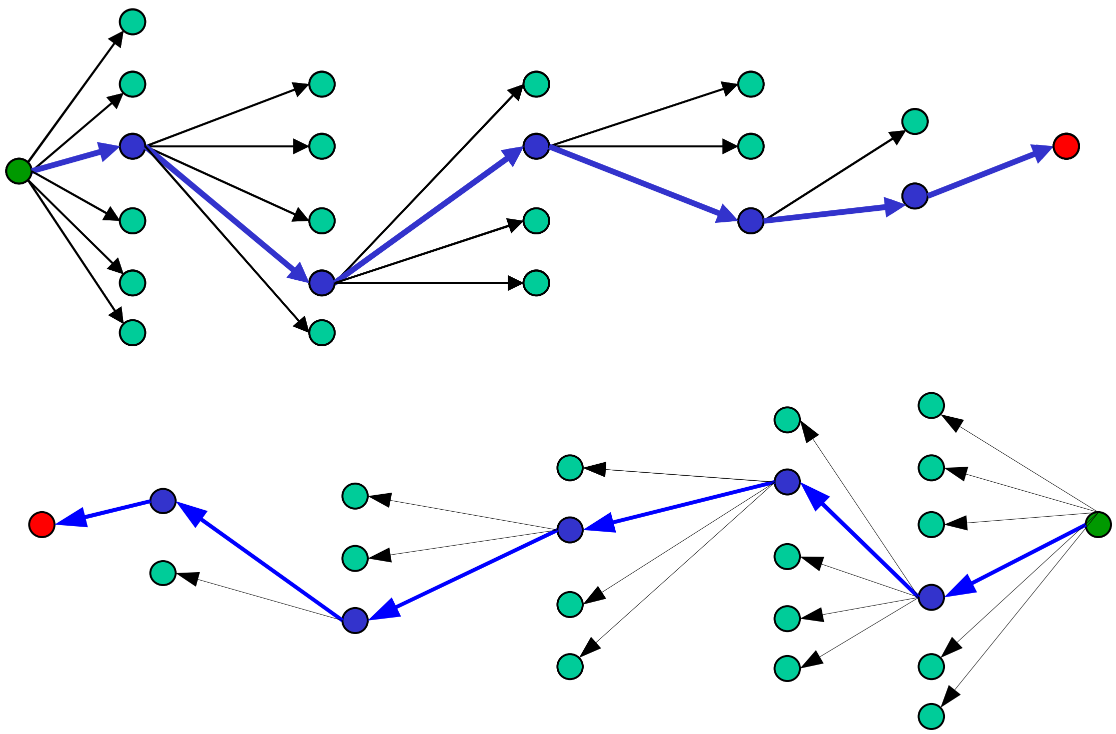
Backward path-relinking

- Variants: trade-offs between computation time and solution quality
  - Backward PR usually does better: Better start from the better of the two input solutions, neighborhood of the initial solution is explored more than of the guide!



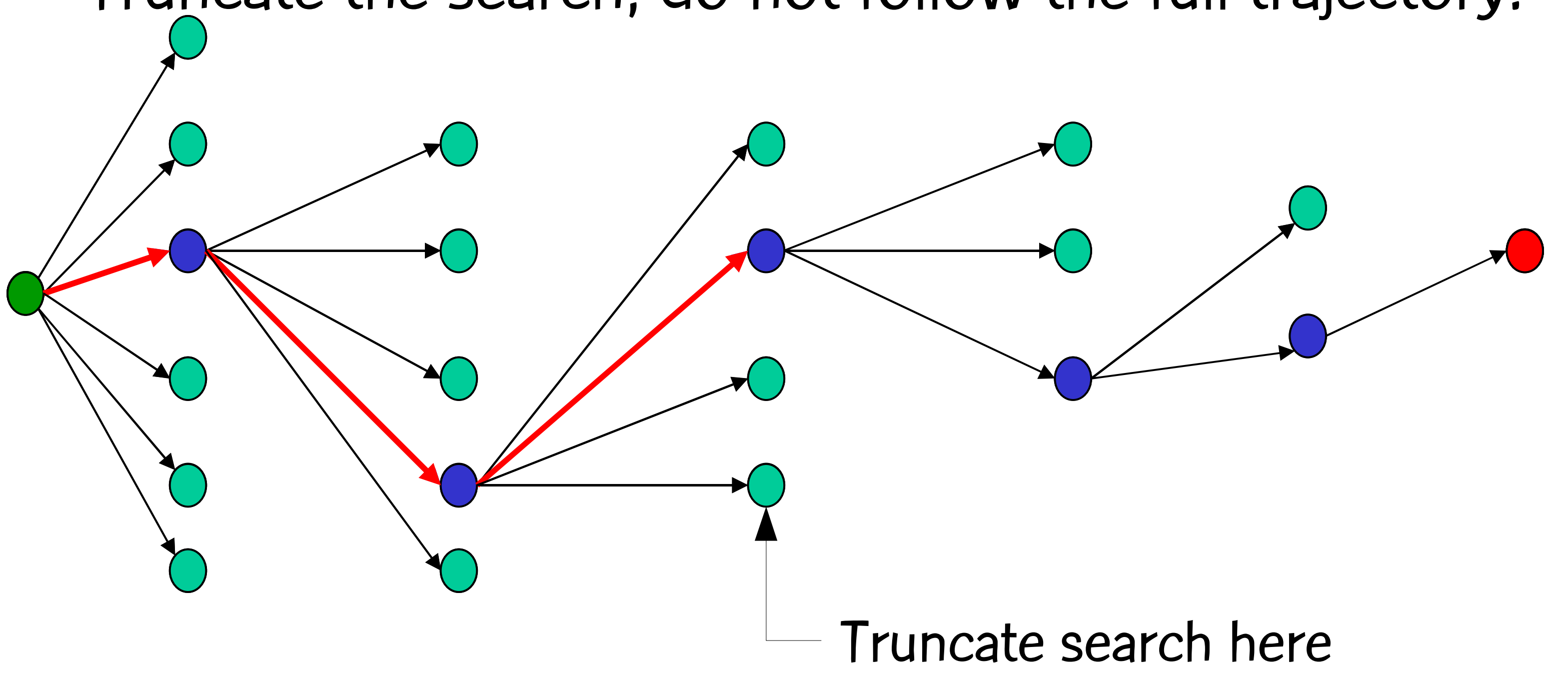
Back and forth path-relinking

- Variants: trade-offs between computation time and solution quality
  - Explore both trajectories: twice as much time, often with only marginal improvements!



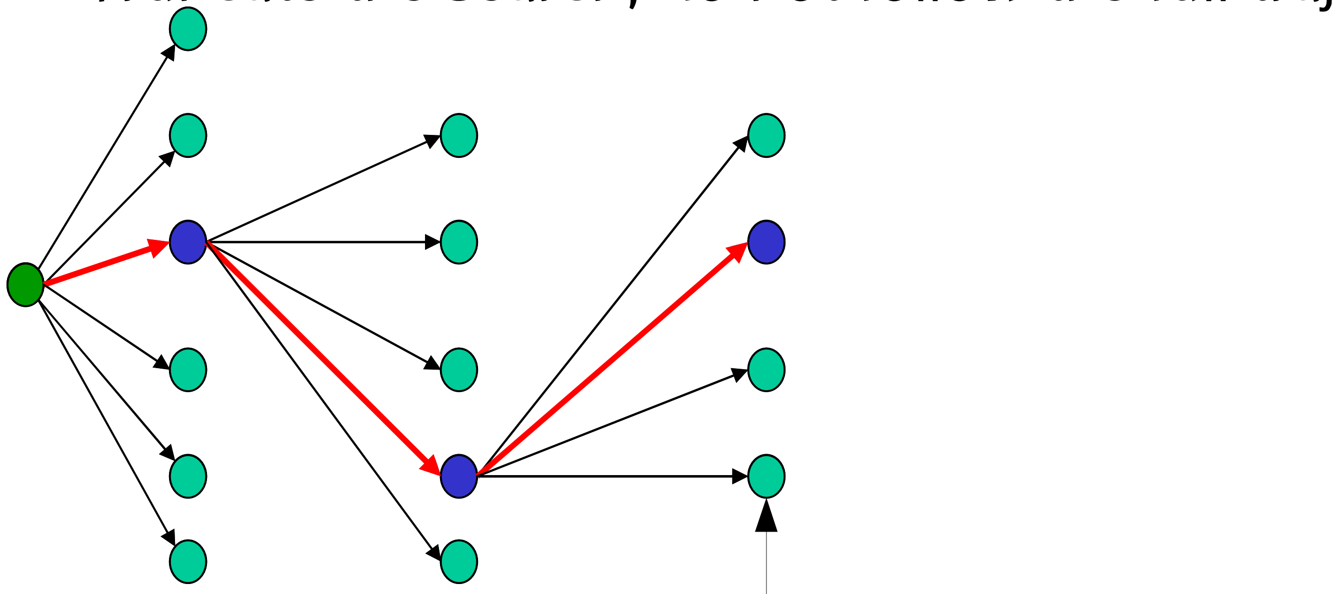
Truncated path-relinking

- Variants: trade-offs between computation time and solution quality
  - Truncate the search, do not follow the full trajectory.



Truncated path-relinking

- Variants: trade-offs between computation time and solution quality
  - Truncate the search, do not follow the full trajectory.

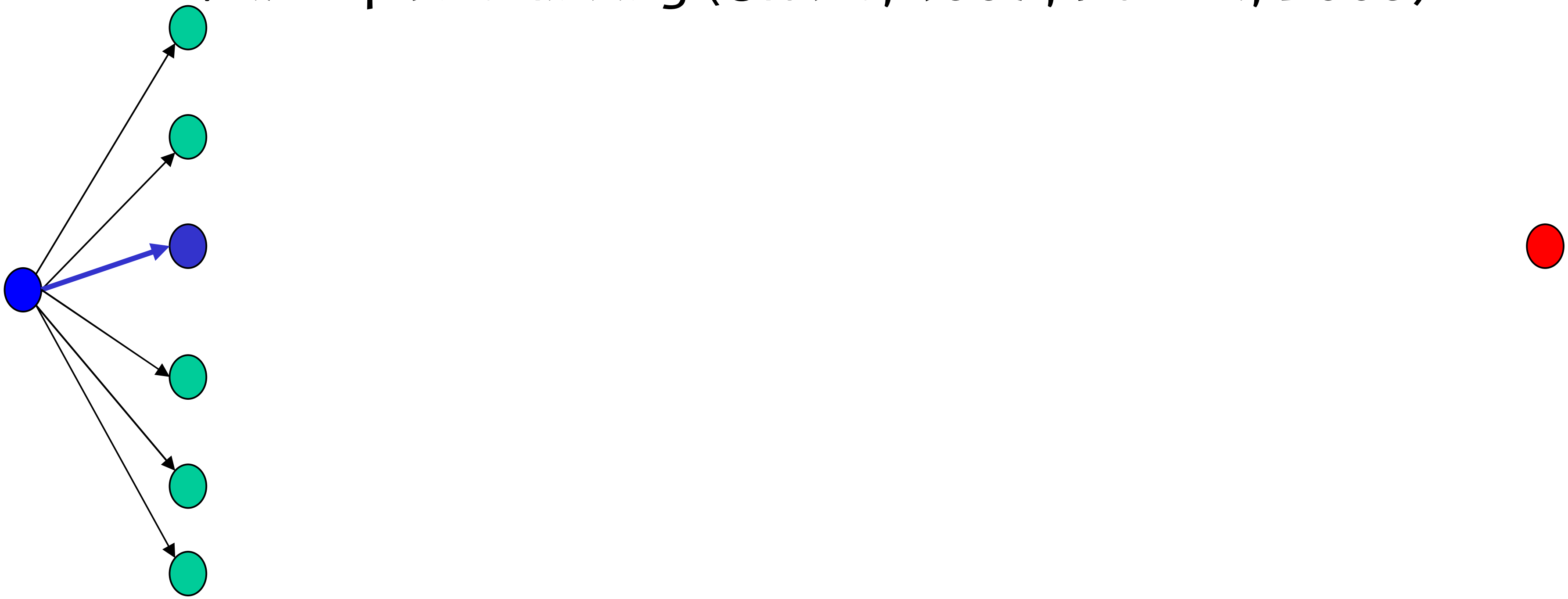


Mixed path-relinking

- Variants: trade-offs between computation time and solution quality
  - Mixed path-relinking (Glover, 1997; Rosseti, 2003)

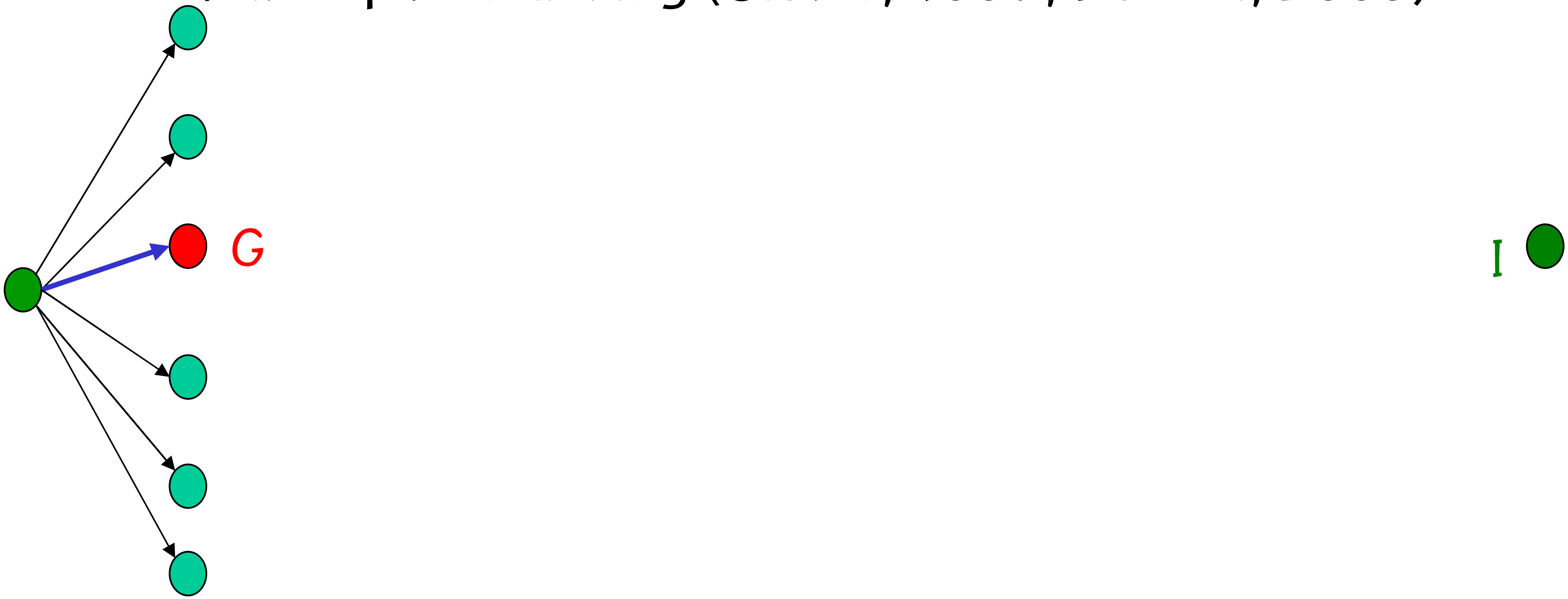
Mixed path-relinking

- Variants: trade-offs between computation time and solution quality
  - Mixed path-relinking (Glover, 1997; Rosseti, 2003)



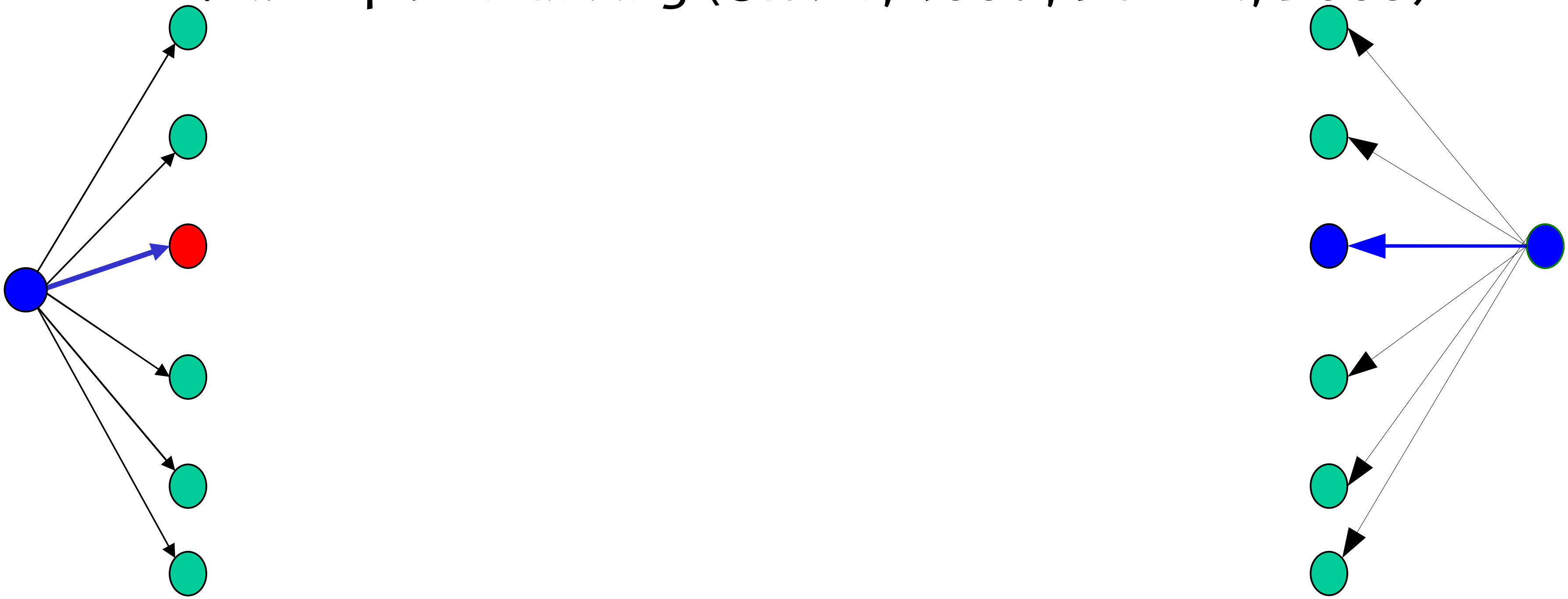
Mixed path-relinking

- Variants: trade-offs between computation time and solution quality
  - Mixed path-relinking (Glover, 1997; Rosseti, 2003)



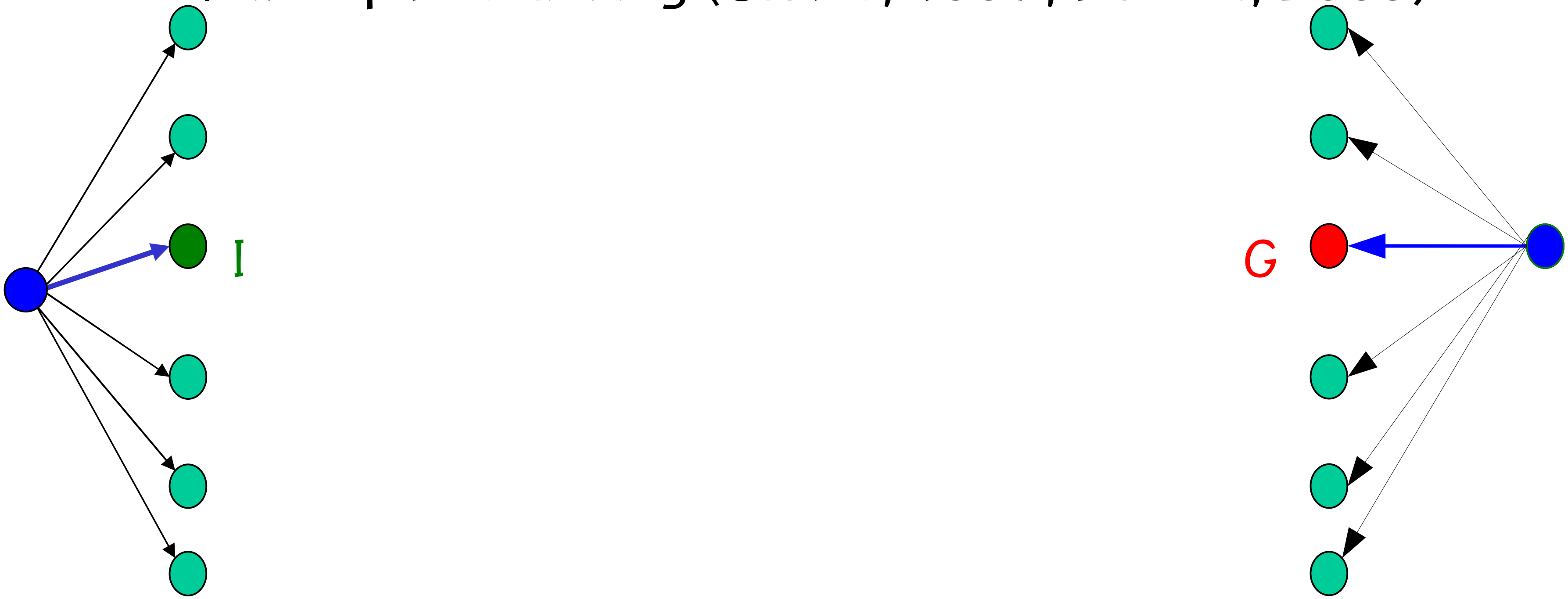
Mixed path-relinking

- Variants: trade-offs between computation time and solution quality
  - Mixed path-relinking (Glover, 1997; Rosseti, 2003)



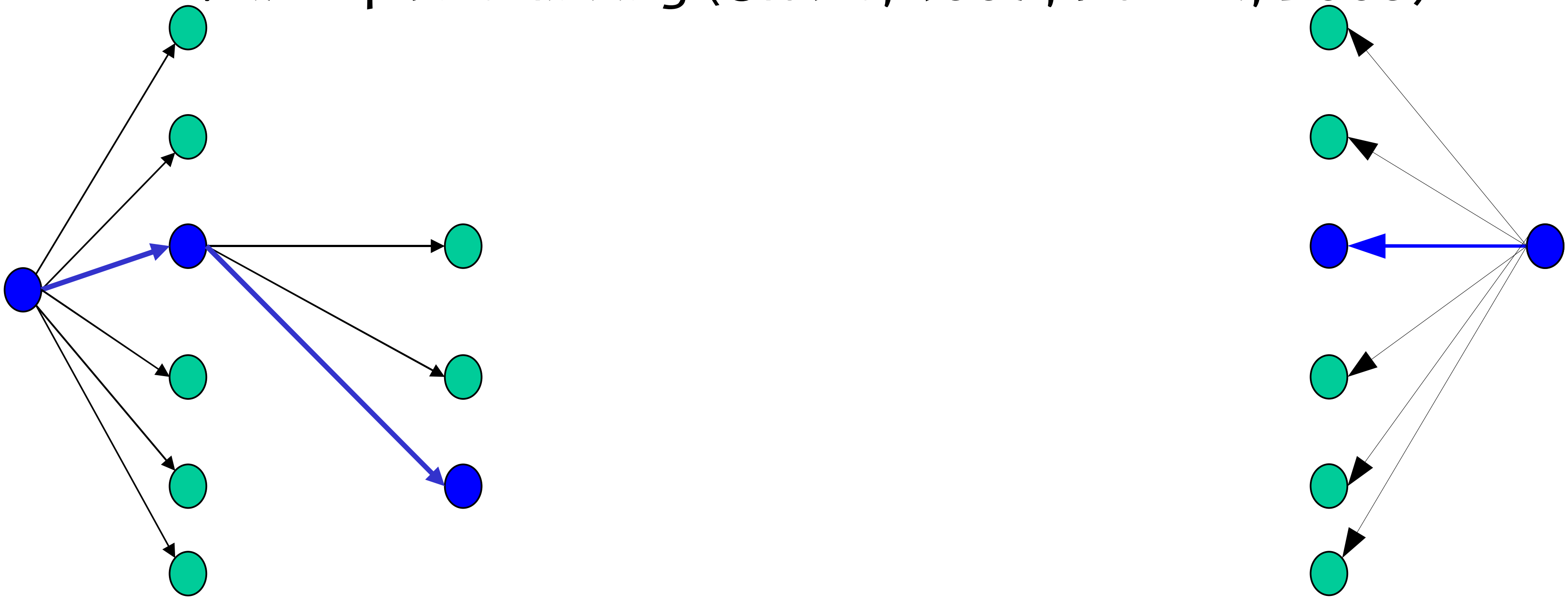
Mixed path-relinking

- Variants: trade-offs between computation time and solution quality
  - Mixed path-relinking (Glover, 1997; Rosseti, 2003)



