A genetic algorithm with optimized crossover for routing on the Internet

Talk given on November 17, 2002 at INFORMS Annual Meeting in San José, California.

Mauricio G. C. Resende

Algorithms & Optimization Research Dept.
Internet & Network Systems Research Center
AT&T Labs Research
Florham Park - New Jersey - USA

Joint work with:

M. Ericsson
P. M. Pardalos
L. S. Buriol
C. C. Ribeiro
M. Thorup

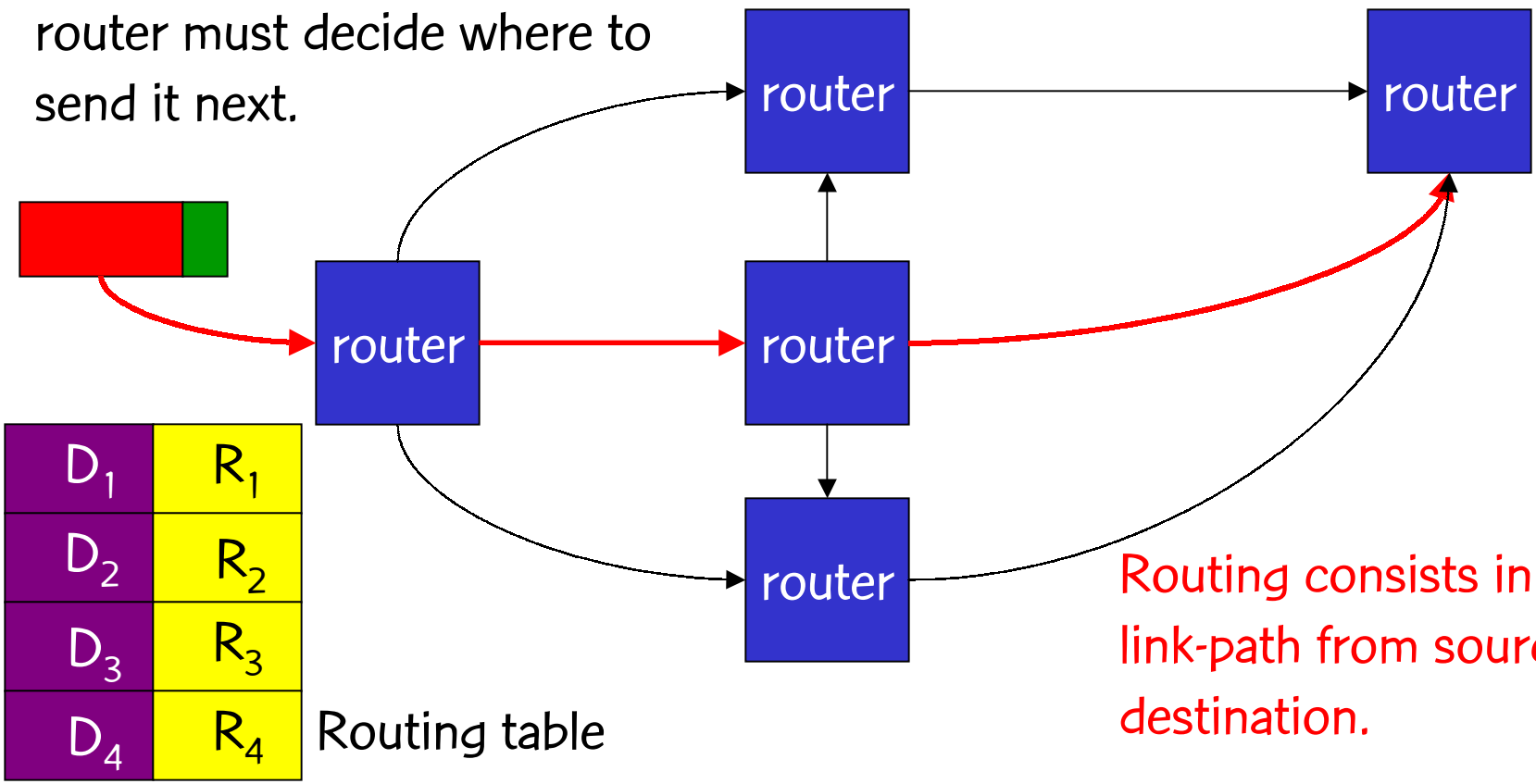
Internet traffic engineering

- Objective: make more efficient use of existing network resources.
- Routing of traffic can have a major impact on efficiency of network resource utilization.