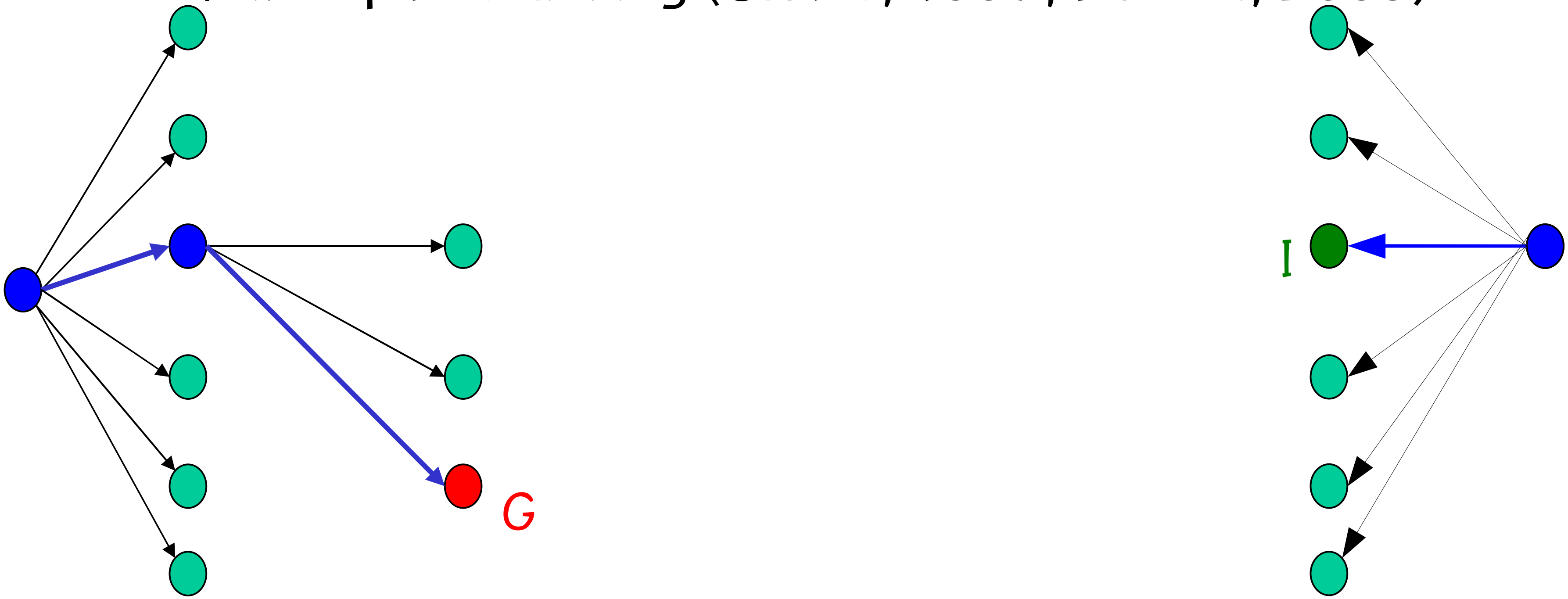
Mixed path-relinking

- Variants: trade-offs between computation time and solution quality
  - Mixed path-relinking (Glover, 1997; Rosseti, 2003)



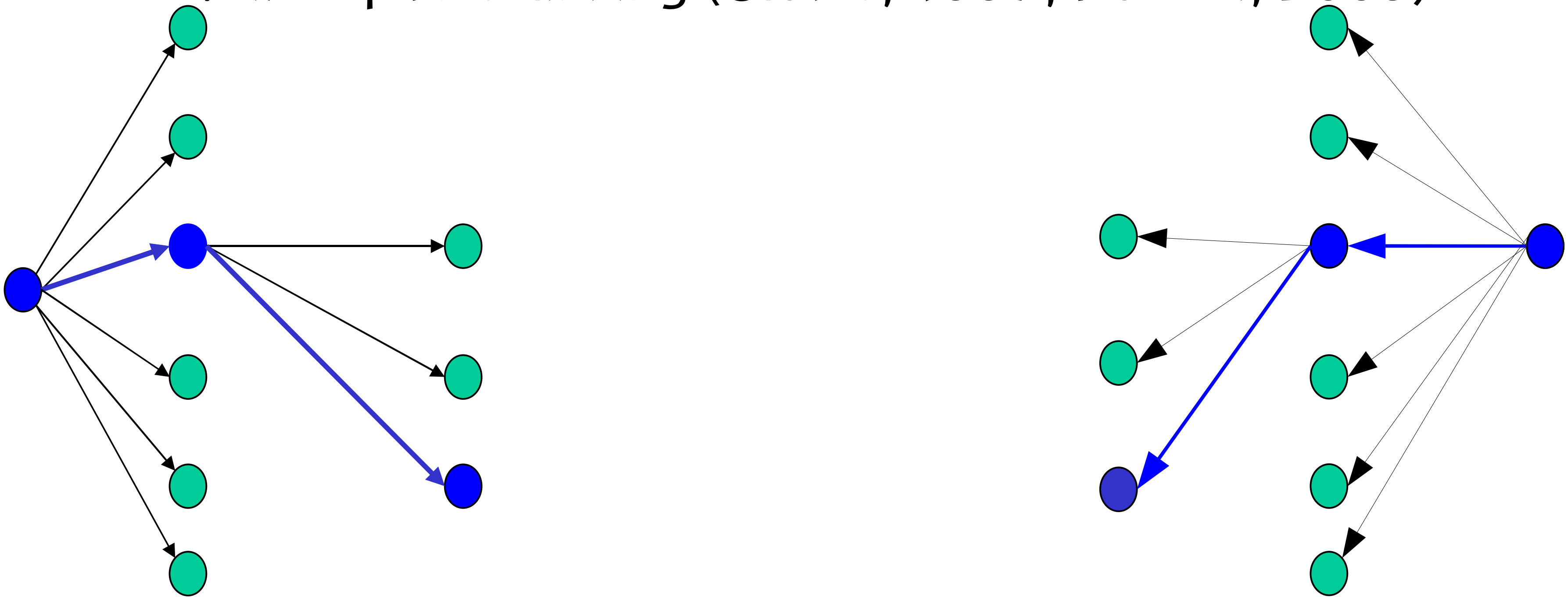
Mixed path-relinking

- Variants: trade-offs between computation time and solution quality
  – Mixed path-relinking (Glover, 1997; Rosseti, 2003)



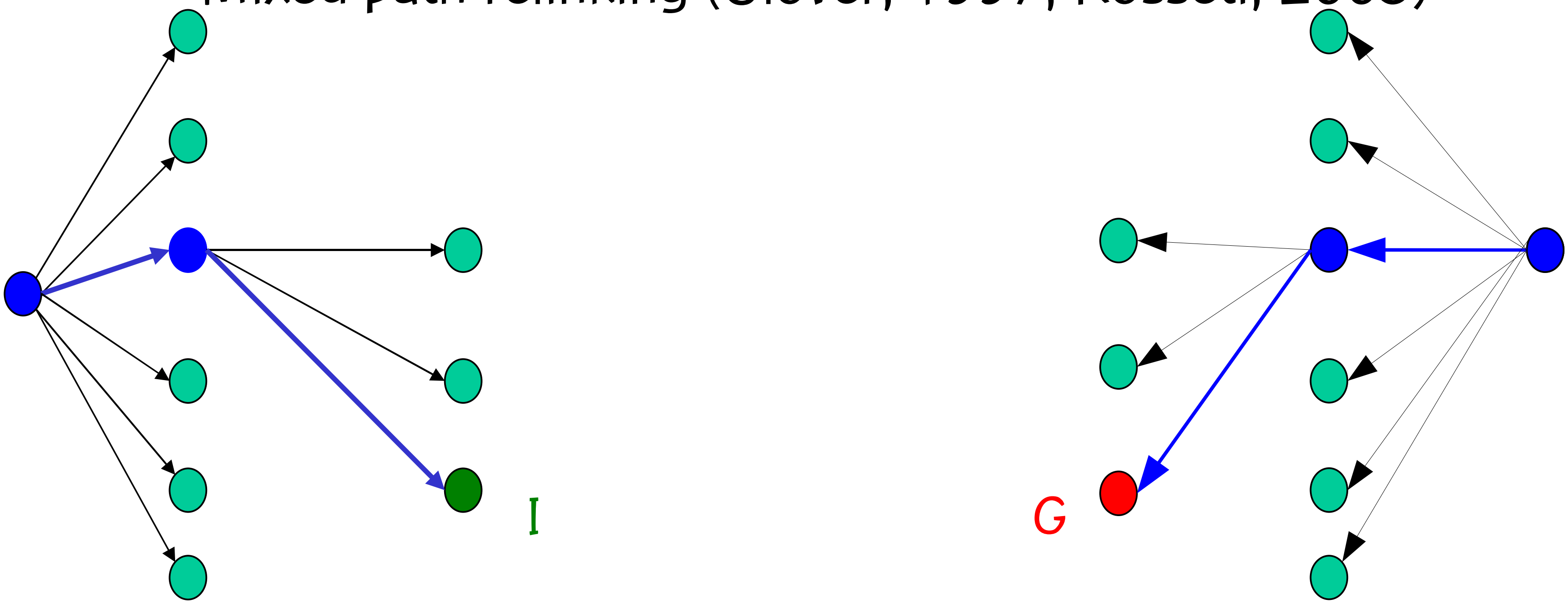
Mixed path-relinking

- Variants: trade-offs between computation time and solution quality
  - Mixed path-relinking (Glover, 1997; Rosseti, 2003)



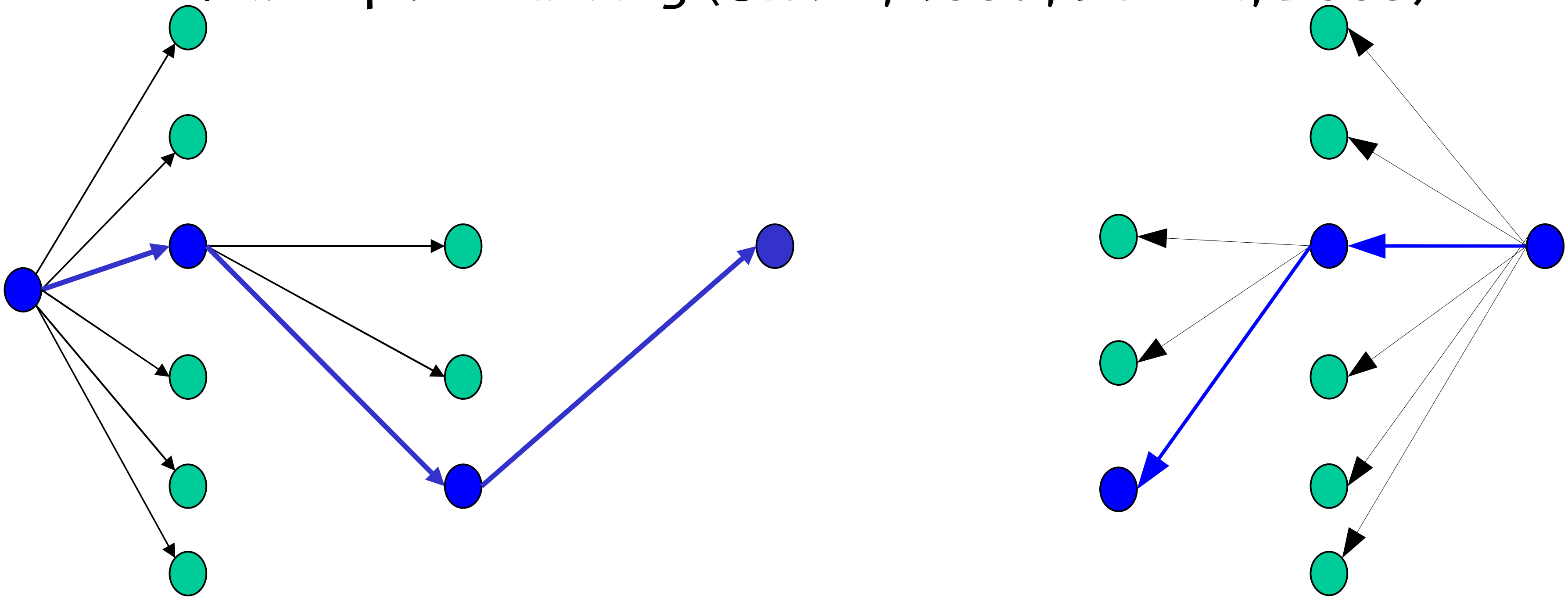
Mixed path-relinking

- Variants: trade-offs between computation time and solution quality
  - Mixed path-relinking (Glover, 1997; Rosseti, 2003)



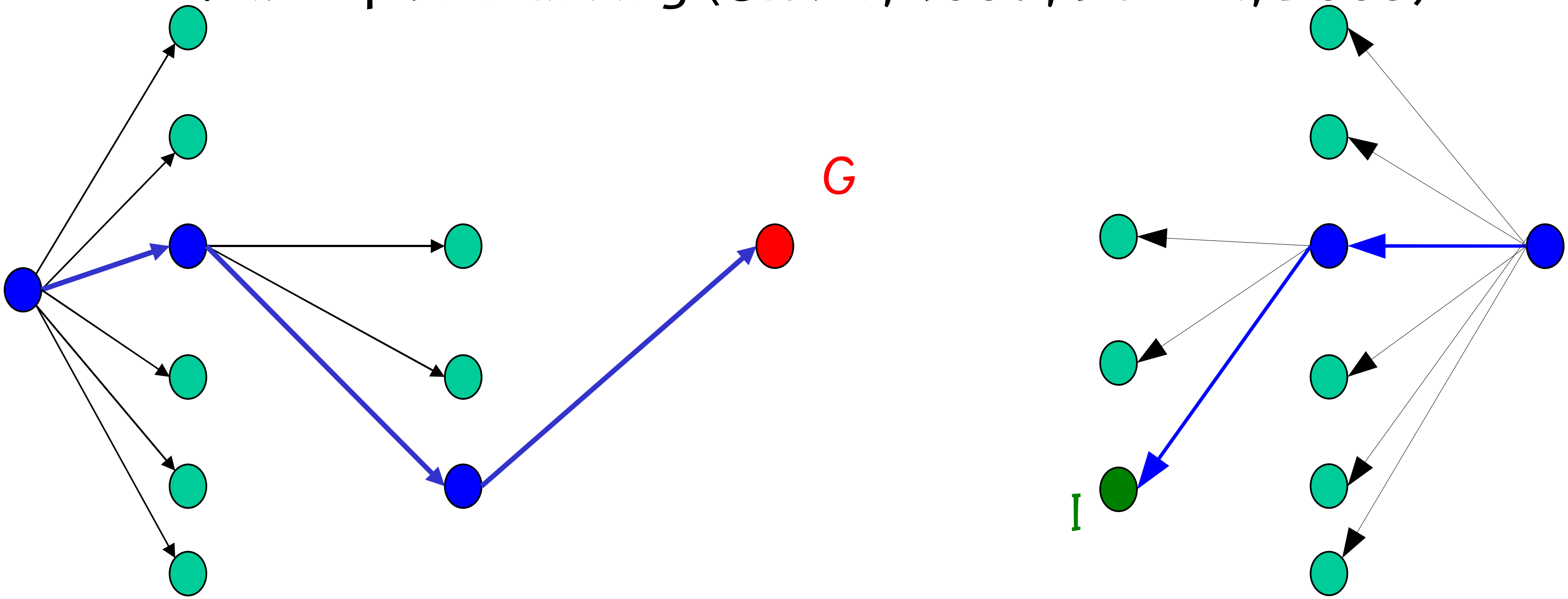
Mixed path-relinking

- Variants: trade-offs between computation time and solution quality
  - Mixed path-relinking (Glover, 1997; Rosseti, 2003)



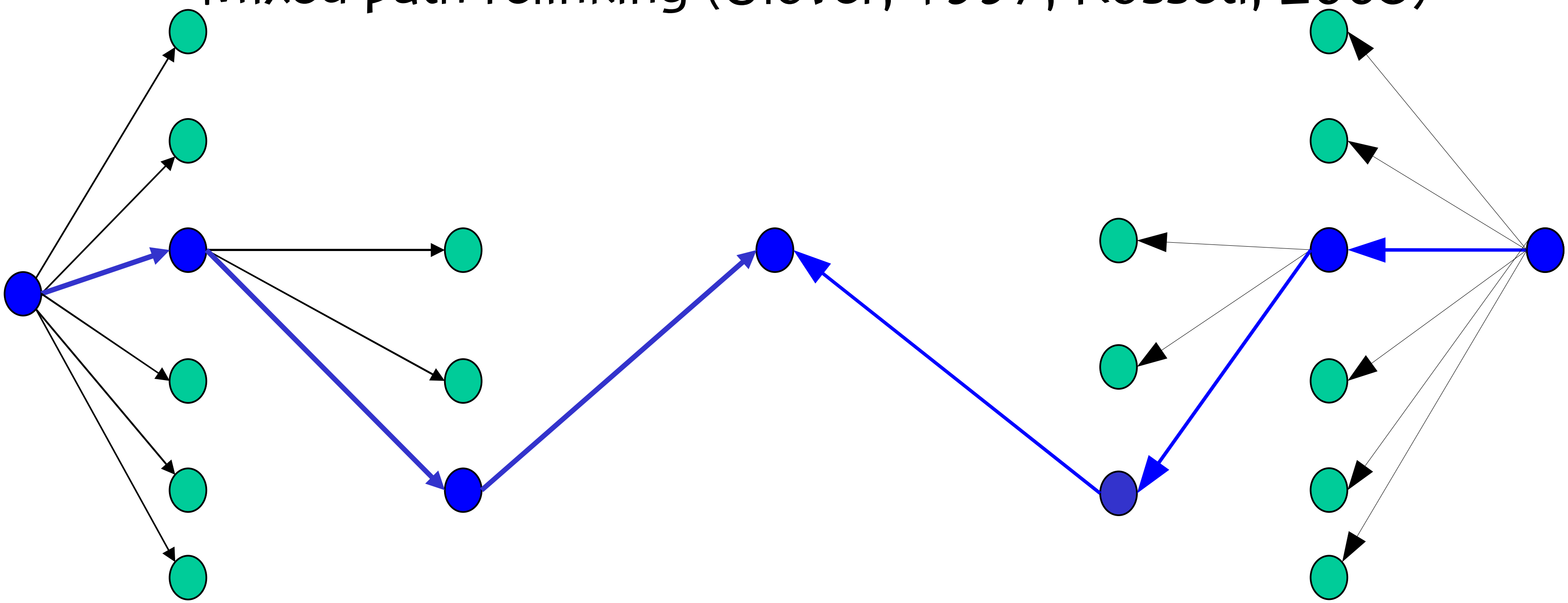
Mixed path-relinking

- Variants: trade-offs between computation time and solution quality
  - Mixed path-relinking (Glover, 1997; Rosseti, 2003)



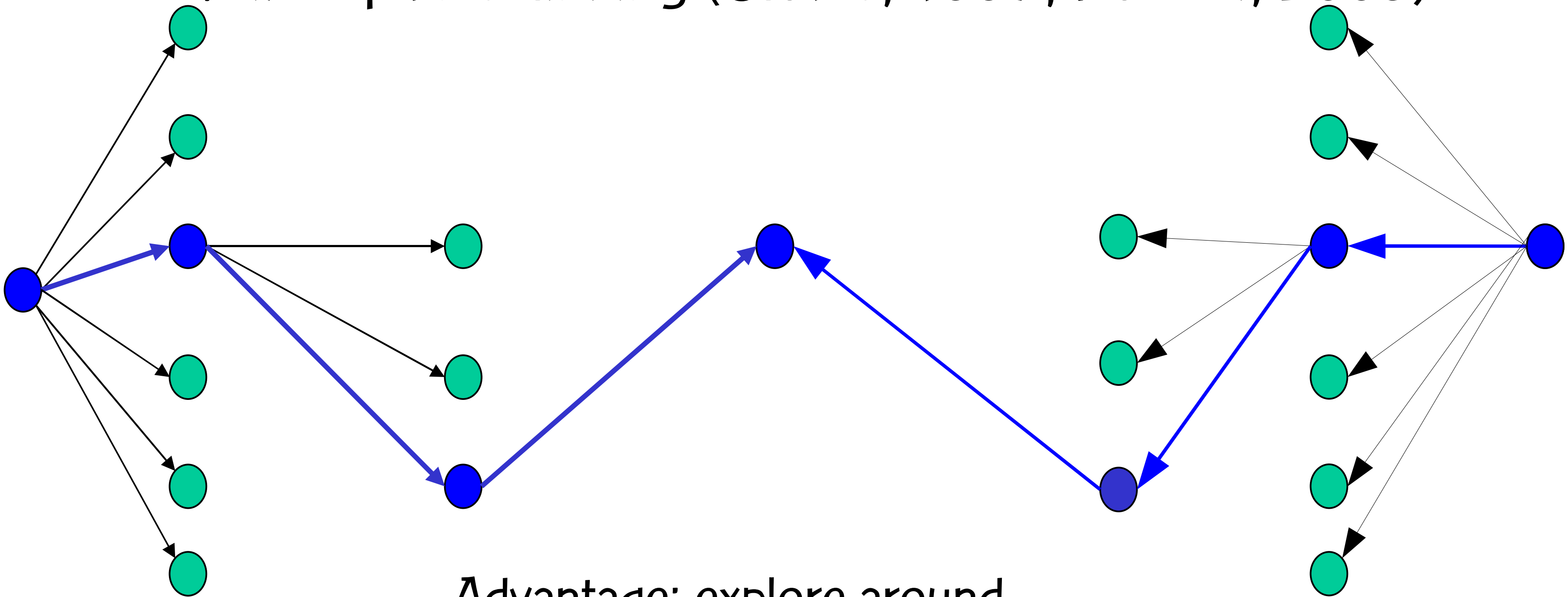
Mixed path-relinking

- Variants: trade-offs between computation time and solution quality
  - Mixed path-relinking (Glover, 1997; Rosseti, 2003)



Mixed path-relinking

- Variants: trade-offs between computation time and solution quality
  - Mixed path-relinking (Glover, 1997; Rosseti, 2003)



Advantage: explore around neighborhoods of both input solutions.

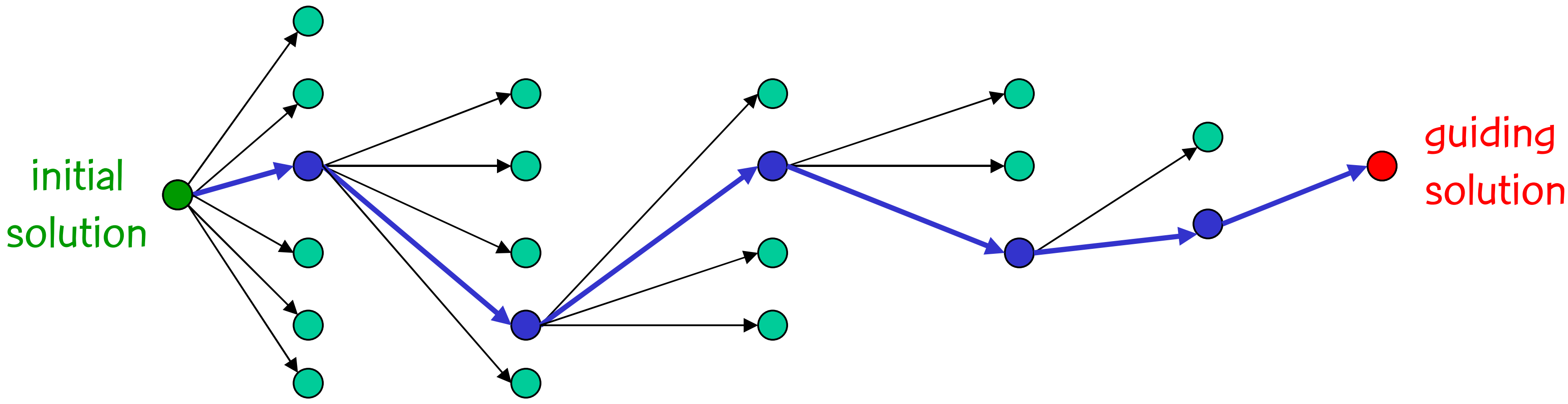
Truncated mixed path-relinking

- Variants: trade-offs between computation time and solution quality
  - Truncated mixed path-relinking

Greedy randomized adaptive path-relinking

(Faria, Binato, Resende, & Falcão, 2005)

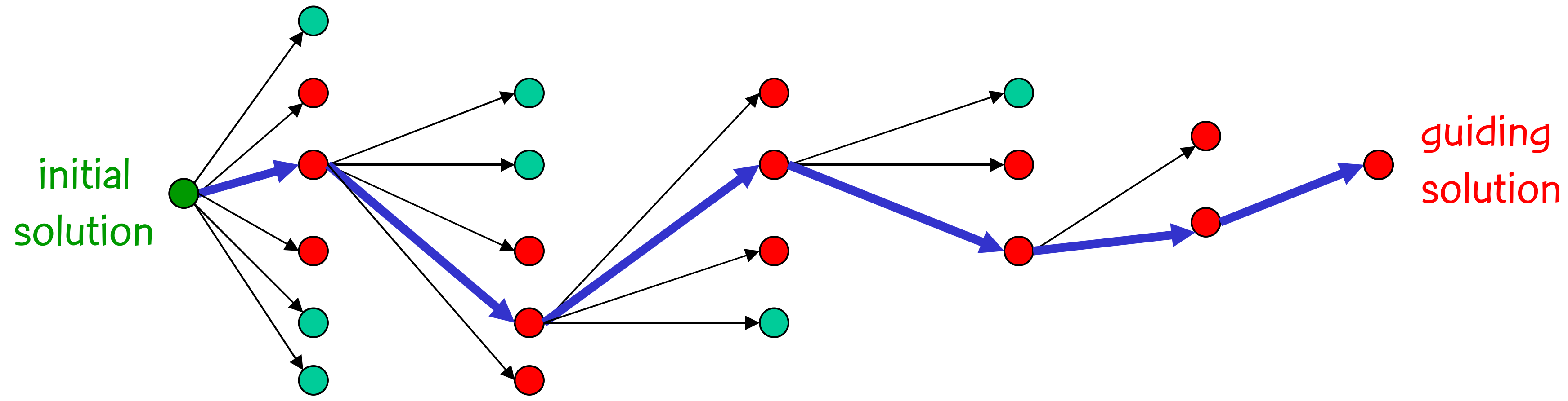
- Incorporates semi-greediness into PR.
- Standard PR selects moves greedily: samples one of exponentially many paths



Greedy randomized adaptive path-relinking

(Faria, Binato, Resende, & Falcão, 2005)

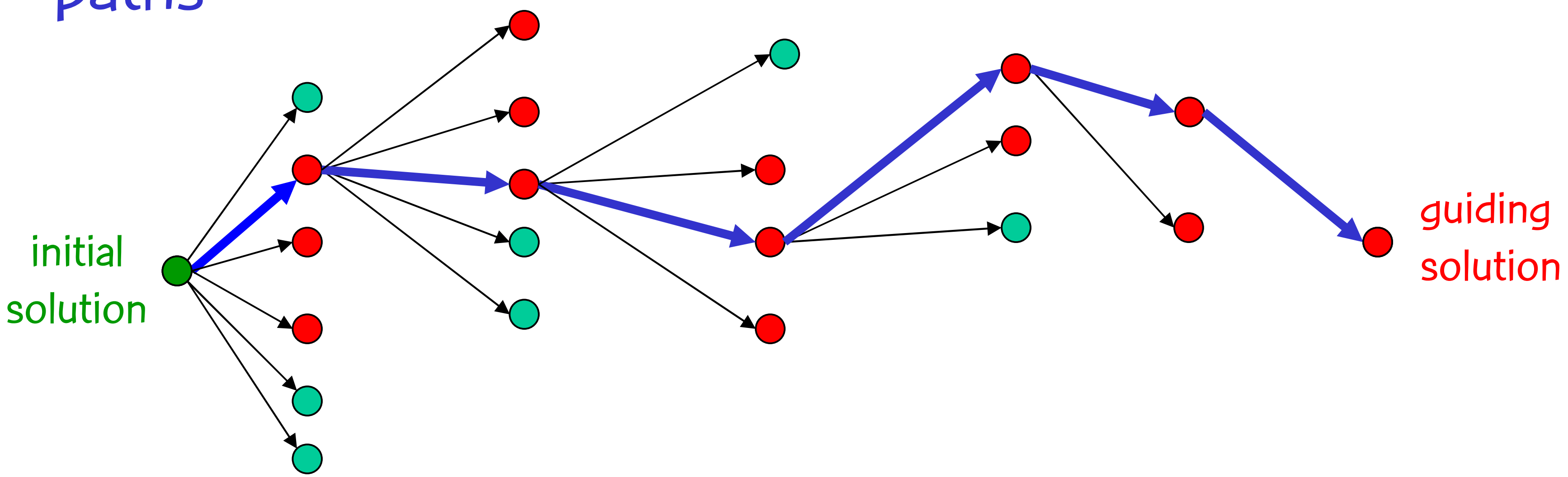
- Incorporates semi-greediness into PR.
- graPR creates RCL with best moves: samples several paths



Greedy randomized adaptive path-relinking

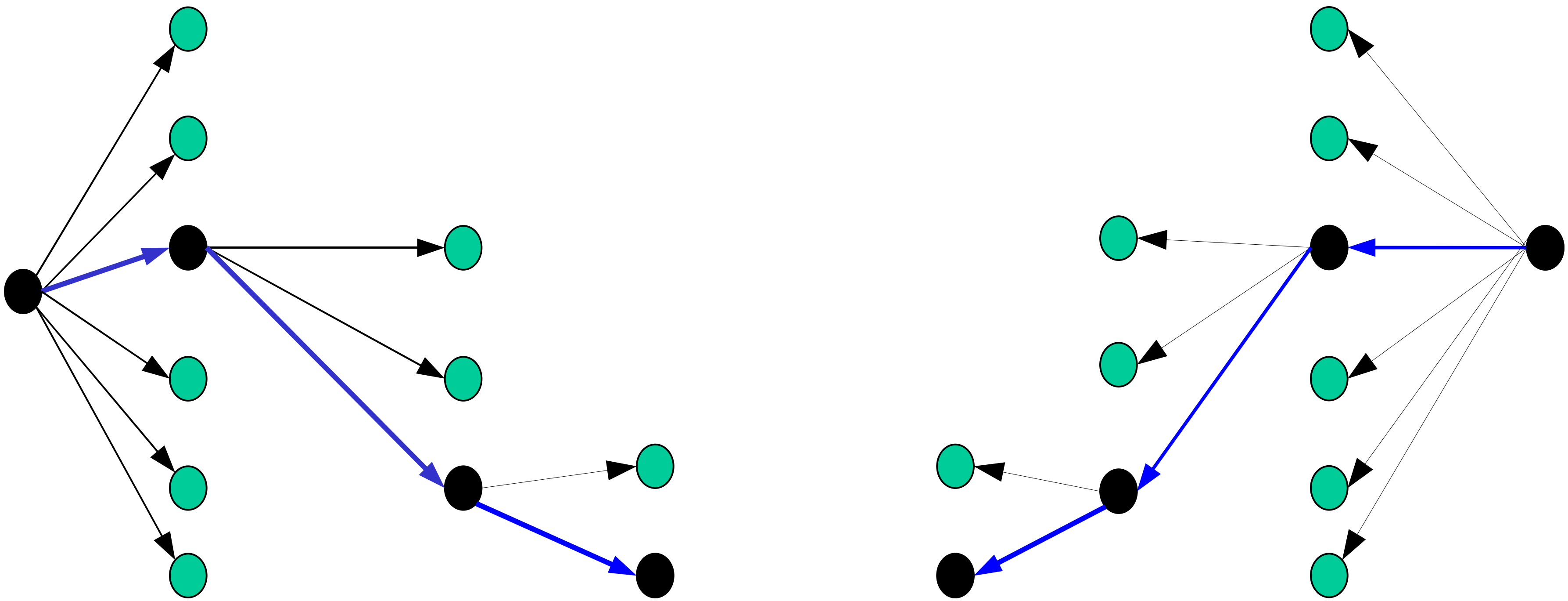
(Faria, Binato, Resende, & Falcão, 2005)

- Incorporates semi-greediness into PR.
- graPR creates RCL with best moves: samples several paths



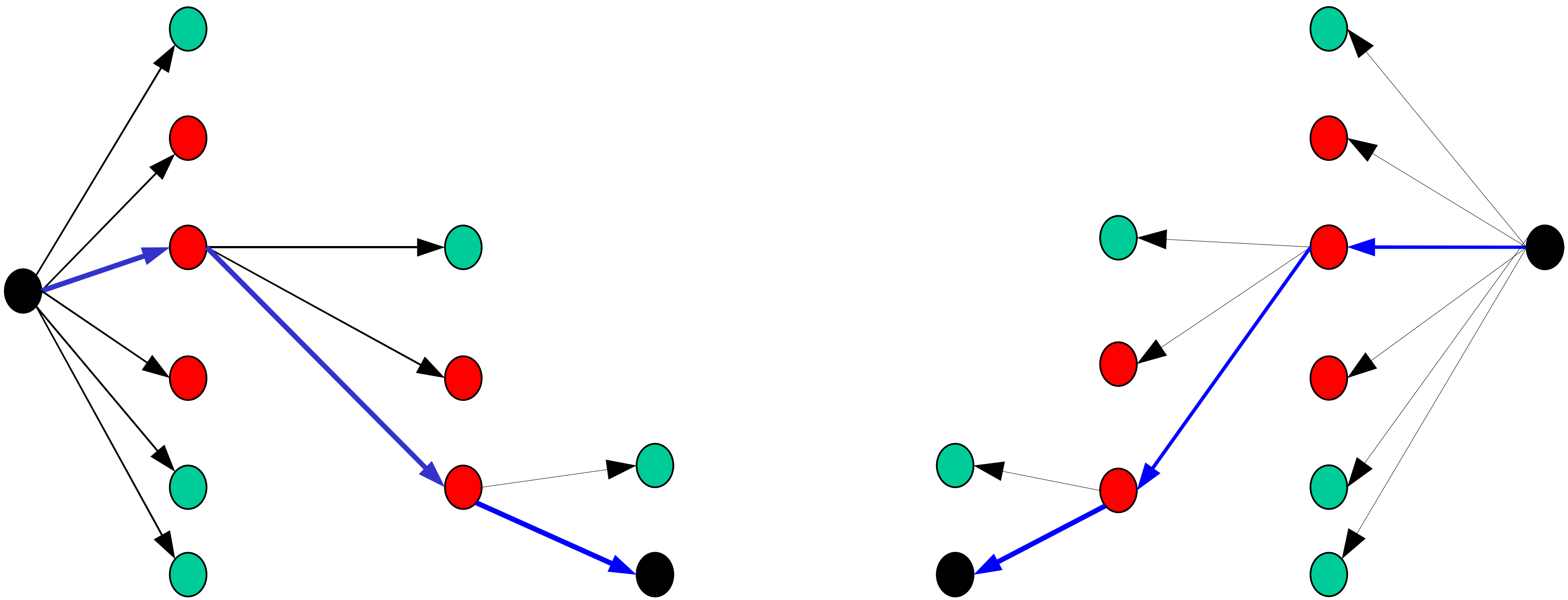
Truncated mixed graPR

When applied to a given pair of solutions truncated mixed PR explores one of exponentially many path segments each time it is executed.



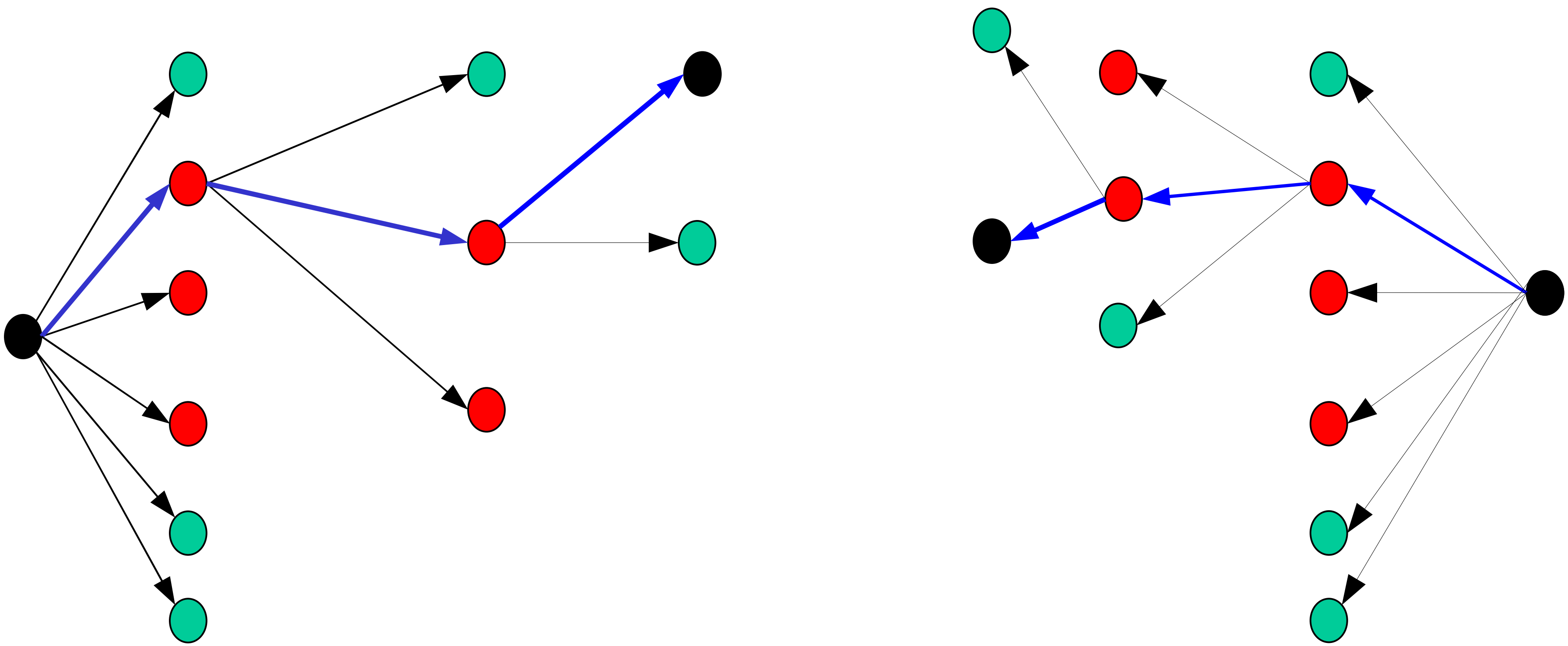
Truncated mixed graPR

With high probability, truncated mixed graPR explores different path segments each time it is executed between the same pair of solutions.



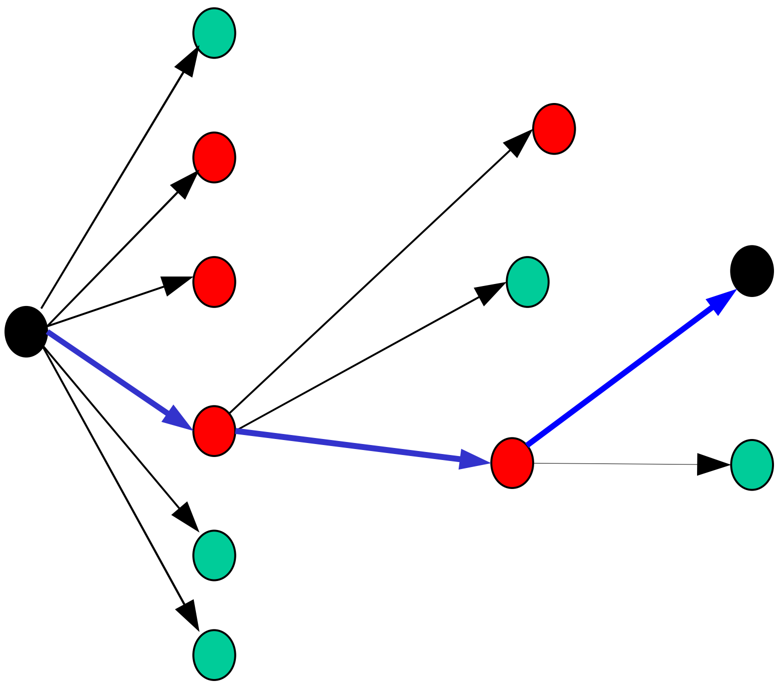
Truncated mixed graPR

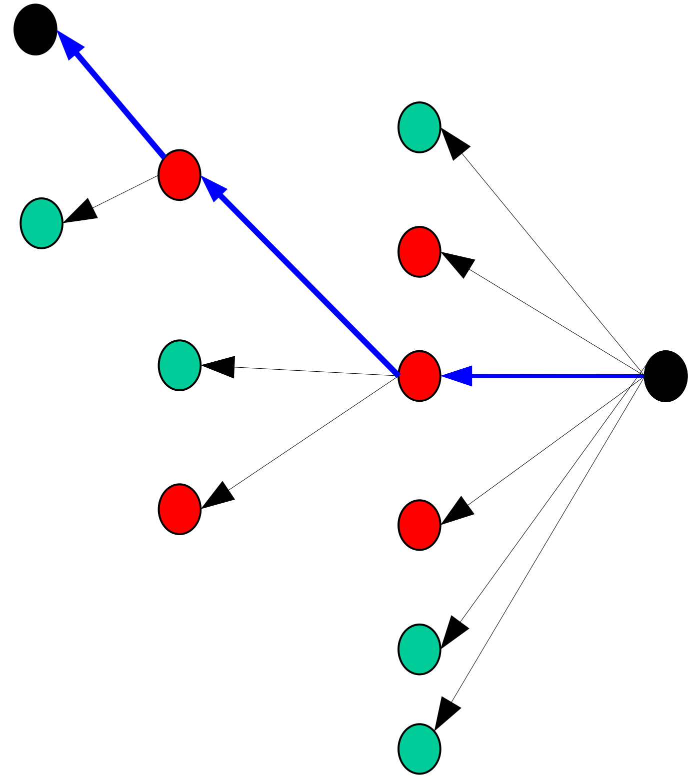
With high probability, truncated mixed graPR explores different path segments each time it is executed between the same pair of solutions.



Truncated mixed graPR

With high probability, truncated mixed graPR explores different path segments each time it is executed between the same pair of solutions.





GRASP with path-relinking

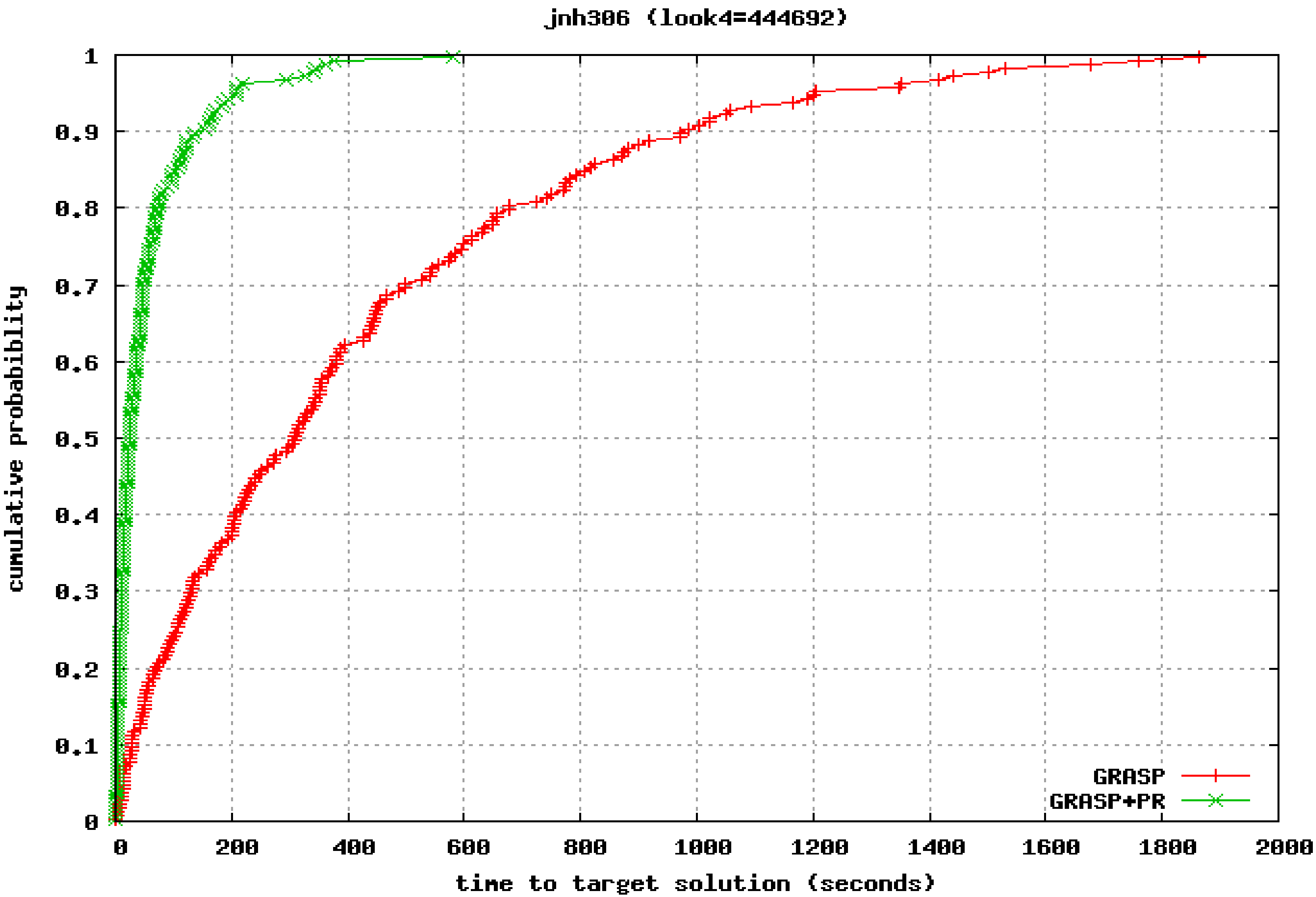
GRASP with path-relinking

- Originally used by Laguna and Martí (1999).
- Maintains a set of elite solutions found during GRASP iterations.
- After each GRASP iteration (construction and local search):
  - Use GRASP solution as initial solution.
  - Select an elite solution uniformly at random: guiding solution.
  - Perform path-relinking between these two solutions.