Packet routing

When packet arrives at router, router must decide where to send it next.

Packet's final destination.



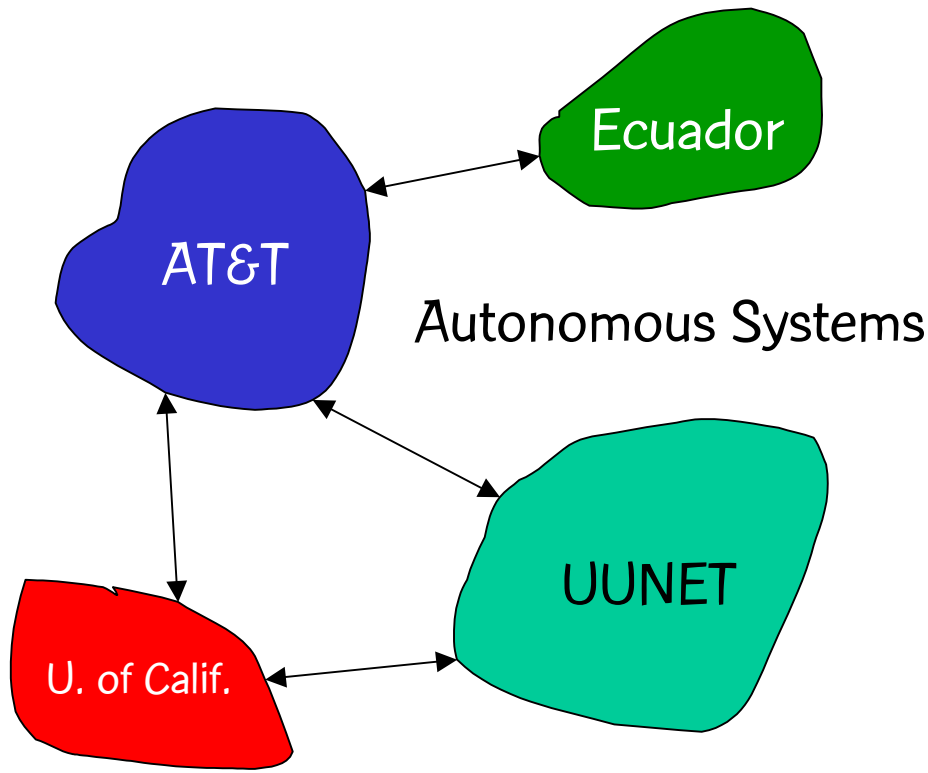
D_1	R_1
D_2	R_2
D_3	R_3
D_4	R_4

Routing table

Routing consists in finding a link-path from source to destination.

OSPF (Open Shortest Path First)

- OSPF is a commonly used intra-domain routing protocol (IGP).
- Routers exchange routing information with all other routers in the autonomous system (AS).
  - Complete network topology knowledge is available to all routers, i.e. state of all routers and links in the AS.