GRASP with path-relinking

- Since 1999, there has been a lot of activity in hybridizing GRASP with path-relinking.
- Survey by Resende & Ribeiro in MIC 2003 book of Ibaraki, Nonobe, and Yagiura (2005).
- Main observation from experimental studies: GRASP with path-relinking outperforms pure GRASP.

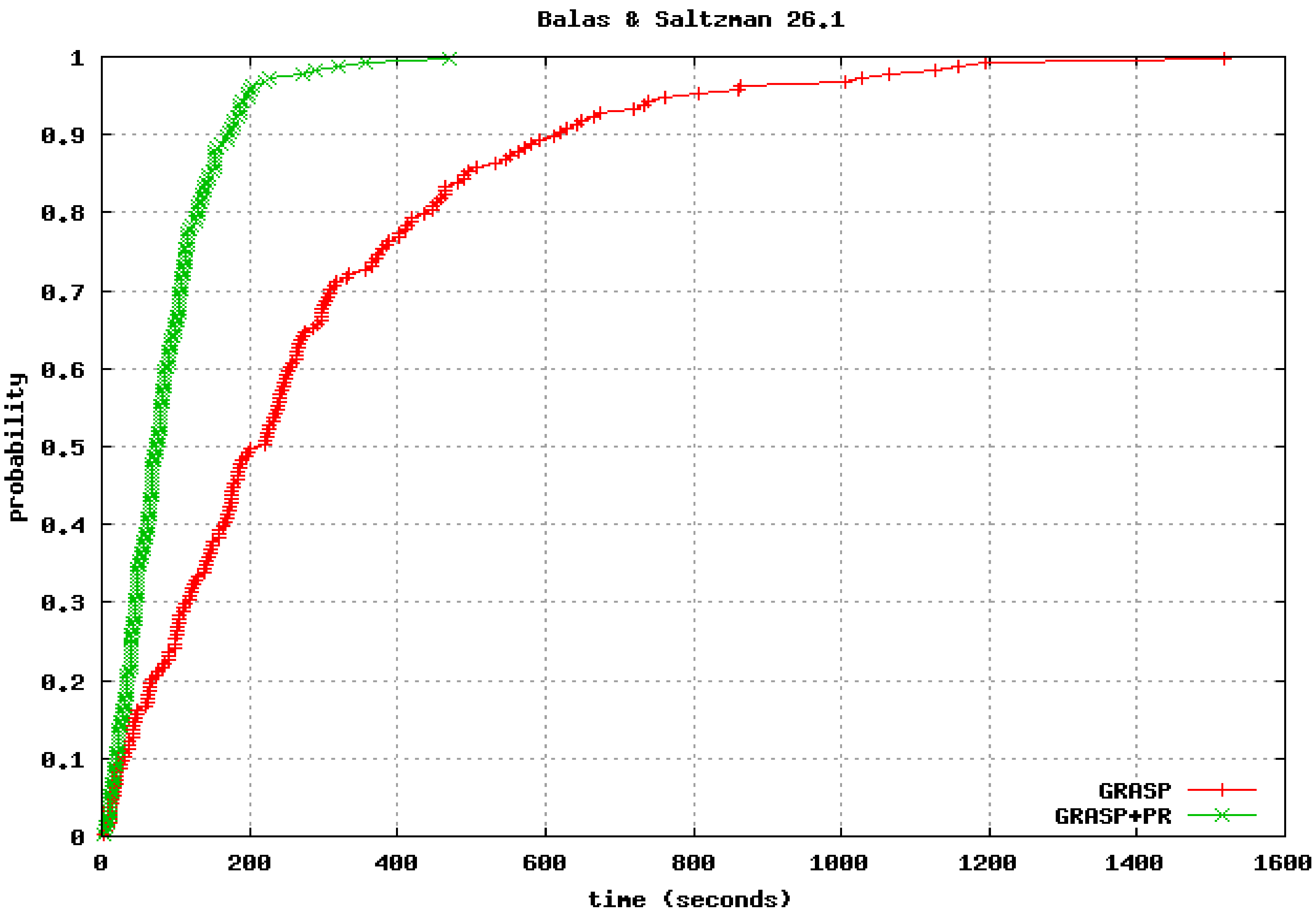
MAX-SAT (Festa, Pardalos, Pitsoulis, and Resende, 2006)



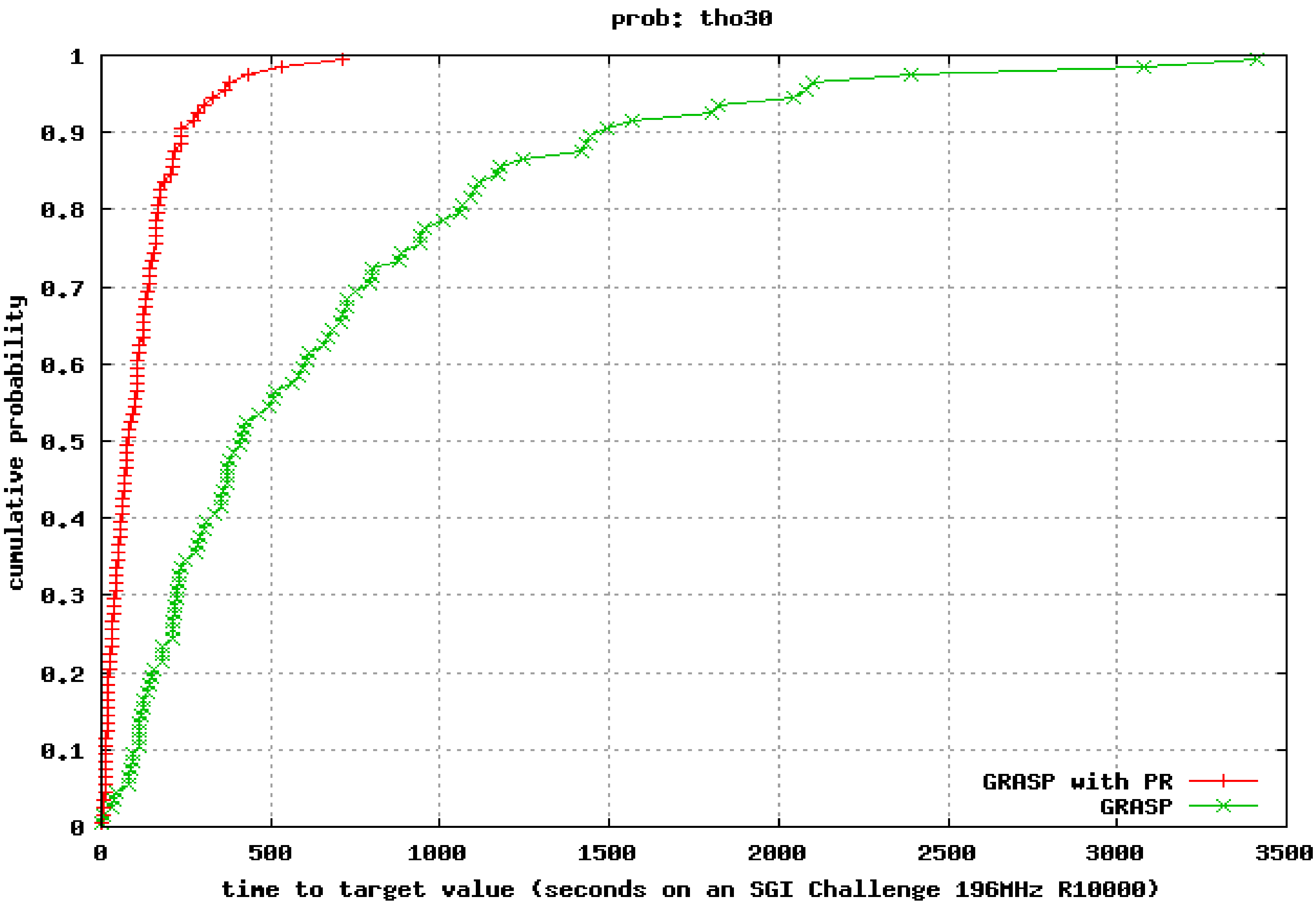
at&t

Your world. Delivered.

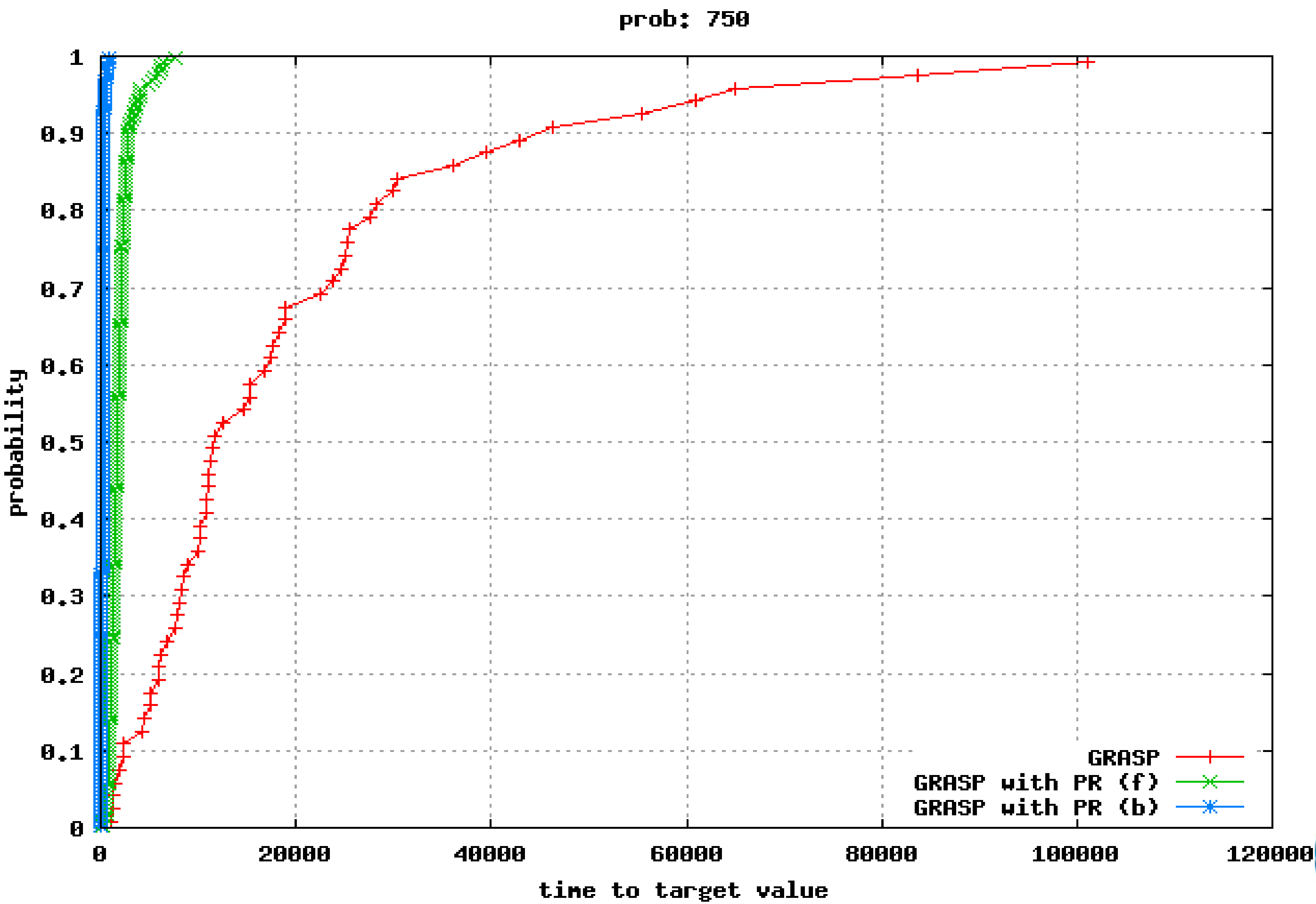
3-index assignment (Aiex, Resende, Pardalos, & Toraldo, 2005)



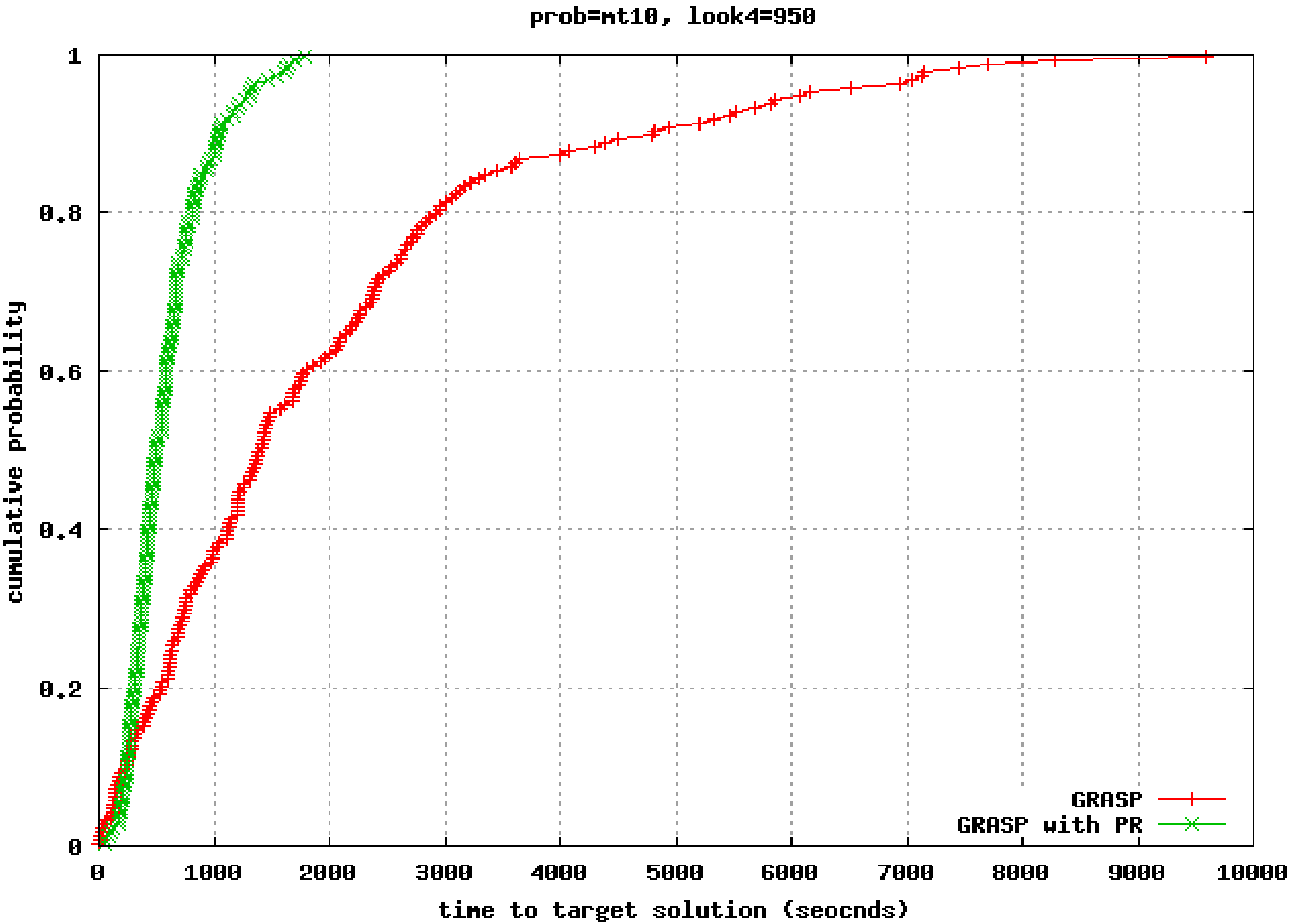
QAP (Oliveira, Pardalos, and Resende, 2004)



Bandwidth packing (Resende and Ribeiro, 2003)



Job shop scheduling (Aiex, Binato, & Resende, 2003)



GRASP with path-relinking:
Pool management

- P is a set (pool) of elite solutions.
- Ideally, pool has a set of good diverse solutions.
- Mechanisms are needed to guarantee that pool is made up of those kinds of solutions.

GRASP with path-relinking:
Pool management

- Each iteration of first |P| GRASP iterations adds one solution to P (if different from others).
- After that: solution x is promoted to P if:
  - x is better than best solution in P.
  - x is not better than best solution in P, but is better than worst and is sufficiently different from all solutions in P.

GRASP with path-relinking:
Pool management

- GRASP with PR works best when paths in PR are long, i.e. when the symmetric difference between the initial and guiding solutions is large.
- Given a solution to relink with an elite solution, which elite solution to choose?
  - Choose at random with probability proportional to the symmetric difference.

GRASP with path-relinking:
Pool management

- Solution quality and diversity are two goals of pool design.
- Given a solution X to insert into the pool, which elite solution do we choose to remove?
  - Of all solutions in the pool with worse solution than X, select to remove the pool solution most similar to X, i.e. with the smallest symmetric difference from X.

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

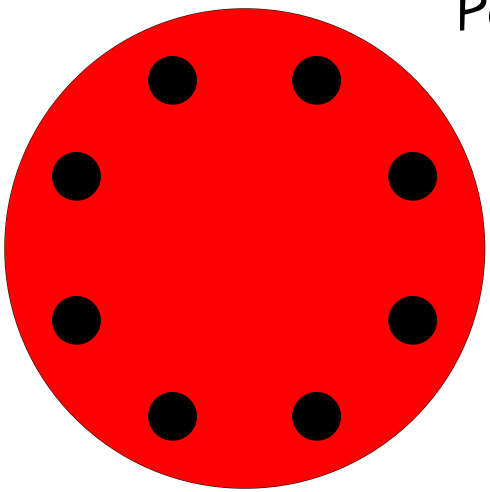
Evolutionary path-relinking (EvPR)

Evolutionary path-relinking

(Resende & Werneck, 2004, 2006)

- 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 an intensification procedure at certain points of the solution process;
  - as a post-optimization procedure at the end of the solution process.

Evolutionary path-relinking (EvPR)

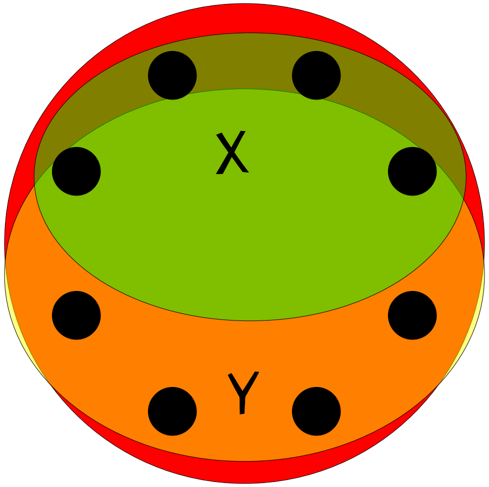


Population P(0)

Each "population" of EvPR starts with a pool of elite solutions of size |P|.

Population P(0) is the current elite set.

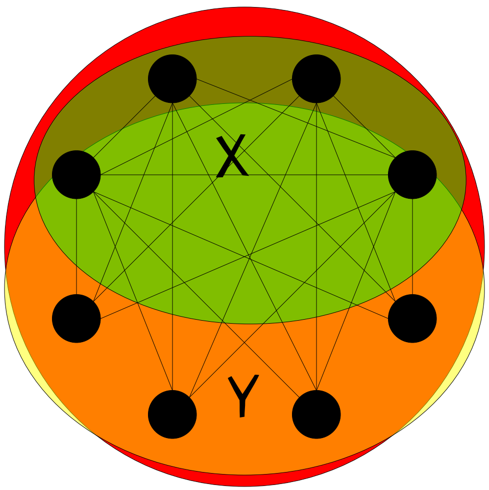
Evolutionary path-relinking (EvPR)



All pairs of elite solutions (x,y) in K-th population P(K), such that $x \in X \subseteq P(K)$ and $y \in Y \subseteq P(K)$, are path-relinked and the resulting $z = PR(x,y)$ is a candidate for inclusion in population P(K+1).

Rules for inclusion into P(K+1) are the same used for inclusion into any pool.

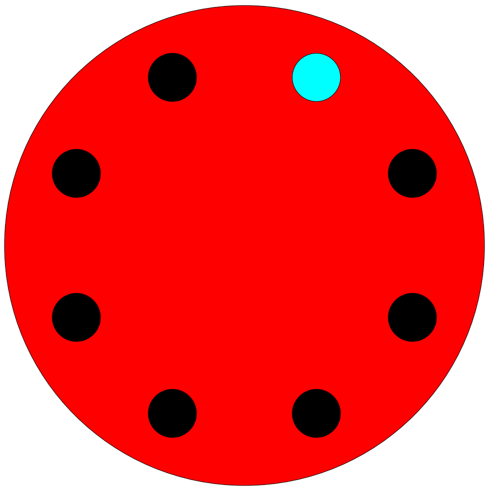
Evolutionary path-relinking (EvPR)



All pairs of elite solutions (x,y) in K-th population P(K), such that $x \in X \subseteq P(K)$ and $y \in Y \subseteq P(K)$, are path-relinked and the resulting $z = PR(x,y)$ is a candidate for inclusion in population P(K+1).

Rules for inclusion into P(K+1) are the same used for inclusion into any pool.

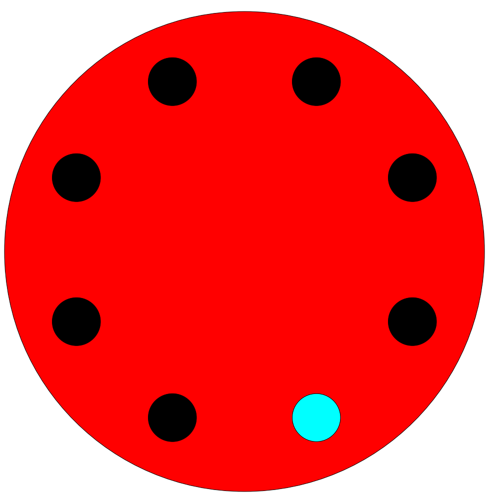
Evolutionary path-relinking (EvPR)



Population P(K)

If best solution in population P(K+1) has same objective function value as best solution in population P(K), process stops.

Else K=K+1 and repeat.

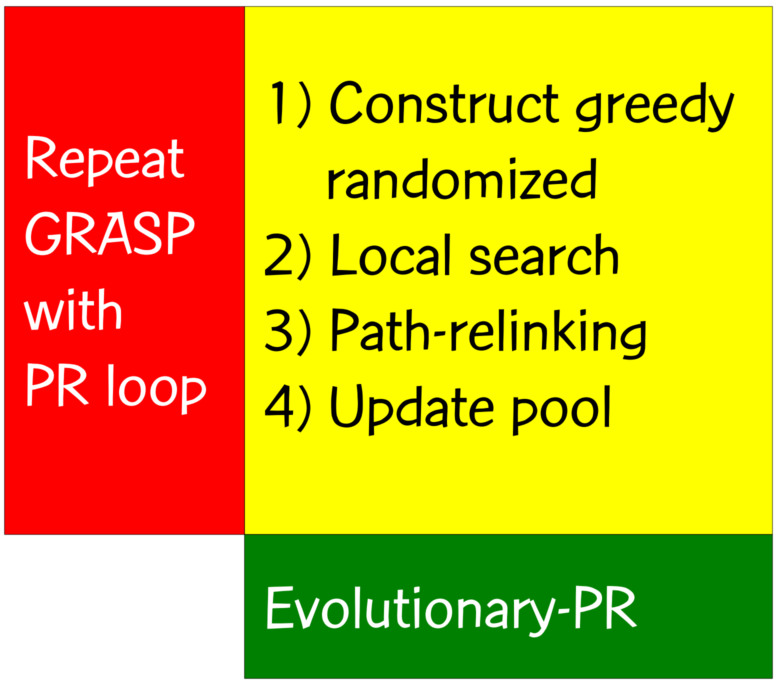


Population P(K+1)

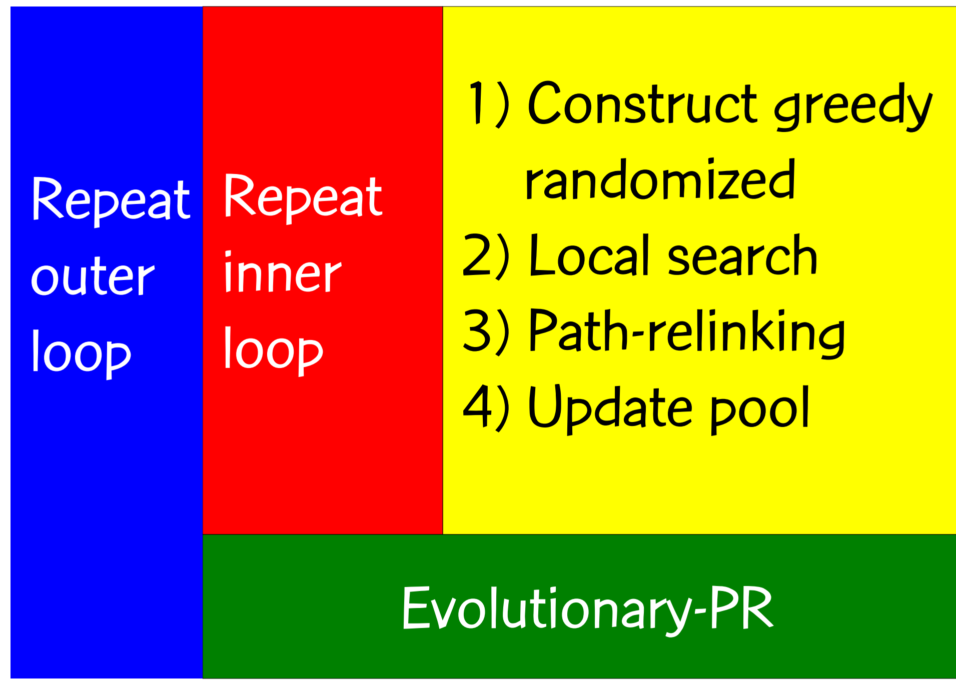
GRASP with evolutionary path-relinking

GRASP with evolutionary path-relinking

As post-optimization



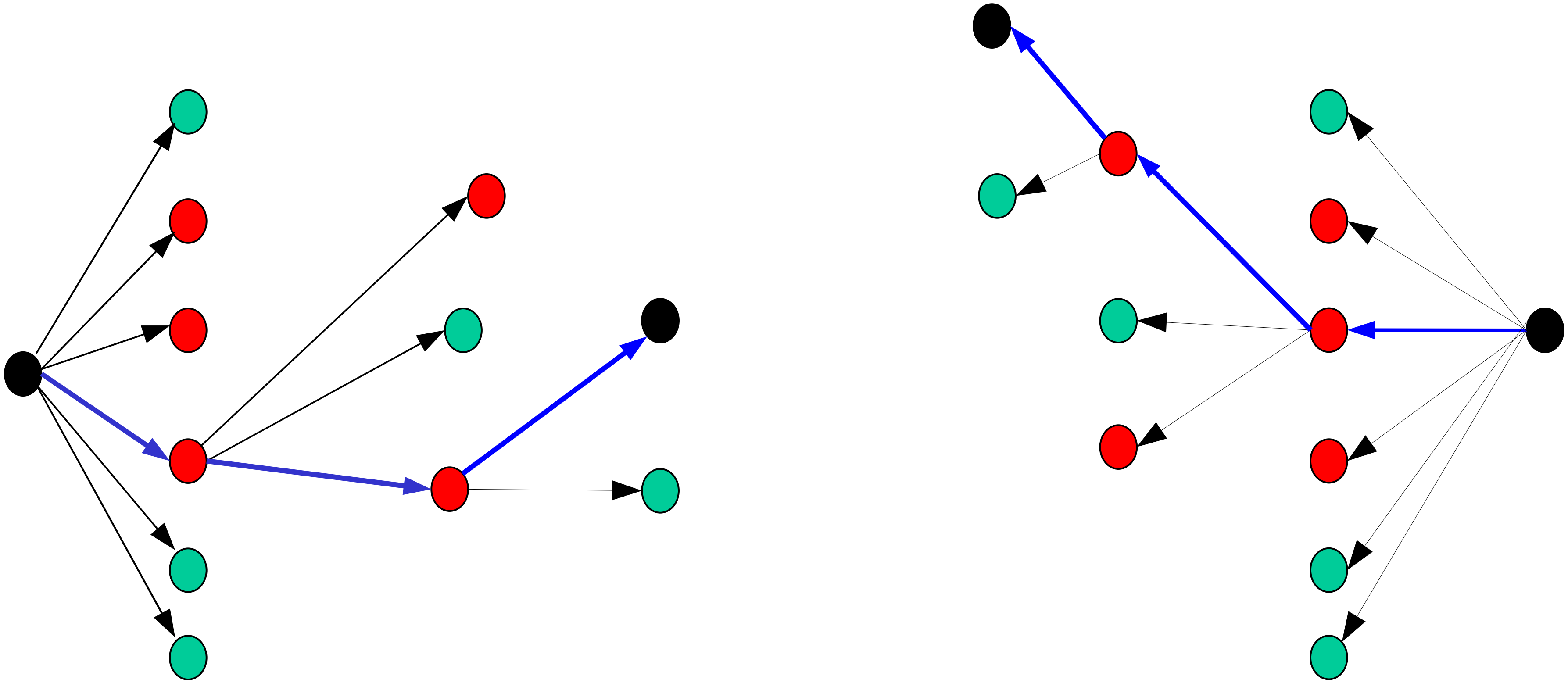
During GRASP + PR



(Resende & Werneck, 2004, 2006)

GRASP with EvPR: Implementation ideas

Truncated mixed graPR

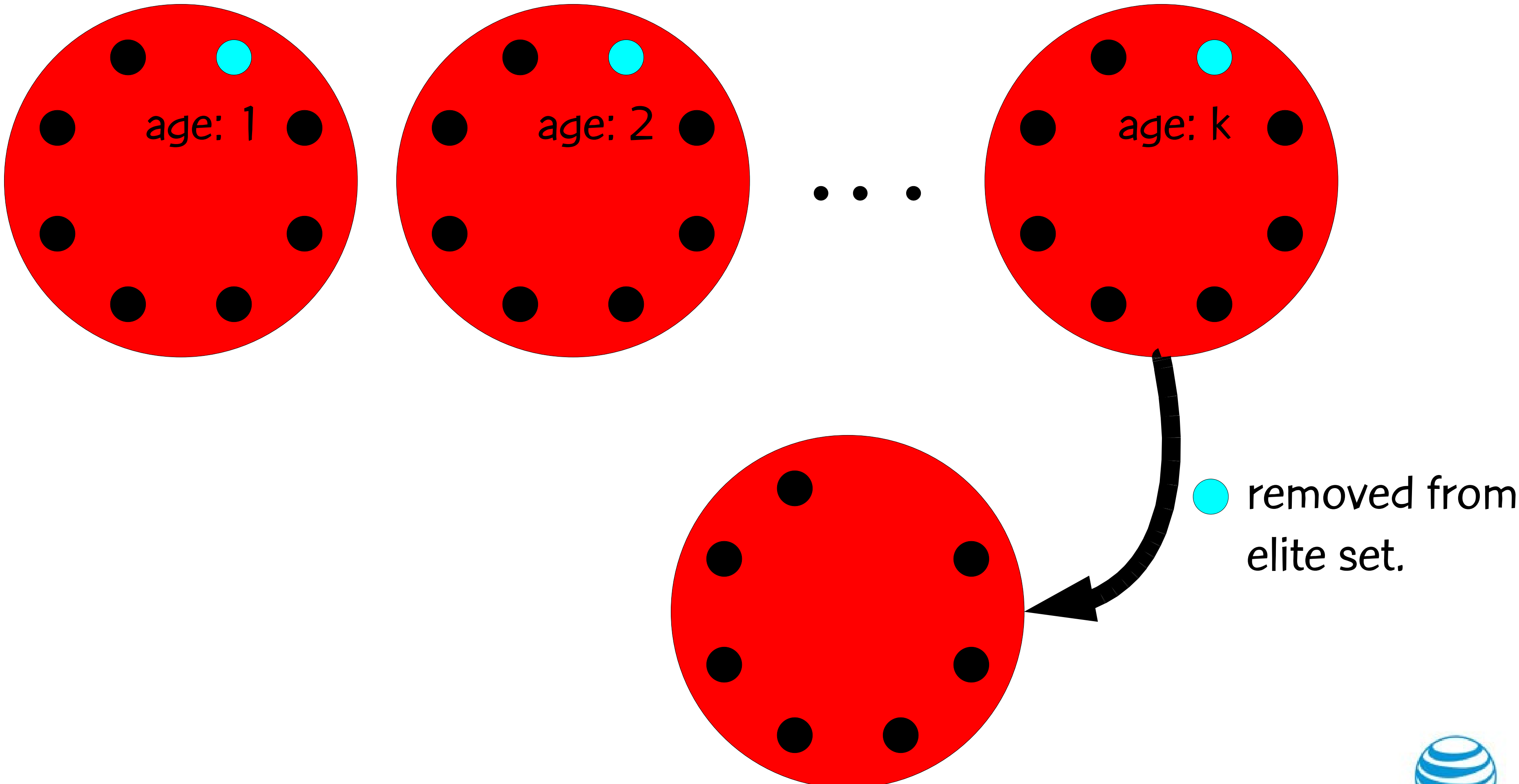


In PR and EvPR, apply one iteration of graPR.

For (x,y), different calls to graPR(x,y) explore different paths.

GRASP with EvPR: Implementation ideas

Force old low-quality elite solutions out

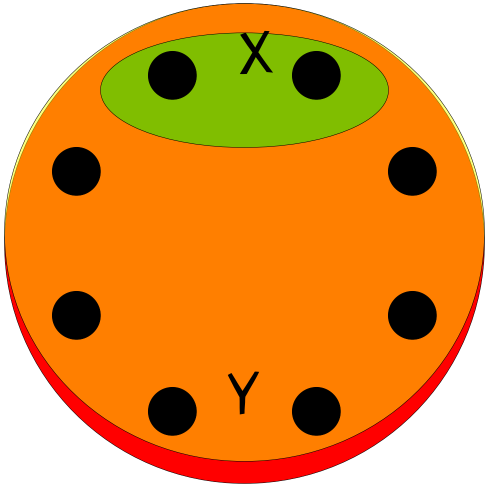


at&t
Your world. Delivered.

GRASP with EvPR: Implementation ideas

Make set X small and with best pool solutions.

Make set Y be entire pool.



Use set X of size 1 or 2.

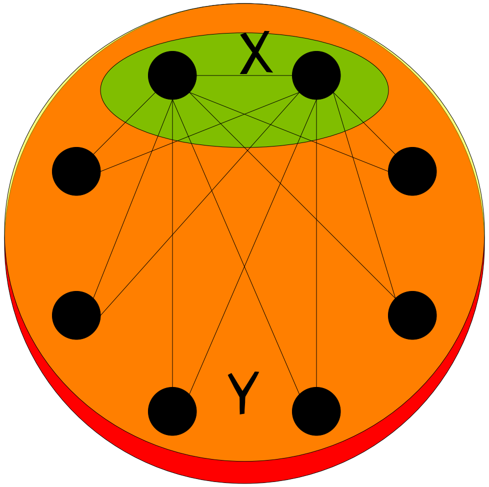
Speeds up EvPR.

Avoids unfruitful calls to graPR(x,y)

GRASP with EvPR: Implementation ideas

Make set X small and with best pool solutions.

Make set Y be entire pool.



Use set X of size 1 or 2.

Speeds up EvPR.

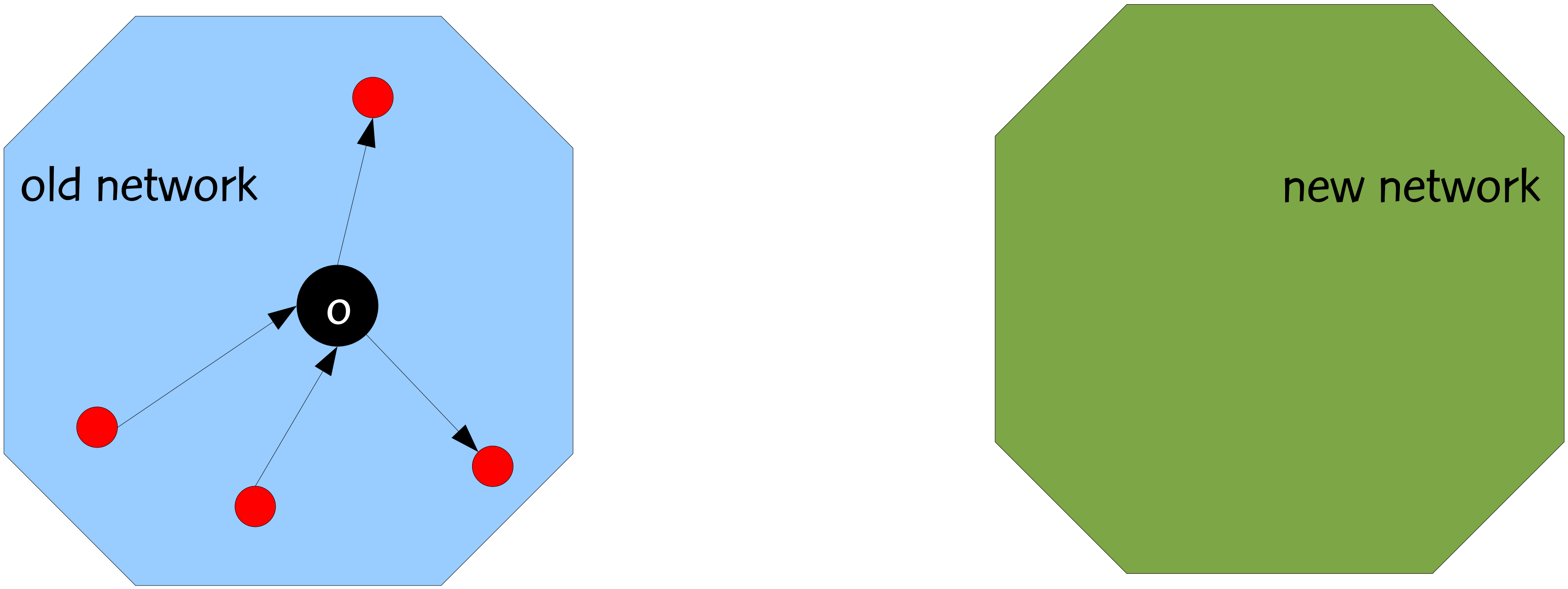
Avoids unfruitful calls to graPR(x,y)

Experimetal results: Network traffic migration scheduling

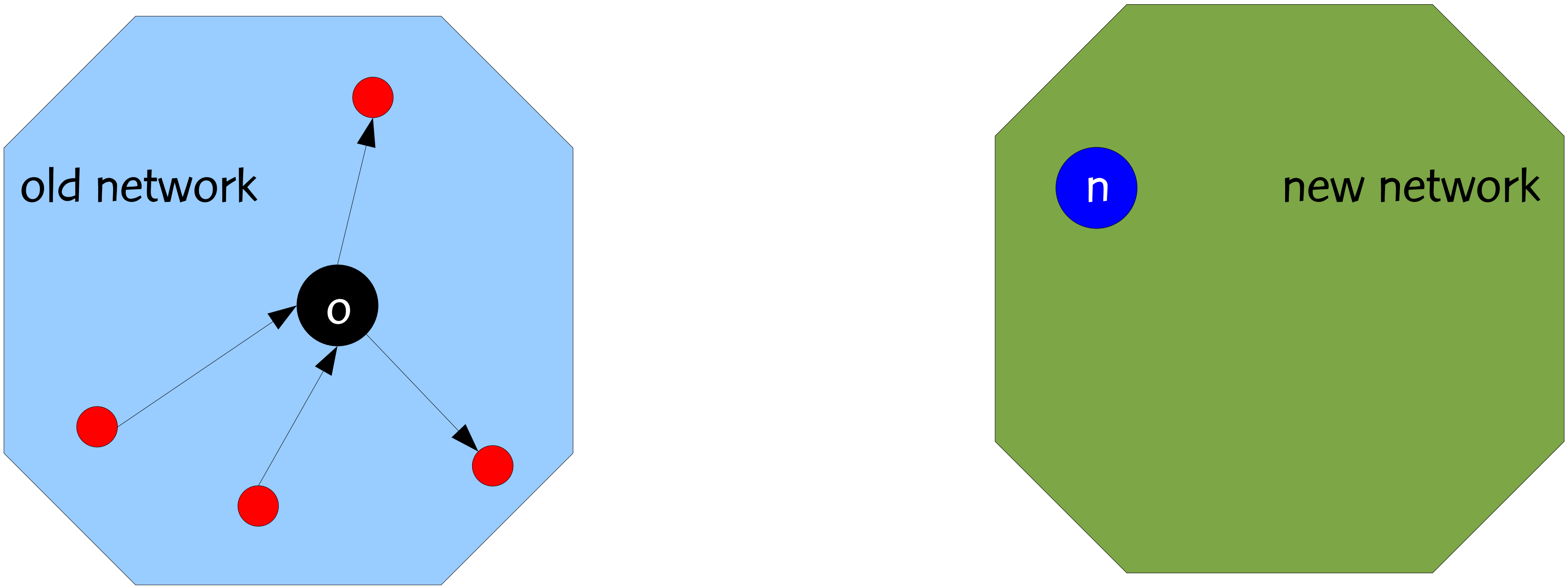
Network traffic migration scheduling

- Traffic from outdated telecommunications network is to be migrated to a new network.
  - e.g. phone traffic is to migrate from 4ESS switch-based network to IP router-based network.
- Nodes in old network are decommissioned, one at a time, and all traffic originating or terminating at the node is moved to a specific node in the new network.