OSPF routing

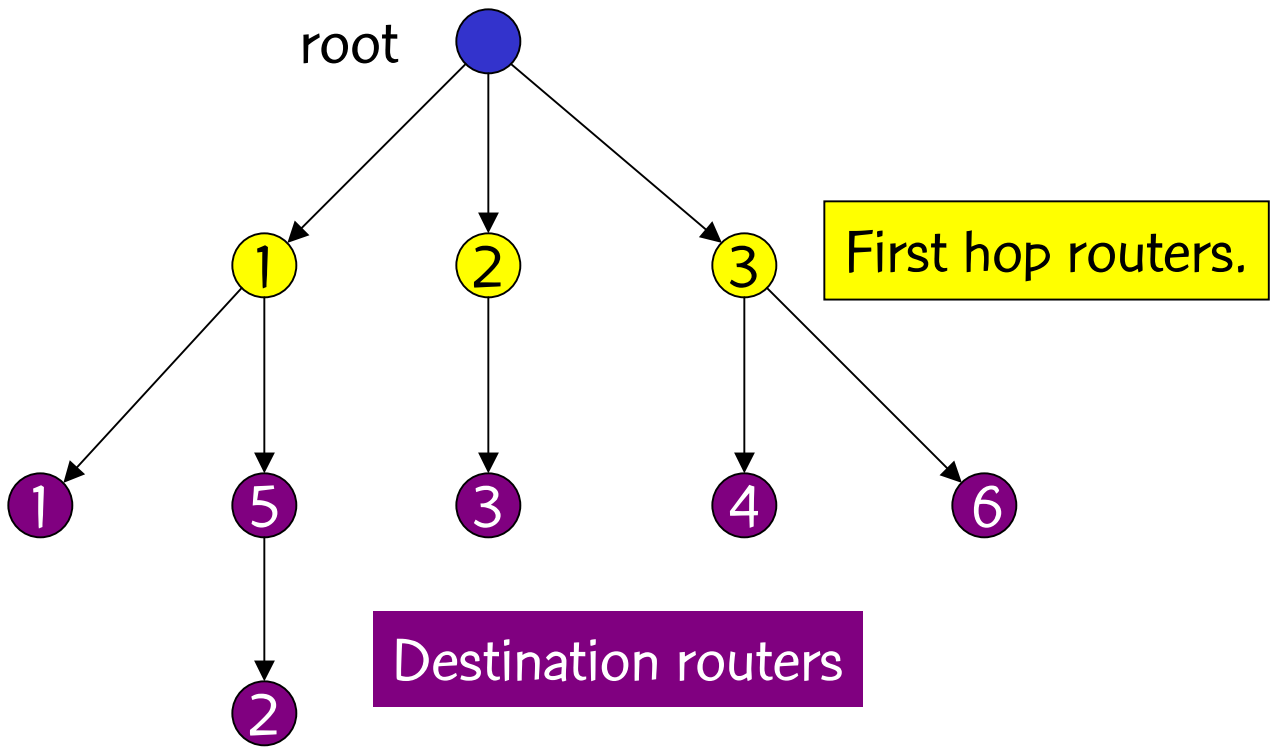
- Assign an integer weight $\in [1, w_{max}]$ to each link in AS. In general, $w_{max} = 65535 = 2^{16} - 1$.
- Each router computes tree of shortest weight paths to all other routers in the AS, with itself as the root, using Dijkstra's algorithm.

OSPF routing

Routing table

D_1	R_1
D_2	R_1
D_3	R_2
D_4	R_3
D_5	R_1
D_6	R_3

Routing table is filled with first hop routers for each possible destination.