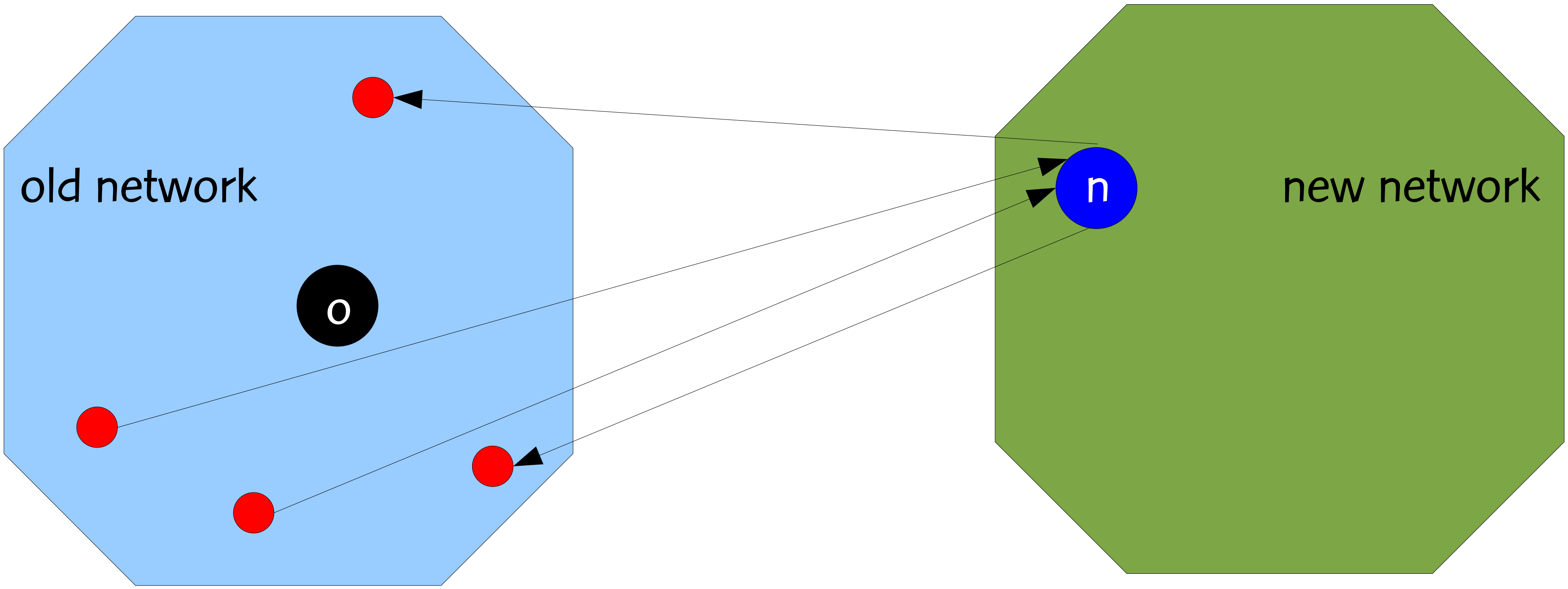
Node decommissioning



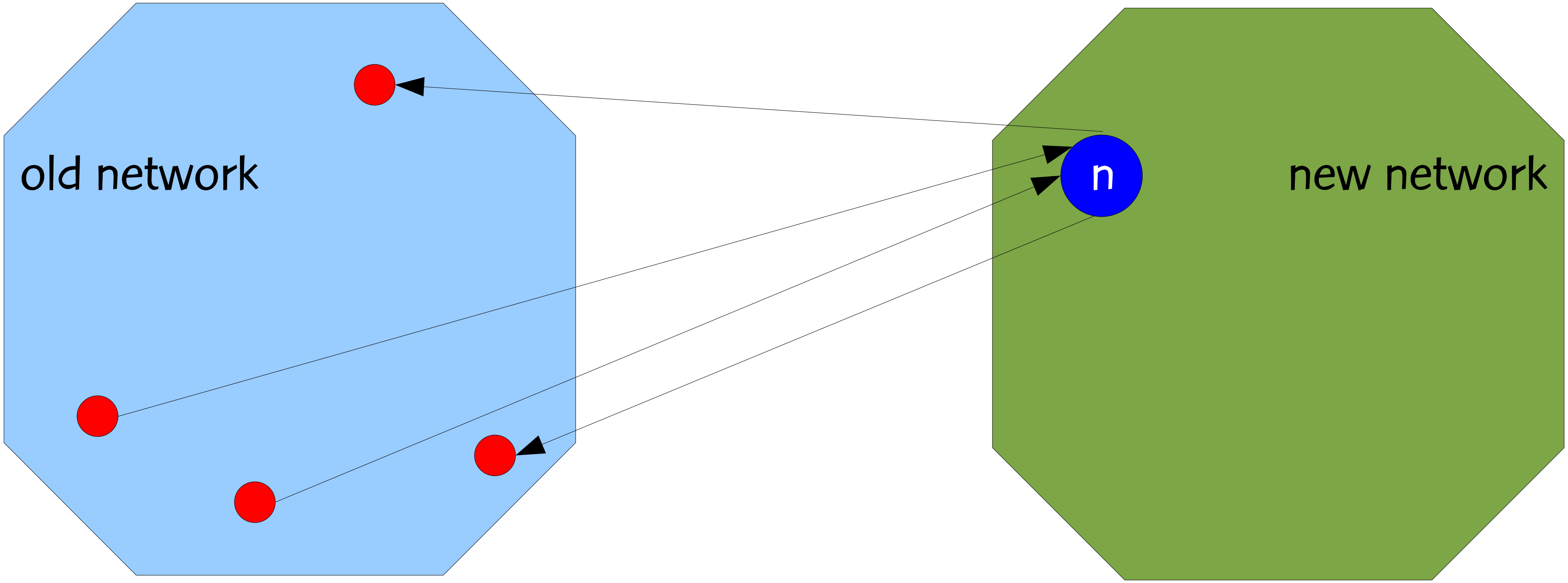
Node decommissioning



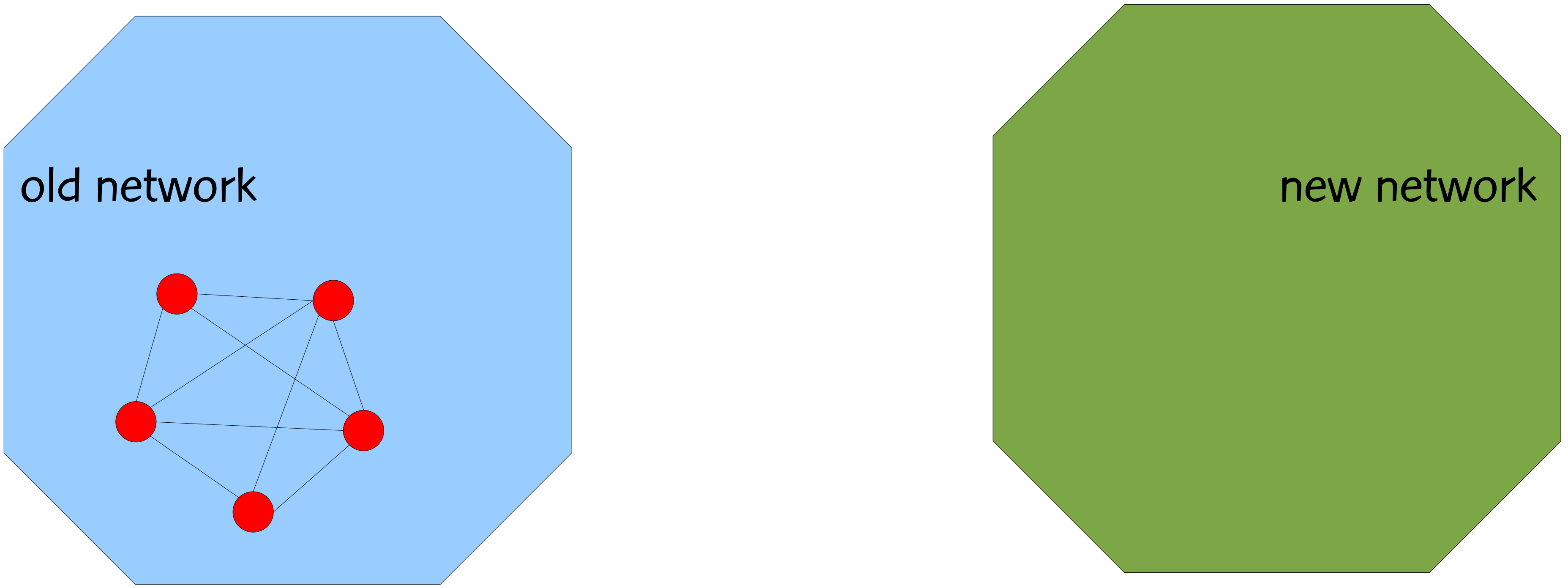
Node decommissioning



Node decommissioning

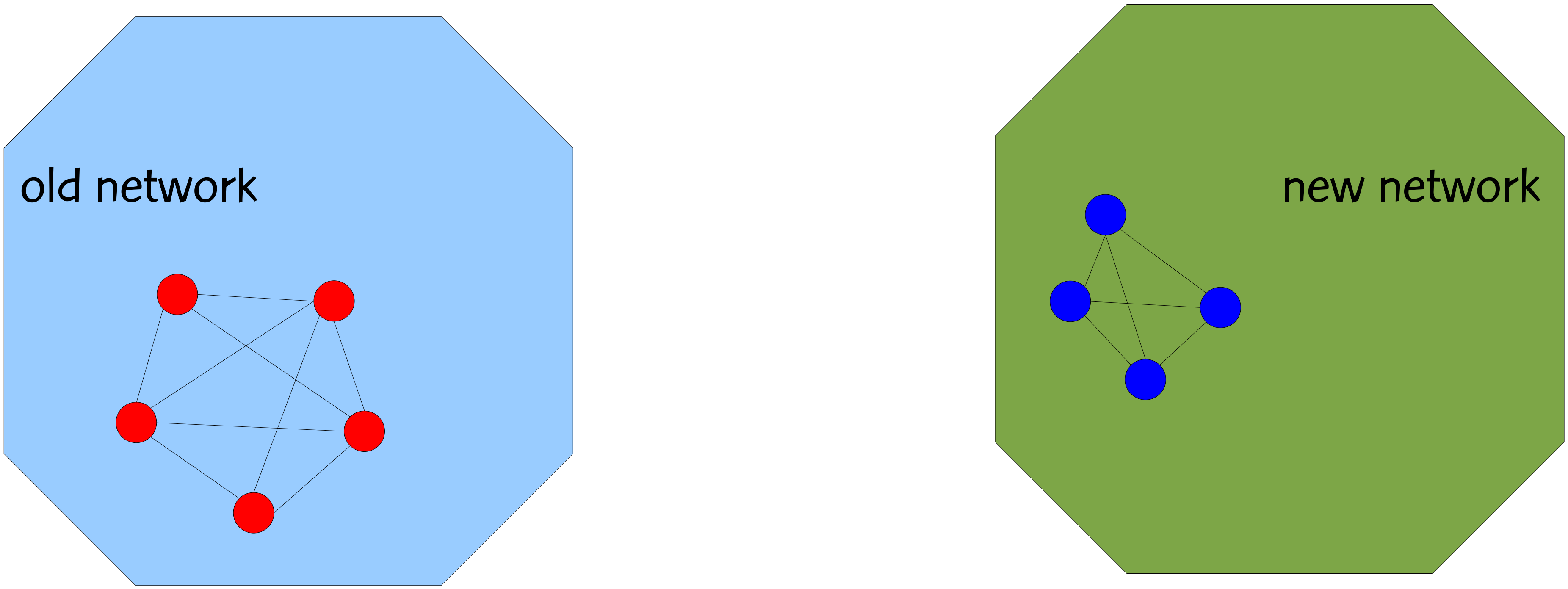


After partial decommissioning of nodes



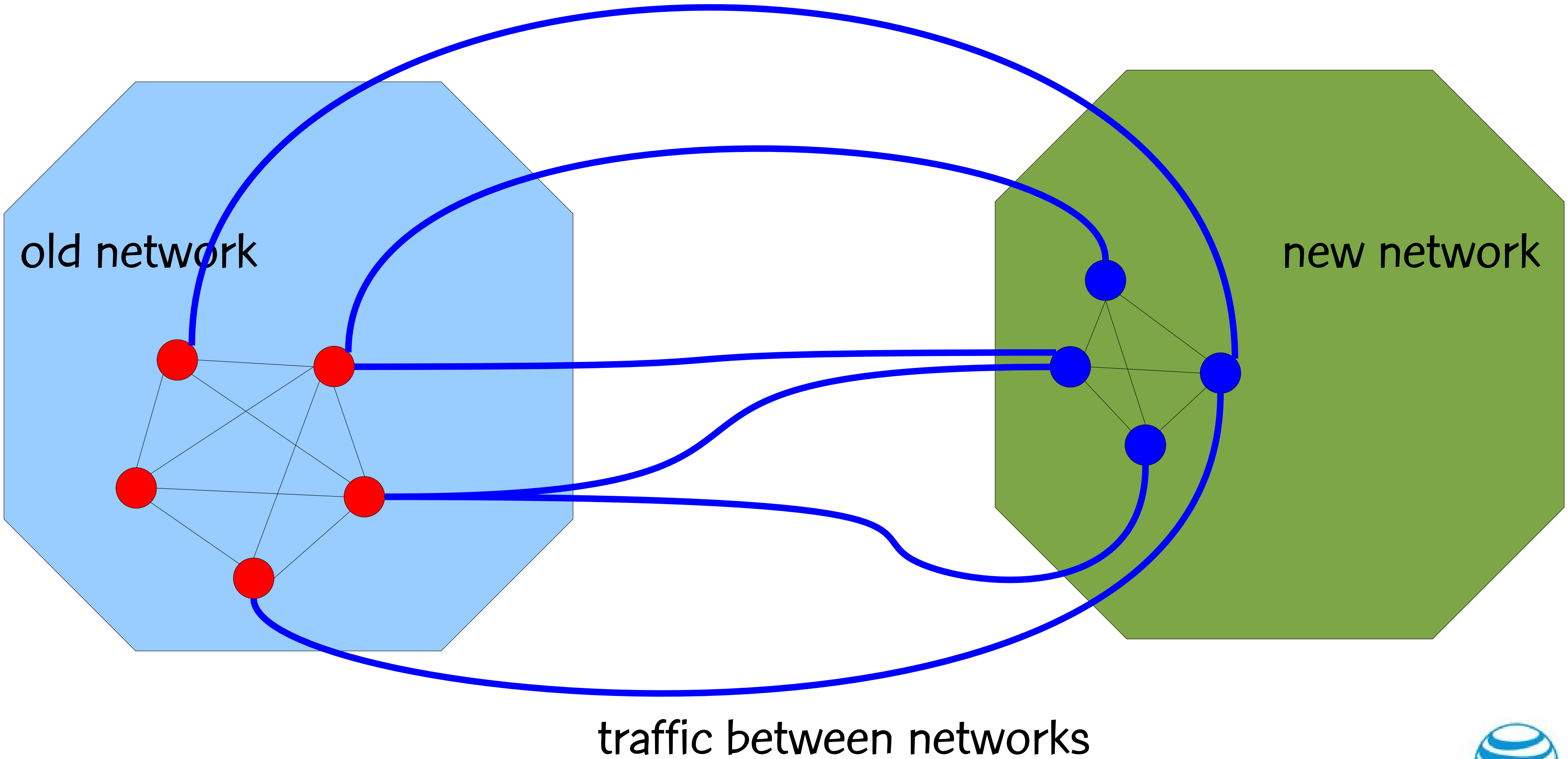
traffic in old network

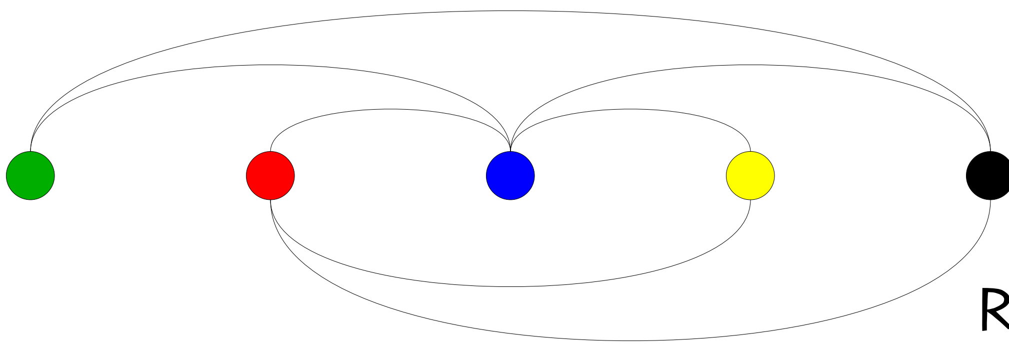
After partial decommissioning of nodes



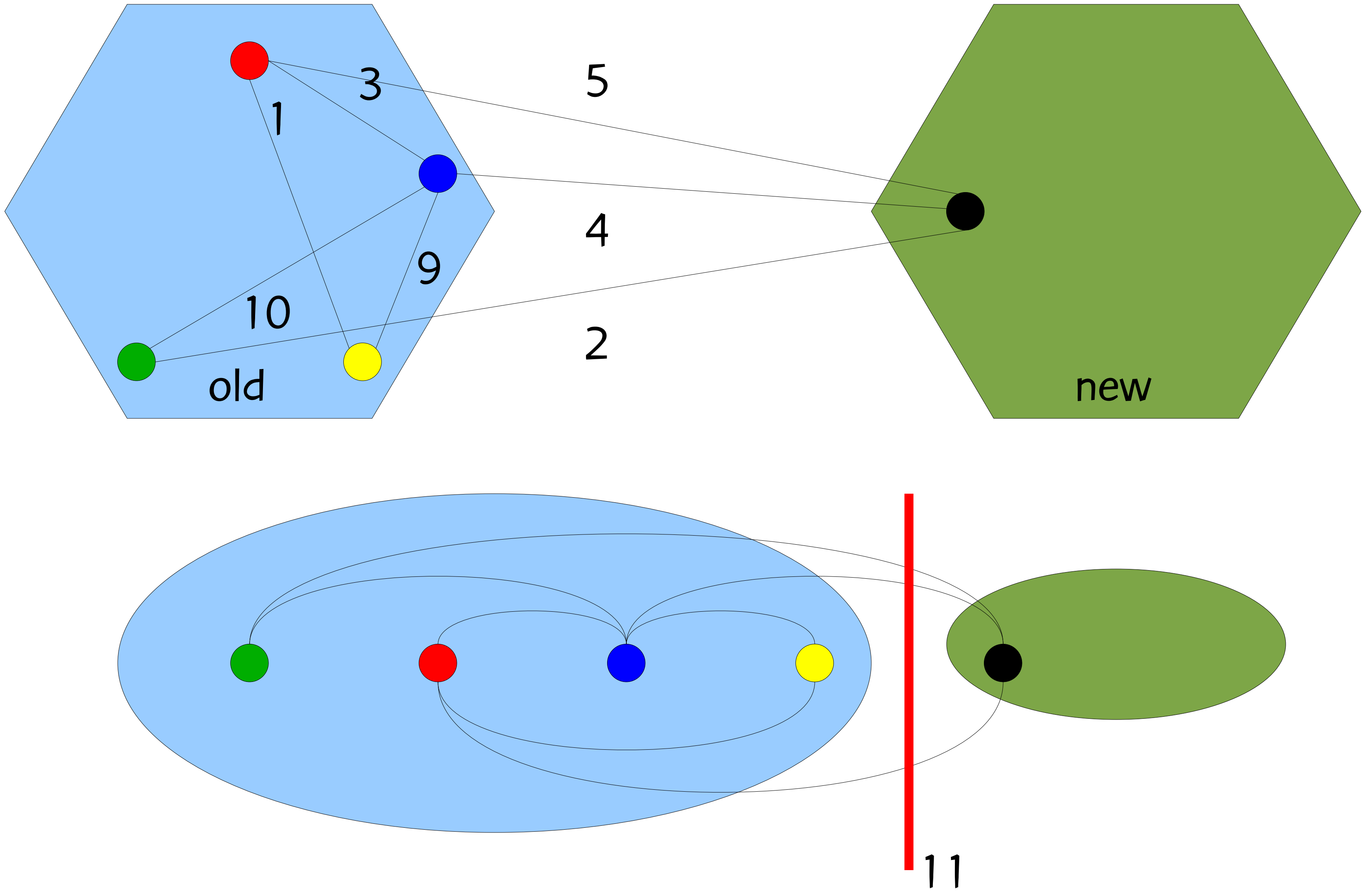
traffic in new network

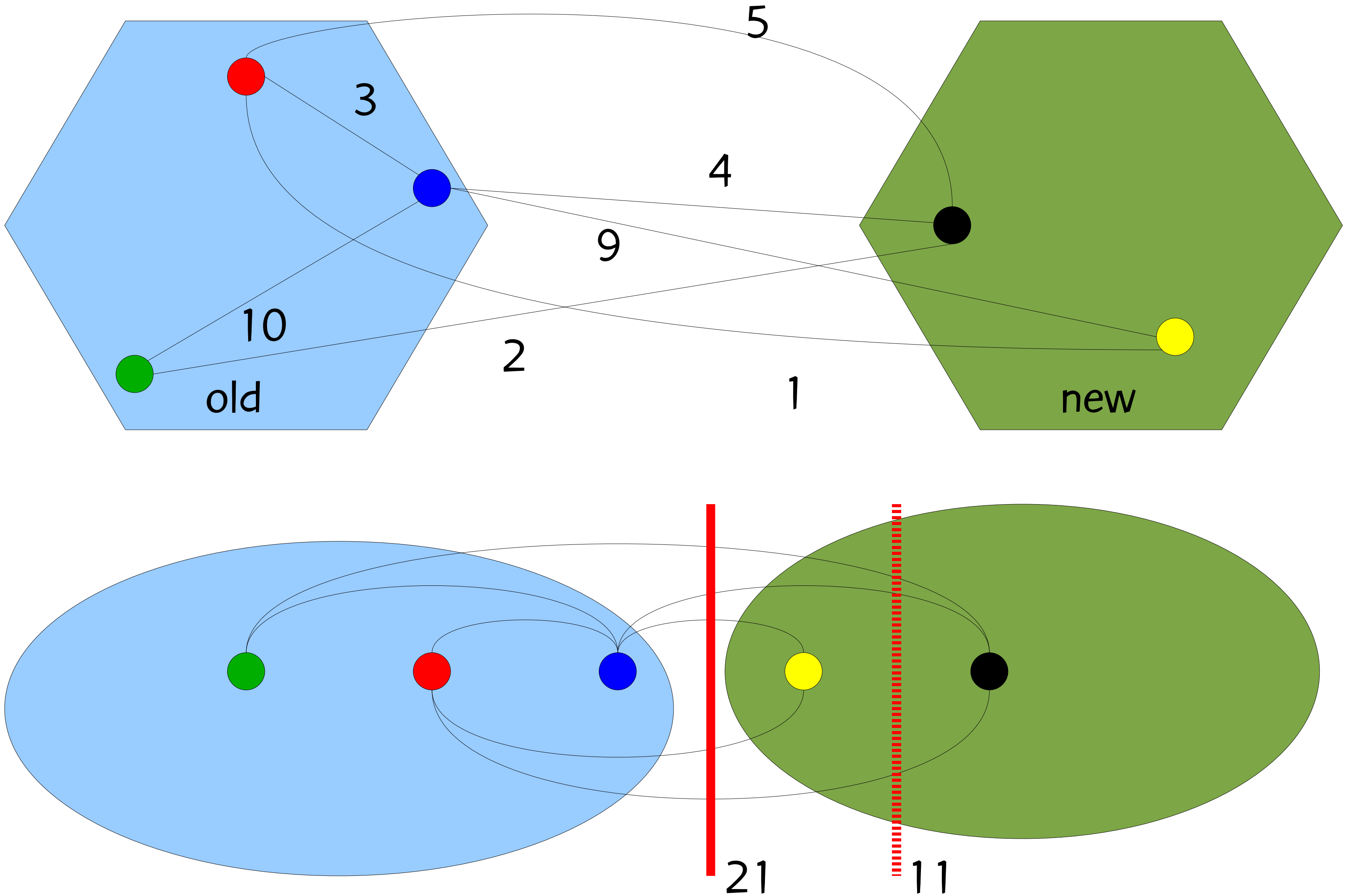
After partial decommissioning of nodes

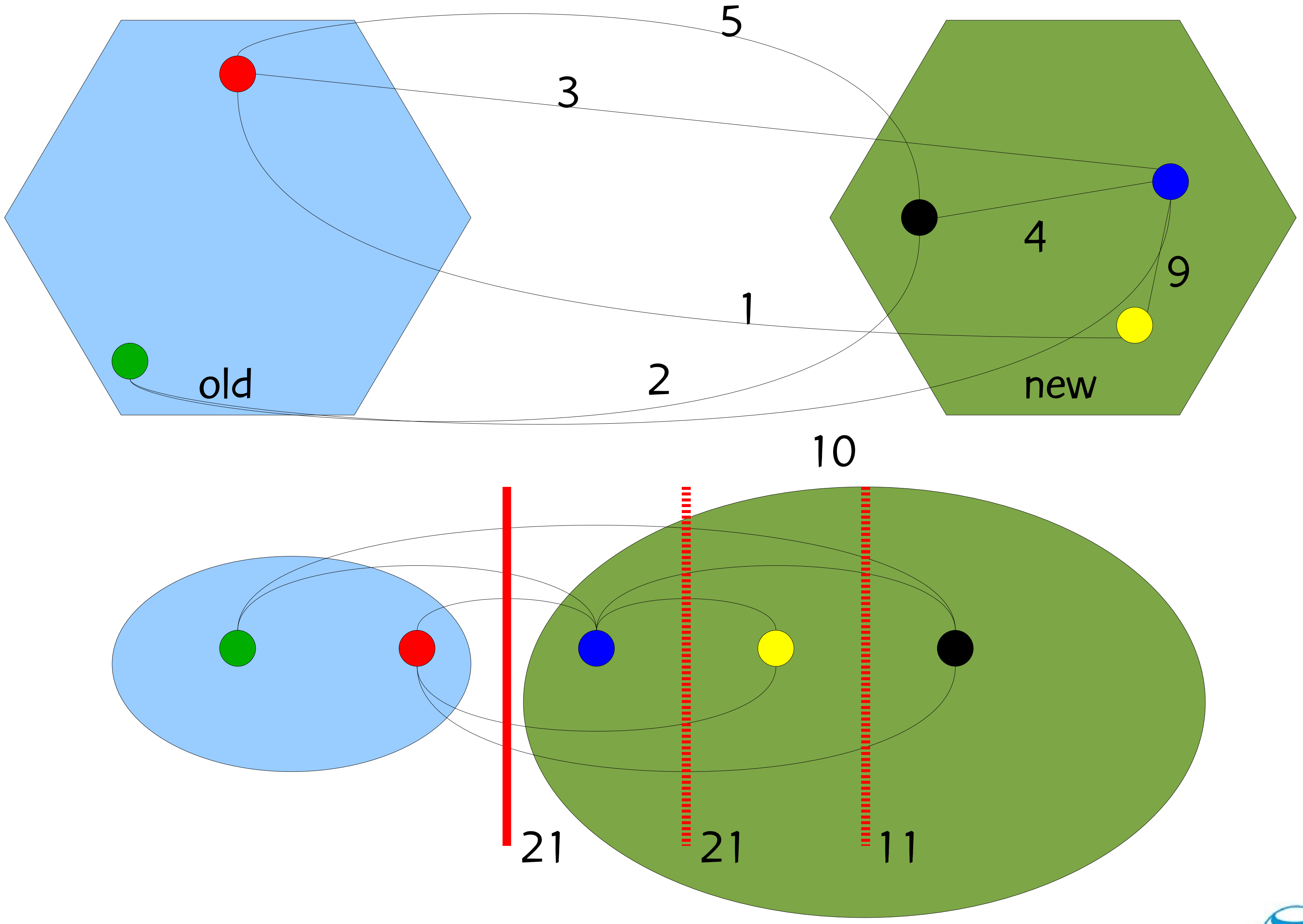


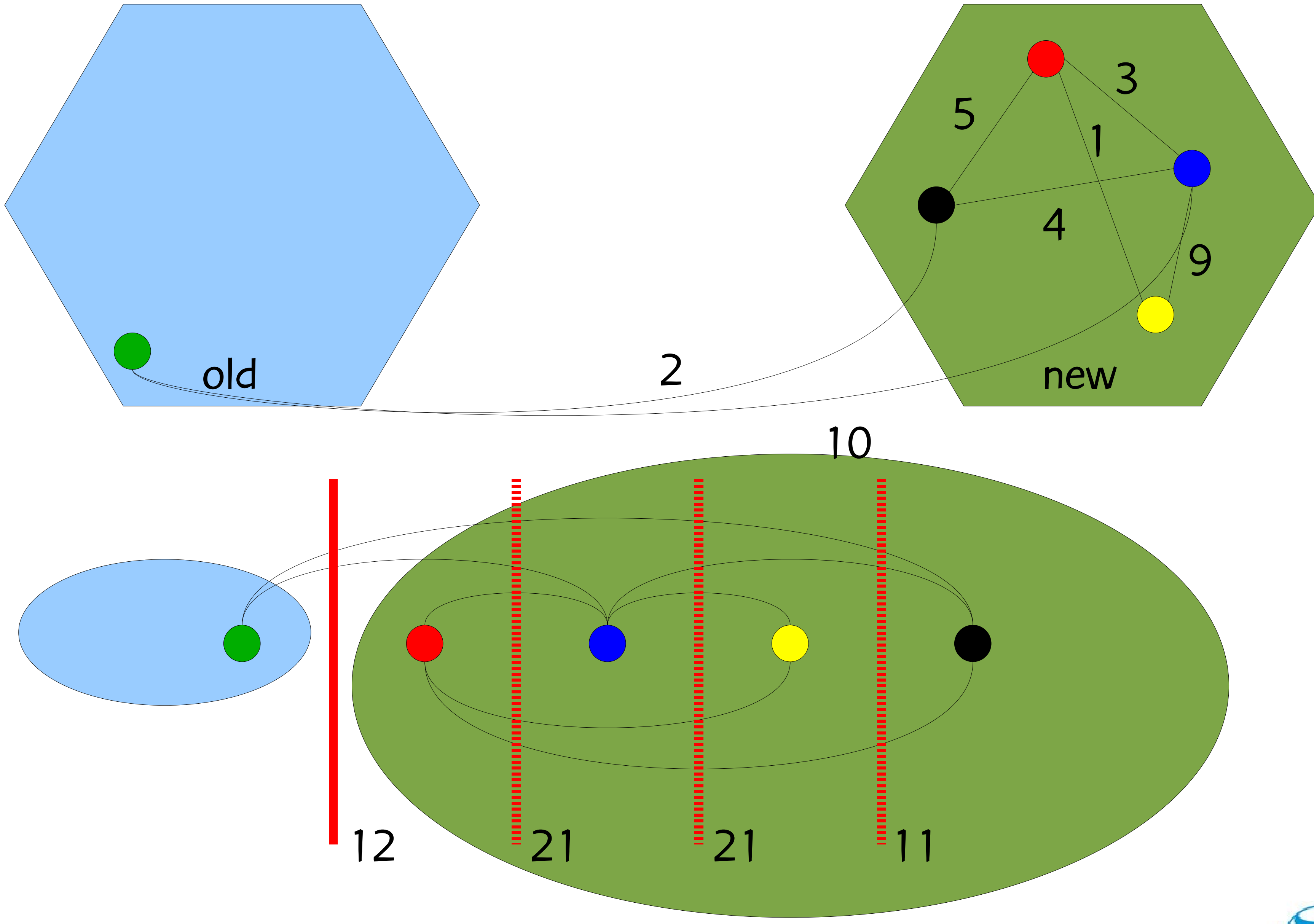


Redraw graph with nodes in line giving order in which nodes are migrated.

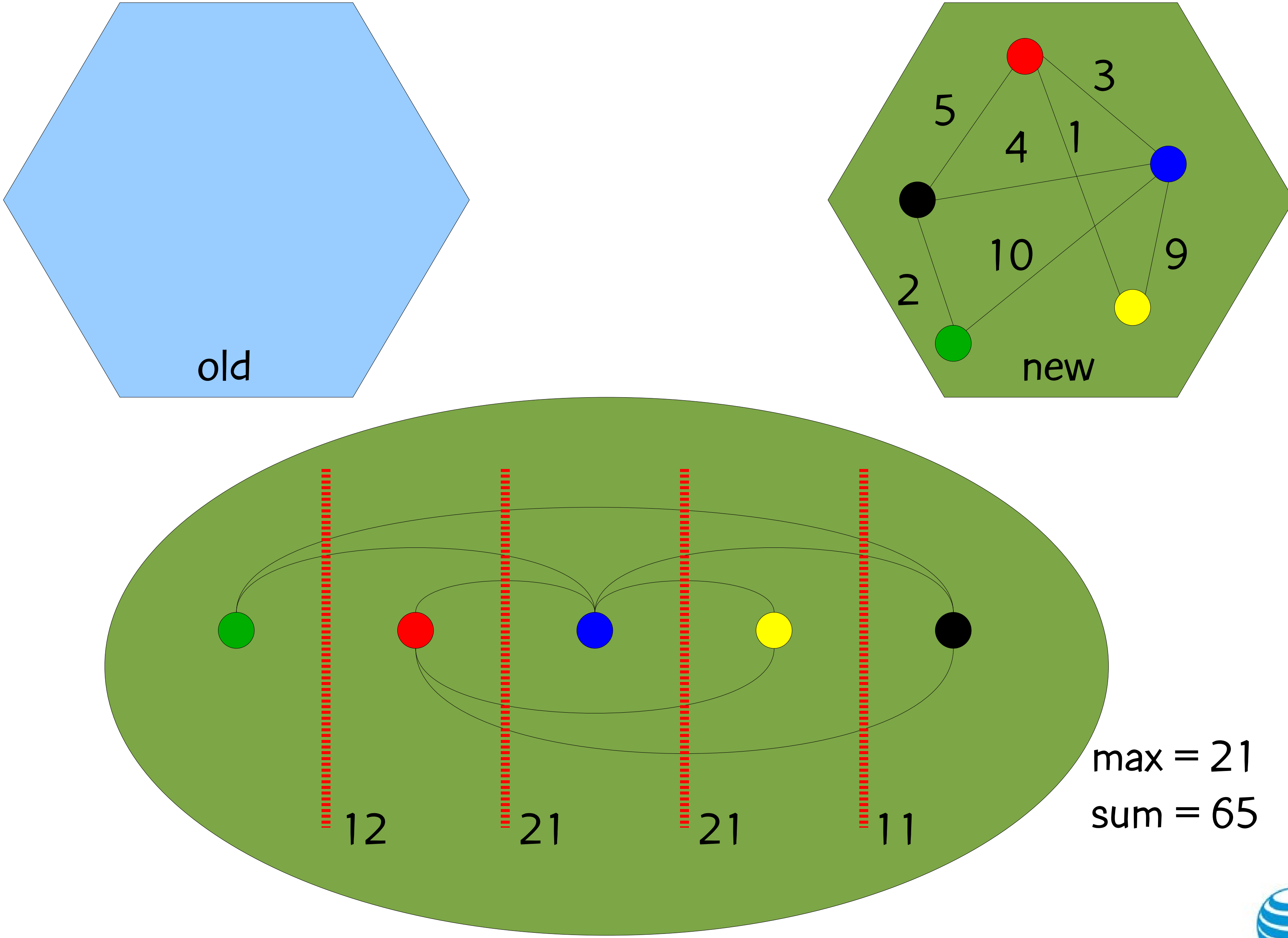