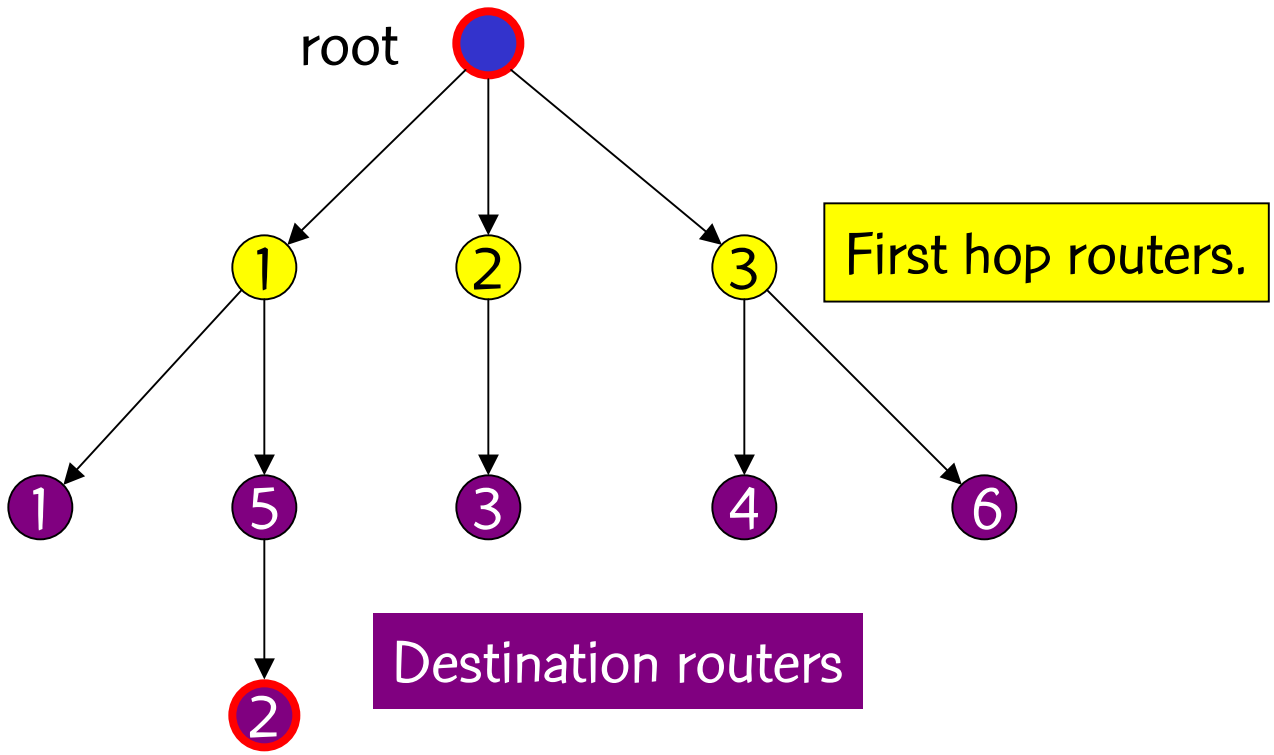


OSPF routing

Routing table is filled
with first hop routers
for each possible destination.