3
5
1
4
9
10
2
old
new
11

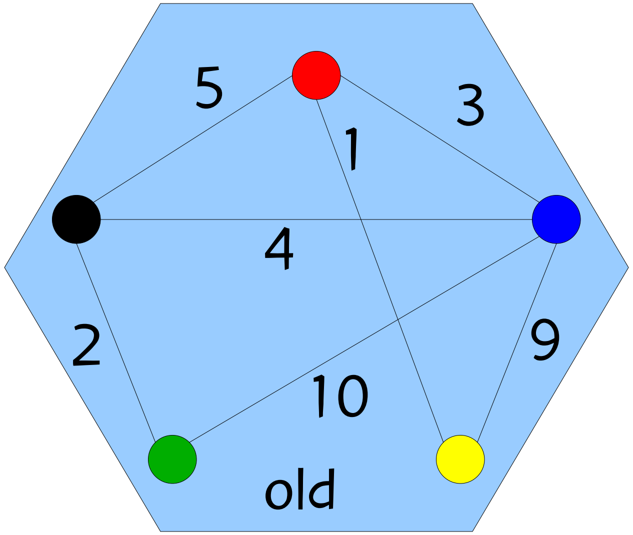

5
3
4
9
10
2
1
old
new
21
11


5
3
4
9
1
2
old
new
10
21
21
11

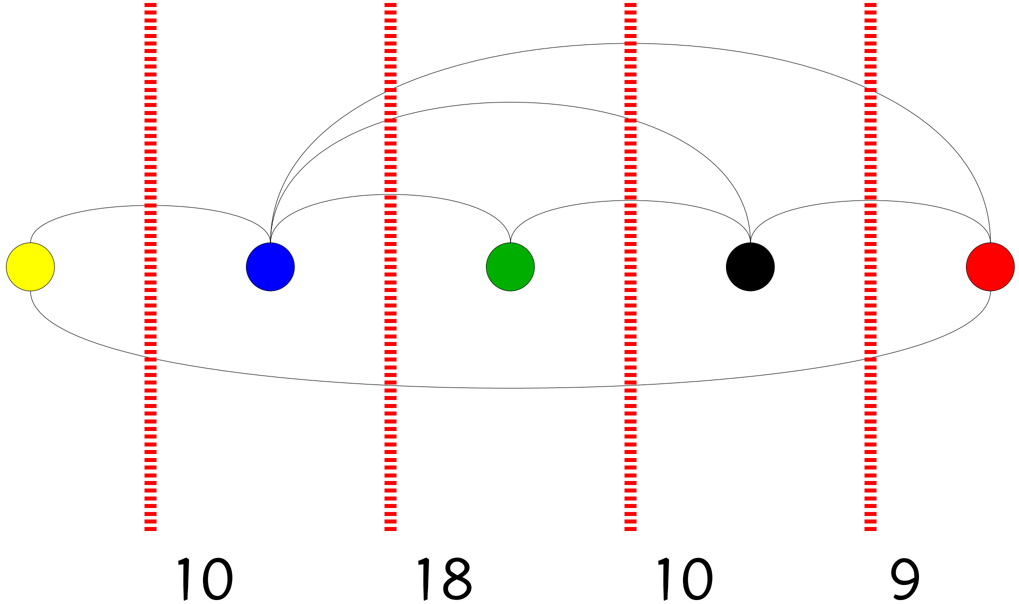


3
5
1
4
9
old
2
new
10
12
21
21
11


old
5
3
4
1
10
9
2
new
12
21
21
11
max = 21
sum = 65

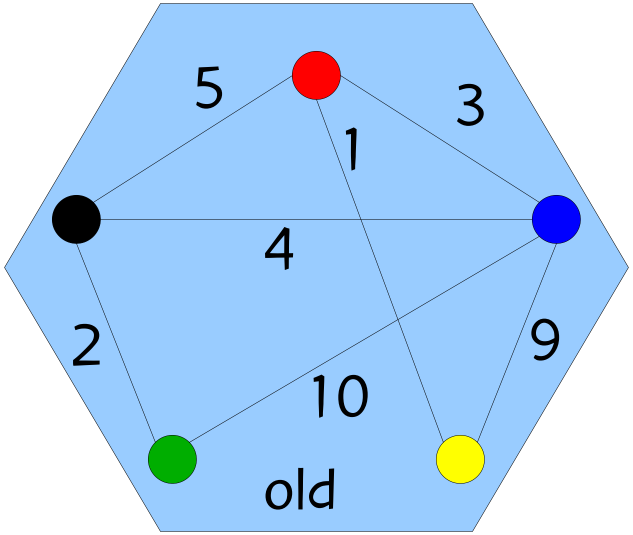


Consider another ordering.

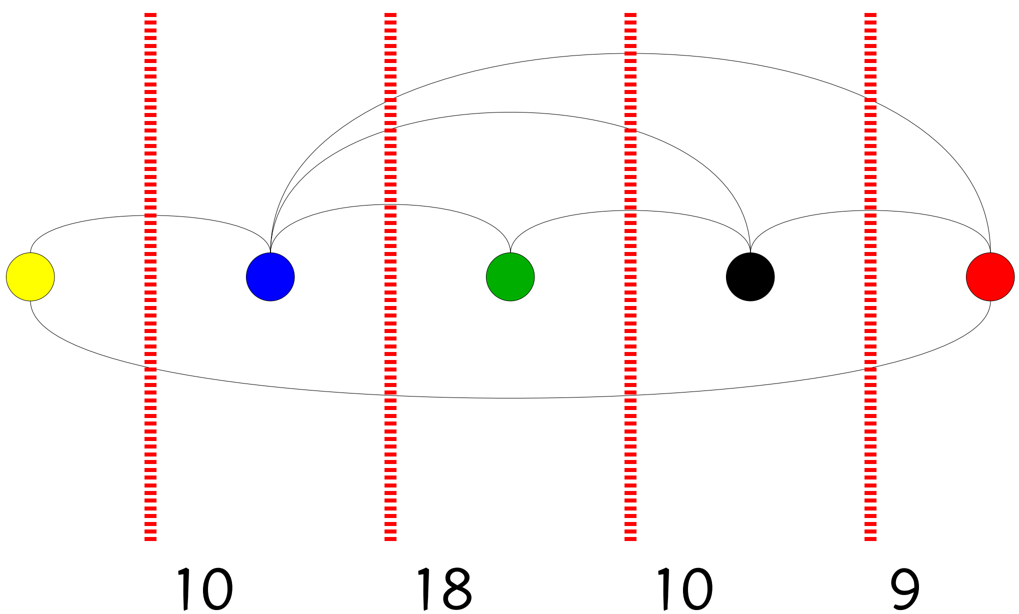


max = 18 < 21

sum = 47 < 65



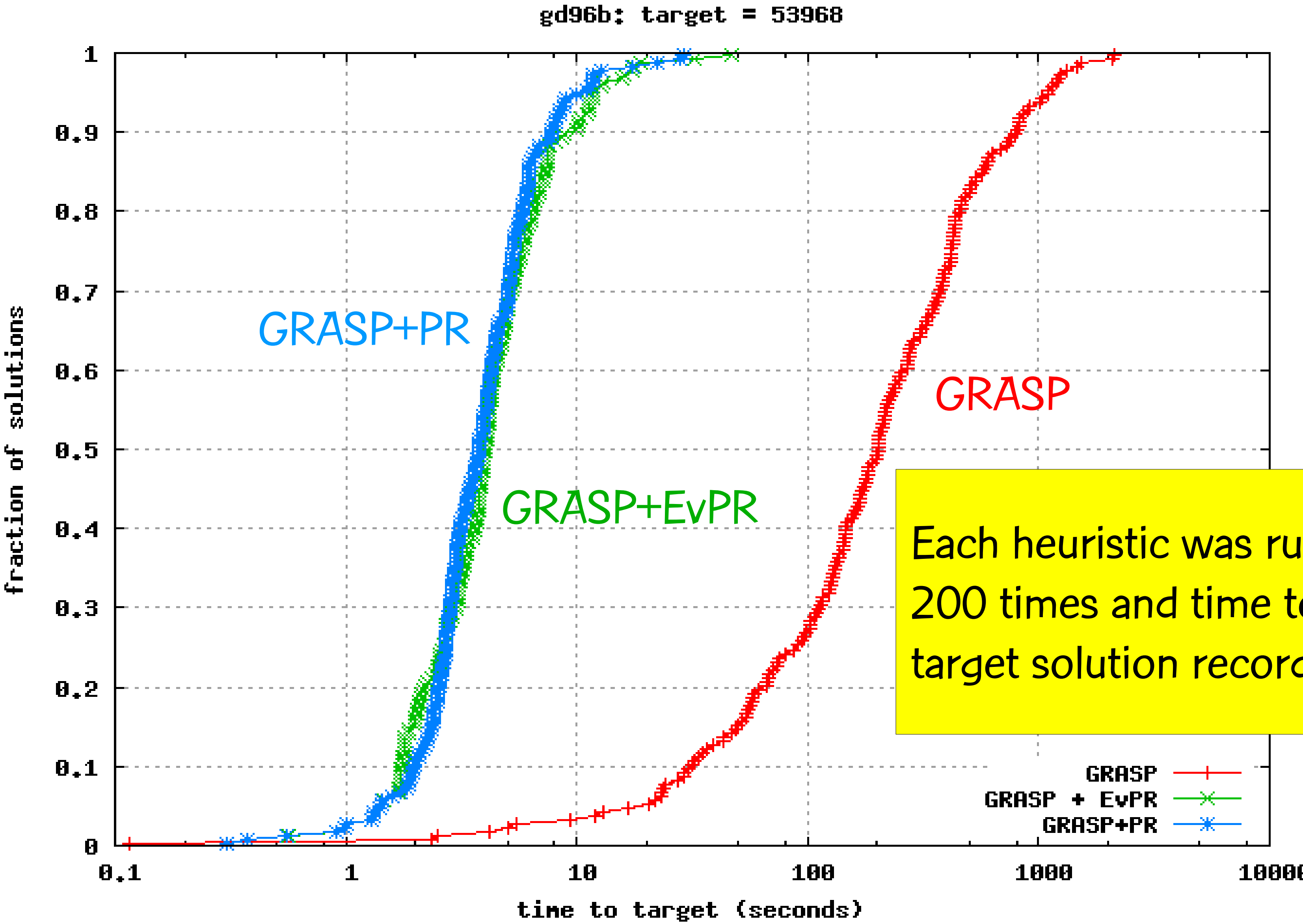
Optimization problem: Find best decommissioning sequence.



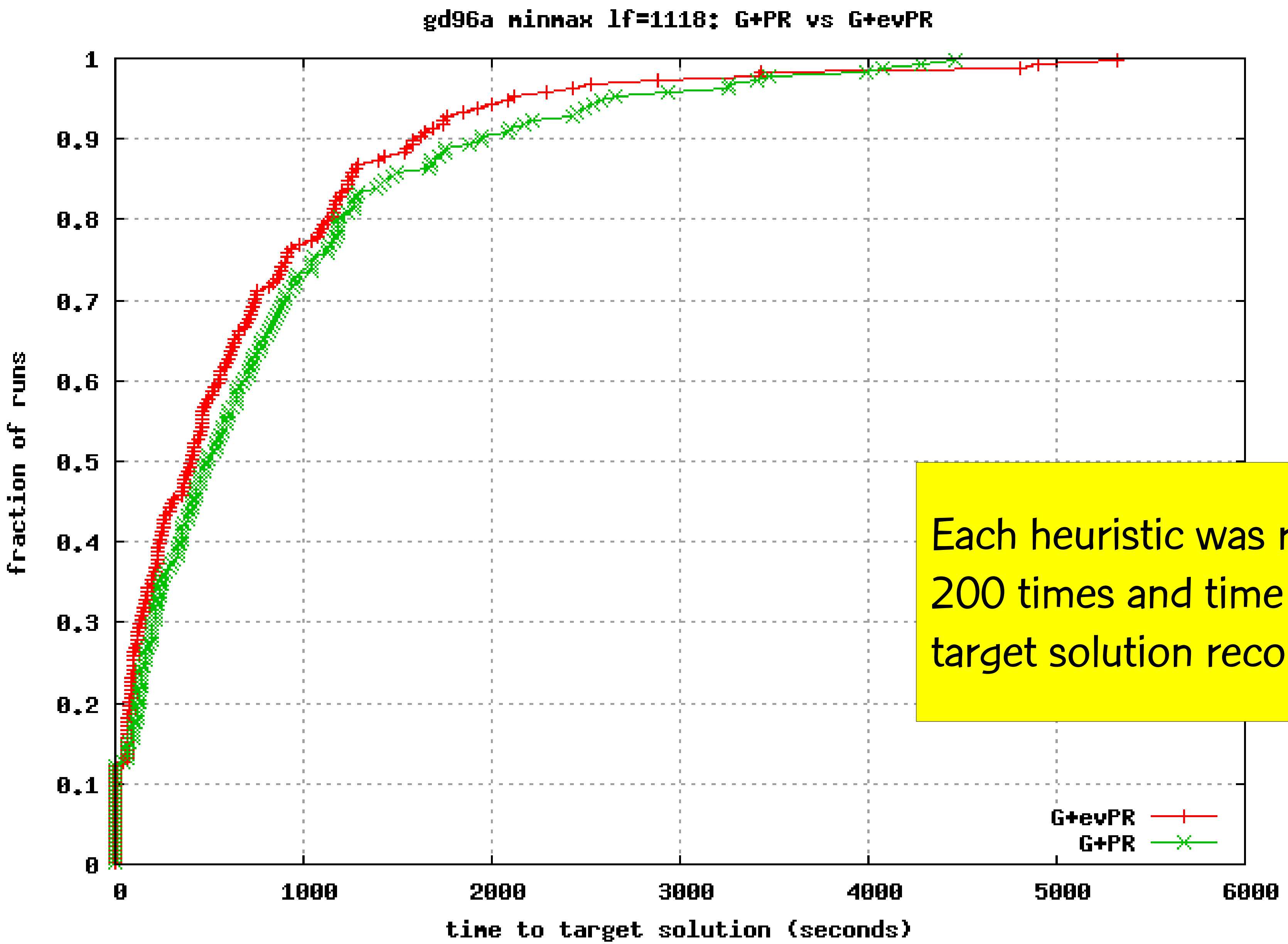
max = 18 < 21

sum = 47 < 65

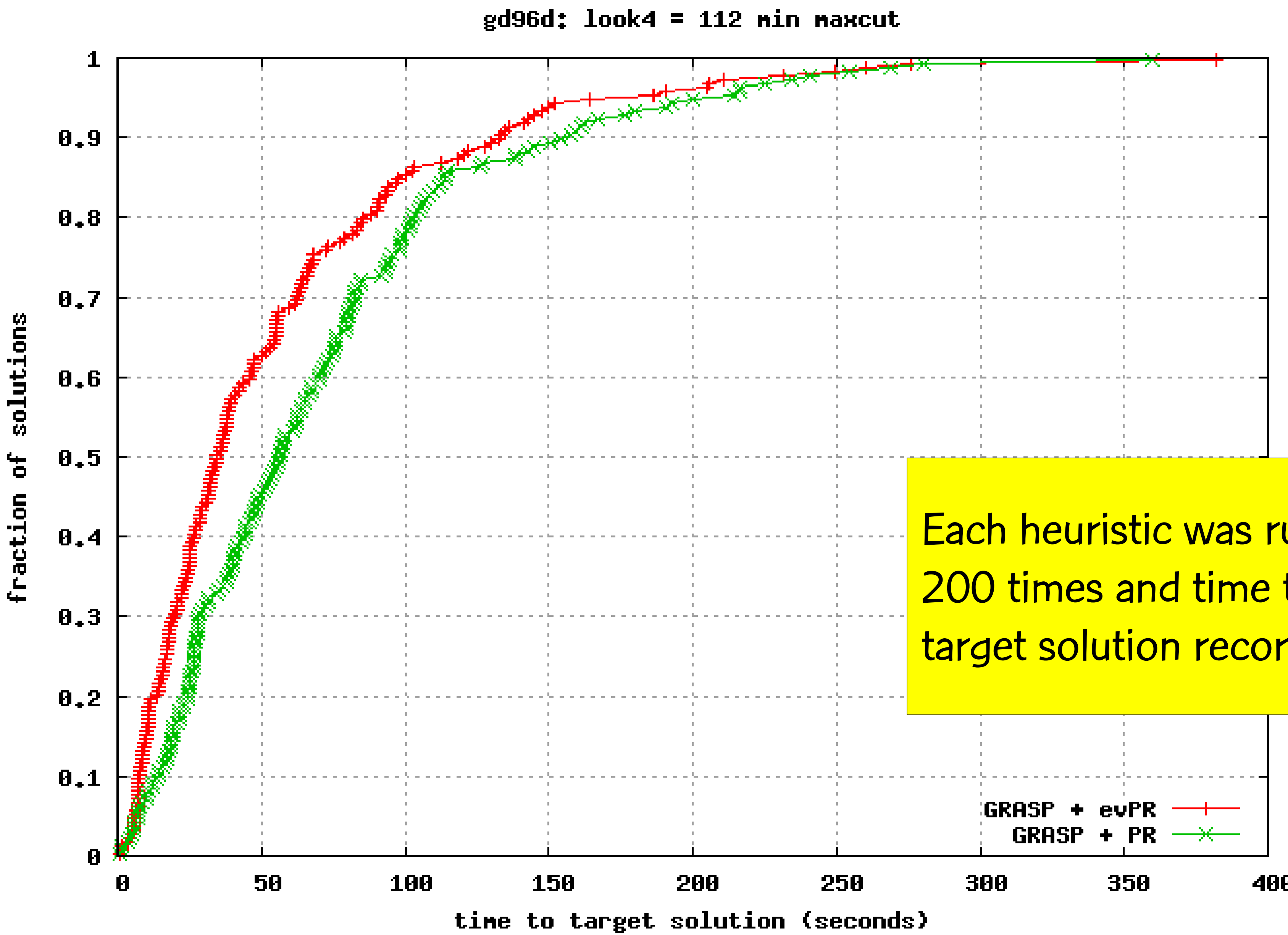
Weights uniformly distributed in interval [1,100]: min sum cuts



Each heuristic was run 200 times and time to target solution recorded.