Routing table

D_1	R_1
D_2	R_1
D_3	R_2
D_4	R_3
D_5	R_1
D_6	R_3

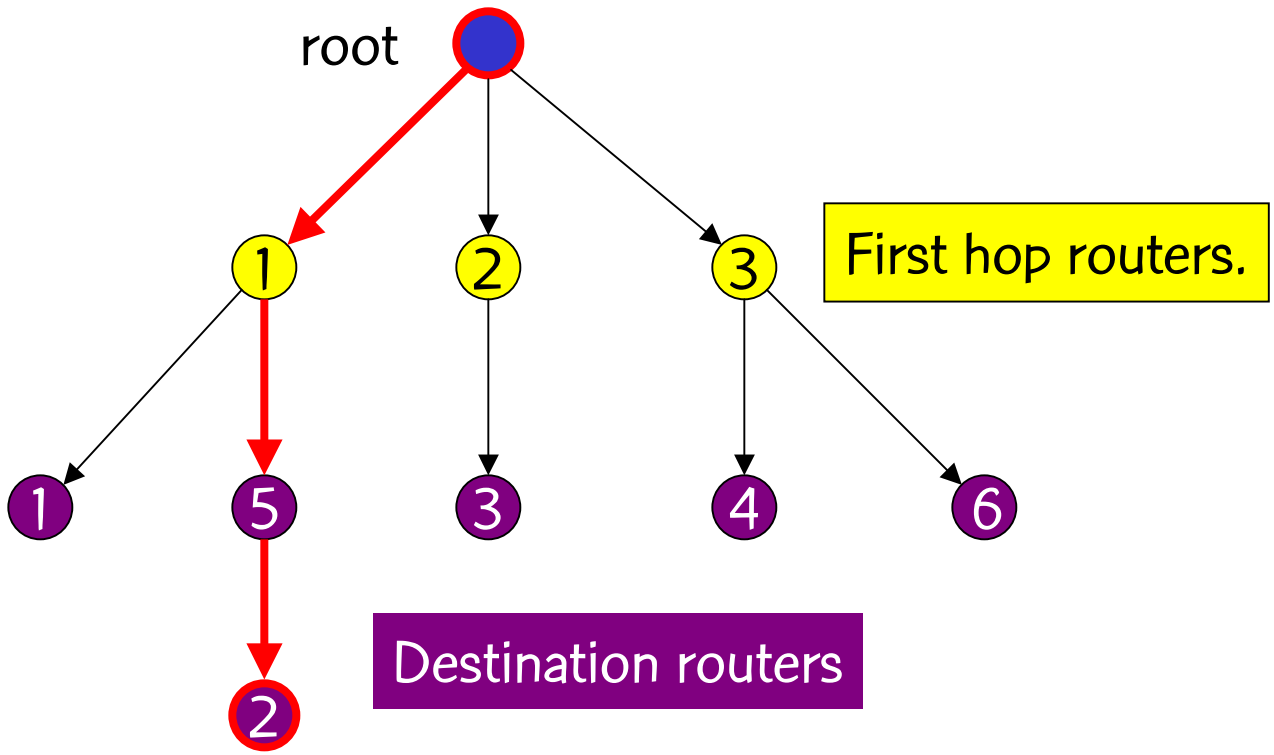


OSPF routing

Routing table is filled
with first hop routers
for each possible destination.

Routing table

D_1	R_1
D_2	R_1
D_3	R_2
D_4	R_3
D_5	R_1
D_6	R_3

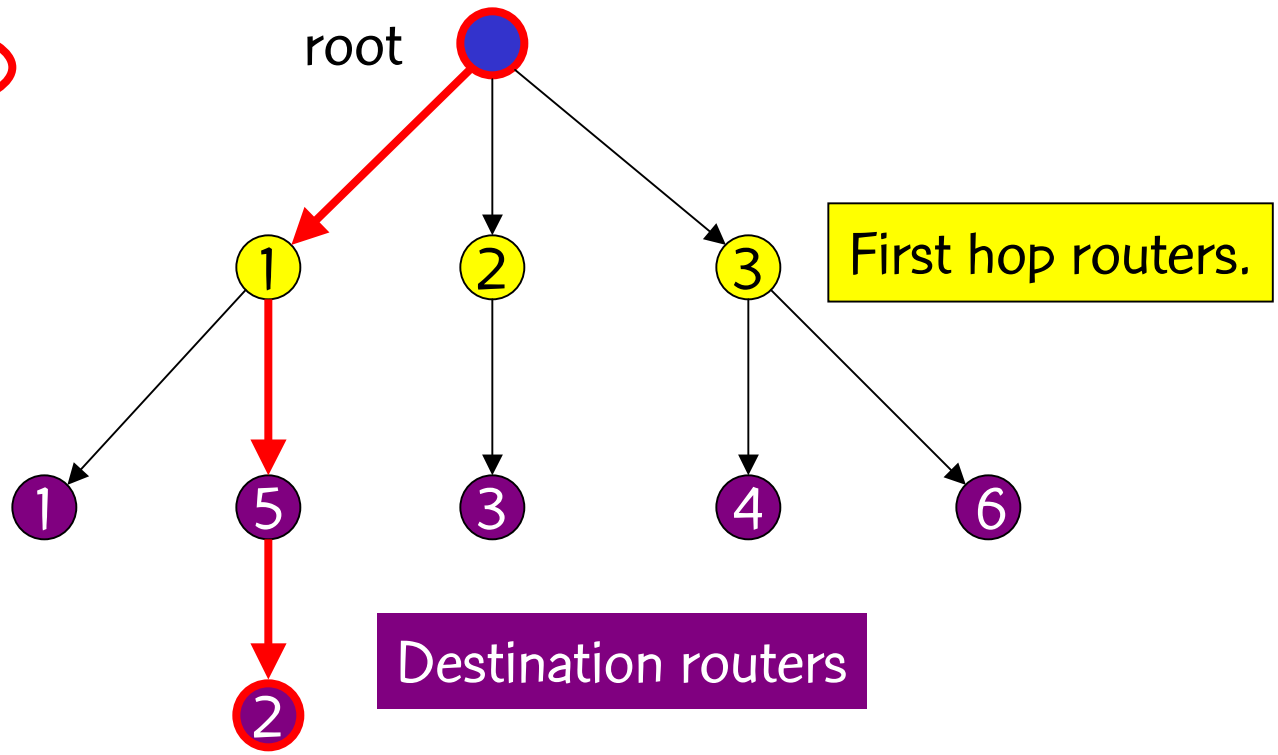


OSPF routing

Routing table is filled
with first hop routers
for each possible destination.

Routing table

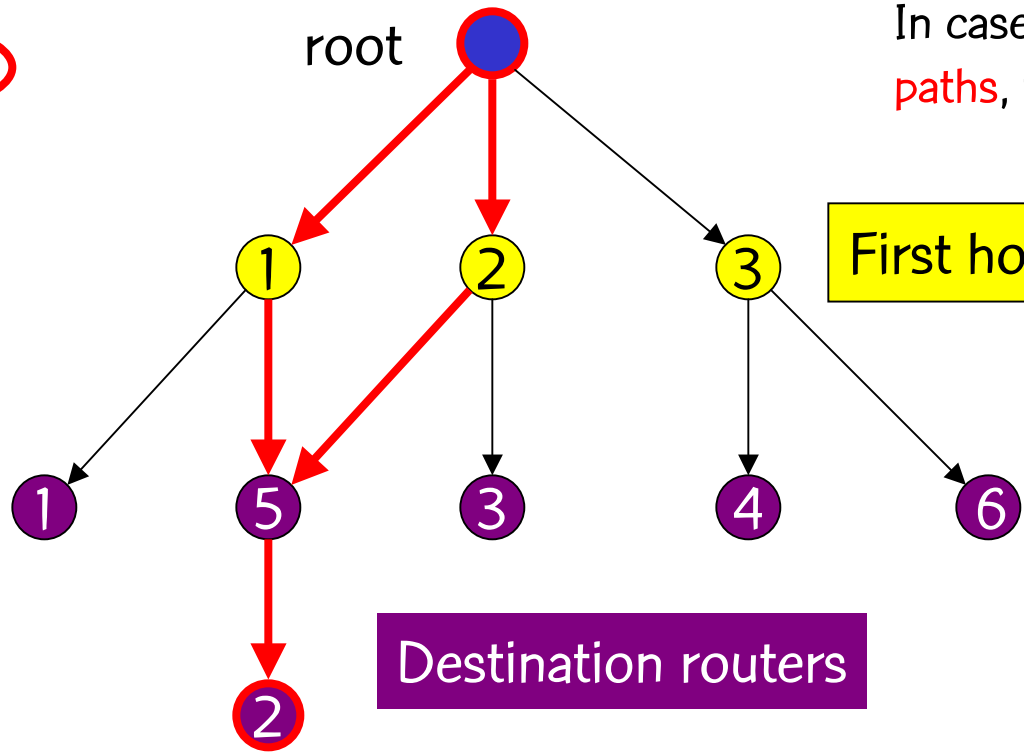
D_1	R_1
D_2	R_1
D_3	R_2
D_4	R_3
D_5	R_1
D_6	R_3



OSPF routing

Routing table

D_1	R_1
D_2	R_1, R_2
D_3	R_2
D_4	R_3
D_5	R_1
D_6	R_3



Routing table is filled with first hop routers for each possible destination. In case of multiple shortest paths, flow is evenly split.

OSPF weight setting

- OSPF weights are assigned by network operator.
  - CISCO assigns, by default, a weight proportional to the inverse of the link bandwidth (Inv Cap).
  - If all weights are unit, the weight of a path is the number of hops in the path.
- We propose a hybrid genetic algorithm to find good OSPF weights.
  - Memetic algorithm
  - Genetic algorithm with optimized crossover

Minimization of congestion

- Consider the directed capacitated network $G = (N,A,c)$, where N are routers, A are links, and c_a is the capacity of link $a \in A$.
- We use the measure of Fortz & Thorup (2000) to compute congestion:

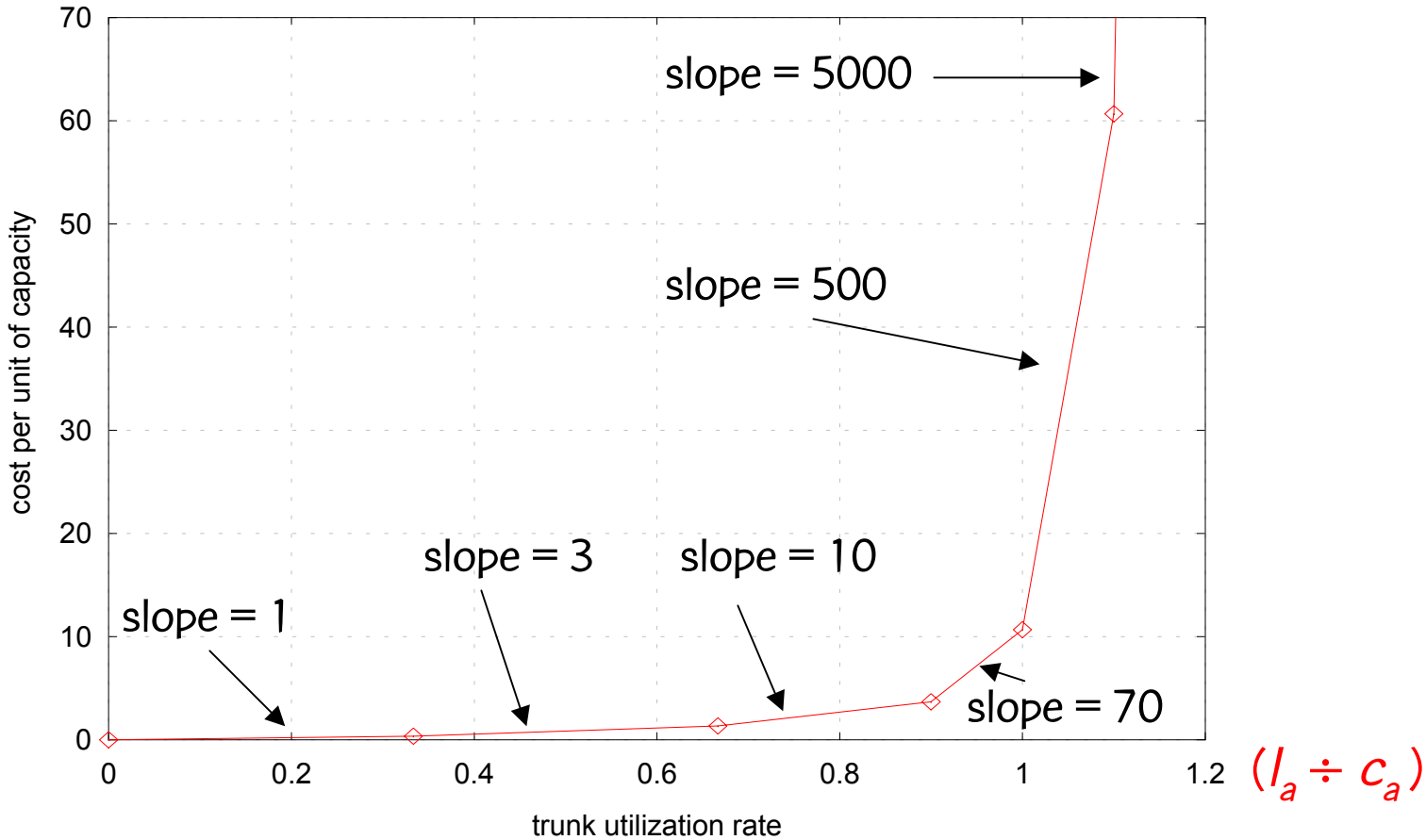
$$\Phi = \Phi_1(l_1) + \Phi_2(l_2) + \ldots + \Phi_{|A|}(l_{|A|})$$

where l_a is the load on link $a \in A$,

$\Phi_a(l_a)$ is piecewise linear and convex,

$\Phi_a(0) = 0$, for all $a \in A$.

Piecewise linear and convex $\Phi_a(I_a)$ link congestion measure



OSPF weight setting problem

- Given a directed network $G = (N, A)$ with link capacities $c_a \in A$ and demand matrix $D = (d_{s,t})$ specifying a demand to be sent from node s to node t:
  - Assign weights $w_a \in [1, w_{max}]$ to each link $a \in A$, such that the objective function Φ is minimized when demand is routed according to the OSPF protocol.

Cost normalization

Consider the demand matrix D = $(d_{s,t})$ and let $h_{s,t}$ be the min hop count between s and t.