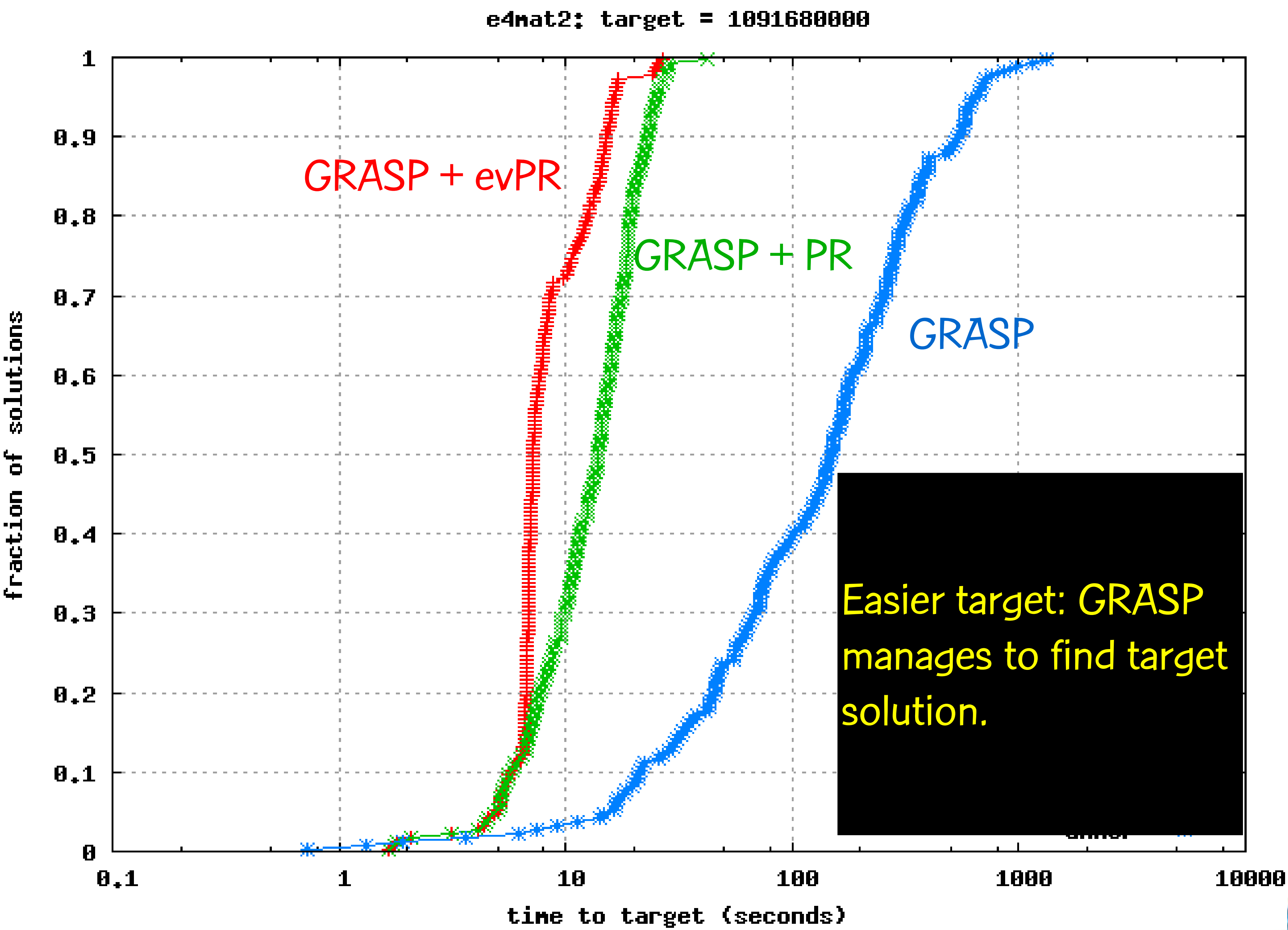
Each heuristic was run 200 times and time to target solution recorded.

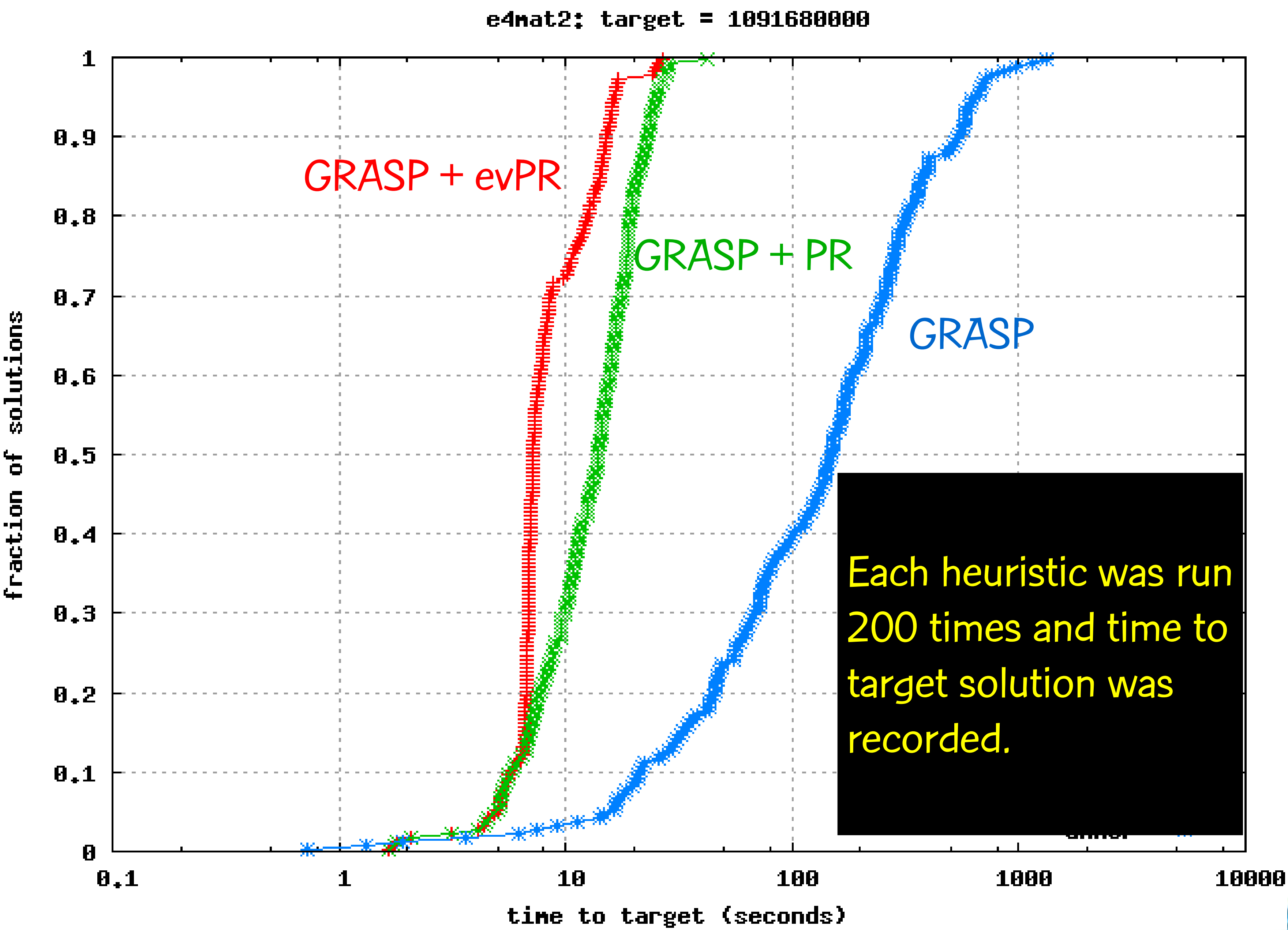


Each heuristic was run 200 times and time to target solution recorded.

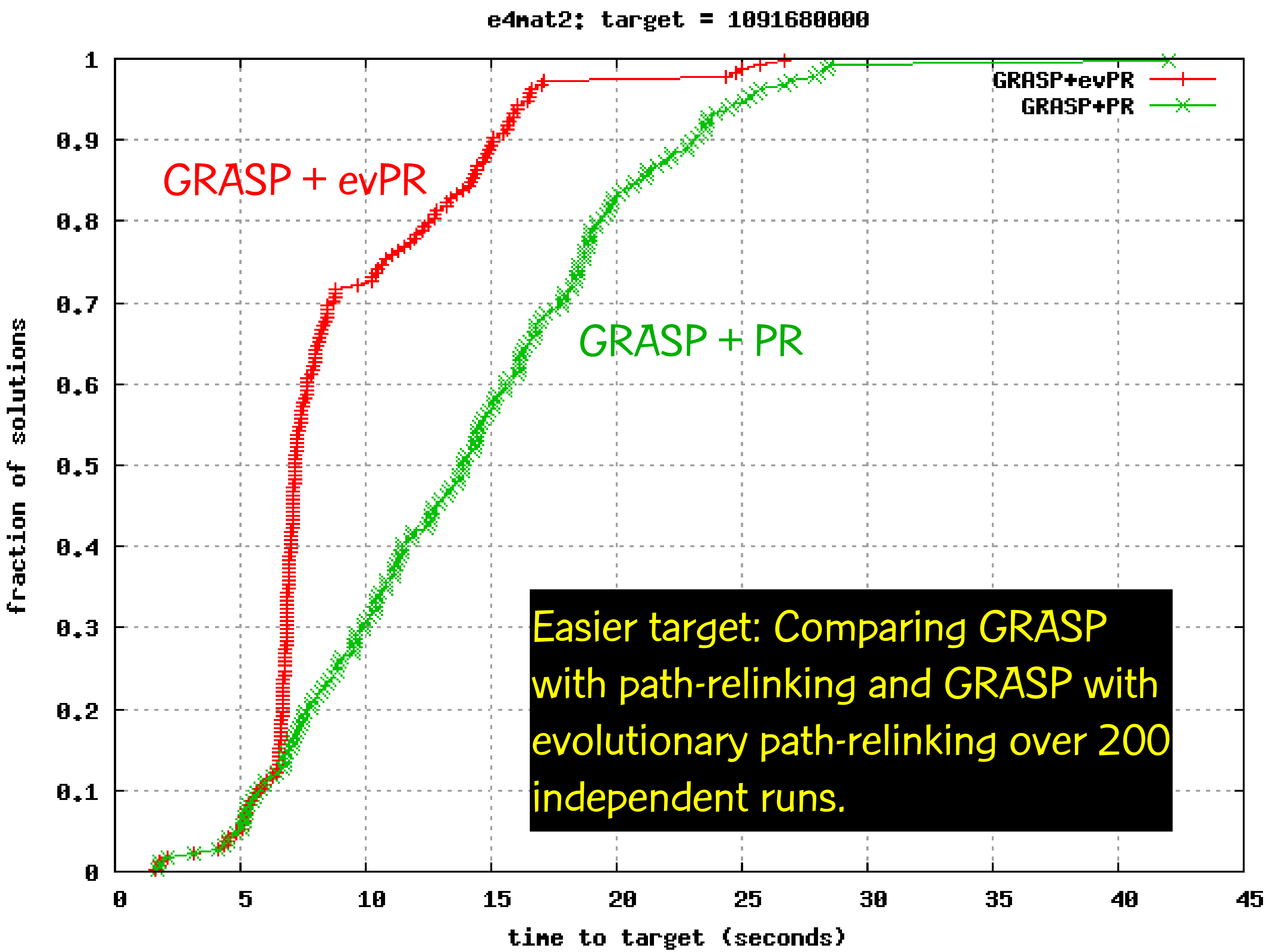
A real-world migration example

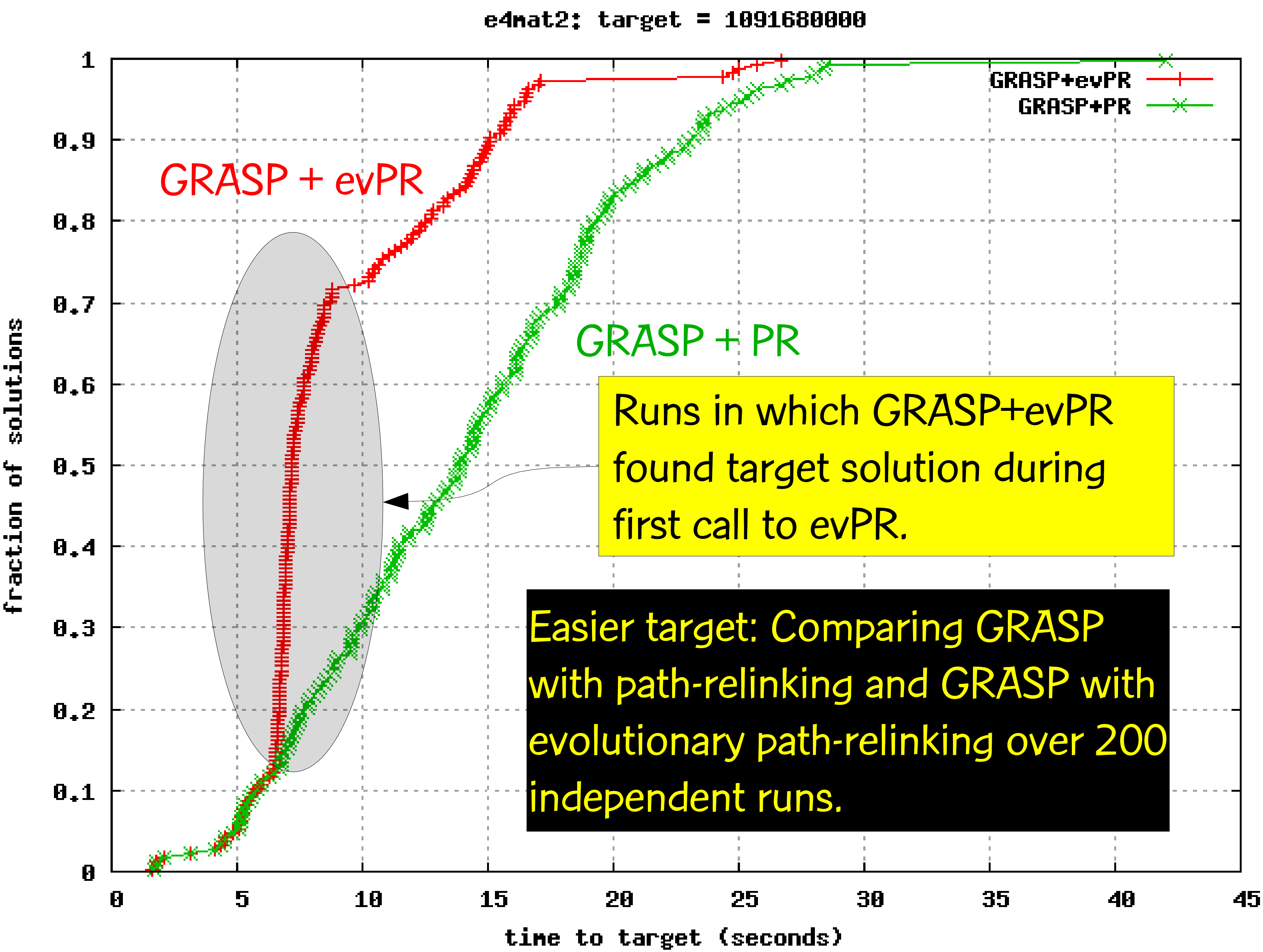
- Old network has 140 switches (nodes) and 9730 trunks (links): 100% edge density
- Traffic between switches is known.
- One switch is "deloaded" at each time period.
  - All traffic into (out of) deloaded switch is moved to new network.
  - New trunks may have to be temporarily deployed to handle the traffic between the old and new networks.


e4mat2: target = 1091680000
fraction of solutions
time to target (seconds)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.1
1
10
100
1000
10000
GRASP + evPR
GRASP + PR
GRASP
Easier target: GRASP manages to find target solution.

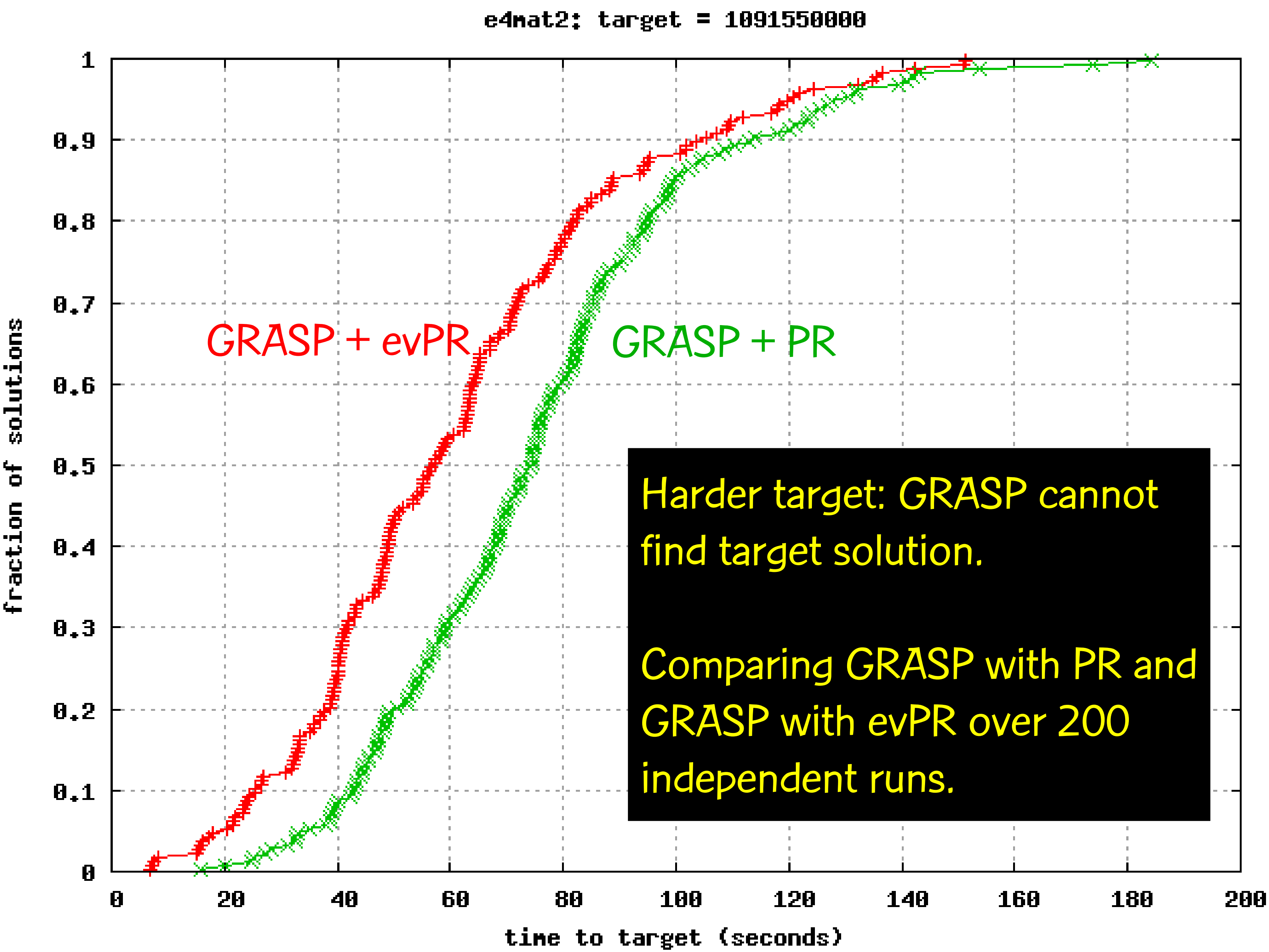


Each heuristic was run 200 times and time to target solution was recorded.


e4mat2: target = 1091680000
fraction of solutions
time to target (seconds)
GRASP+evPR
GRASP+PR
GRASP + evPR
GRASP + PR
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25
30
35
40
45
Easier target: Comparing GRASP with path-relinking and GRASP with evolutionary path-relinking over 200 independent runs.



at&t Your world. Delivered.


e4mat2: target = 1091550000
fraction of solutions
time to target (seconds)
GRASP + evPR
GRASP + PR
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
20
40
60
80
100
120
140
160
180
200
Harder target: GRASP cannot find target solution.
Comparing GRASP with PR and GRASP with evPR over 200 independent runs.

Concluding remarks

- We introduce GRASP with evolutionary path-relinking, an enhancement to GRASP with path-relinking.
- We propose an implementation that uses truncated mixed greedy randomized adaptive path-relinking with elite set aging.
- Computational results show that GRASP with evolutionary path-relinking can outperform GRASP with path-relinking.

Coauthor

Diogo Andrade
Rutcor, Rutgers University

The End

These slides and all papers cited in this talk can be downloaded from my homepage: http://www.research.att.com/~mgcr