Normalize Φ by $\Phi_{uncap} = \sum_{(s,t)\in N\times N} d_{s,t} h_{s,t}$

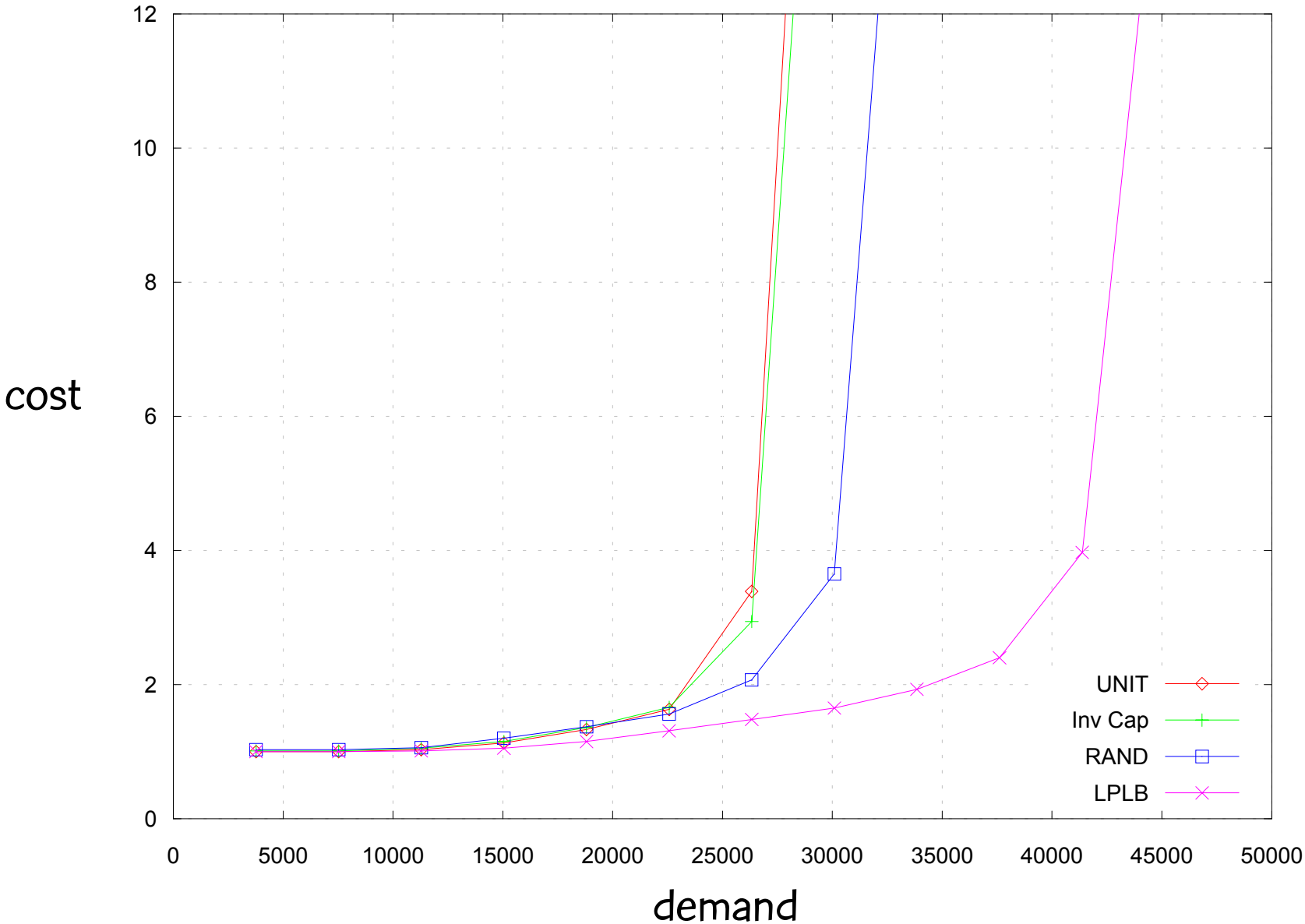
Total load if all traffic goes along unit weight shortest paths.

Normalized cost: $\Phi^* = \Phi / \Phi_{uncap}$

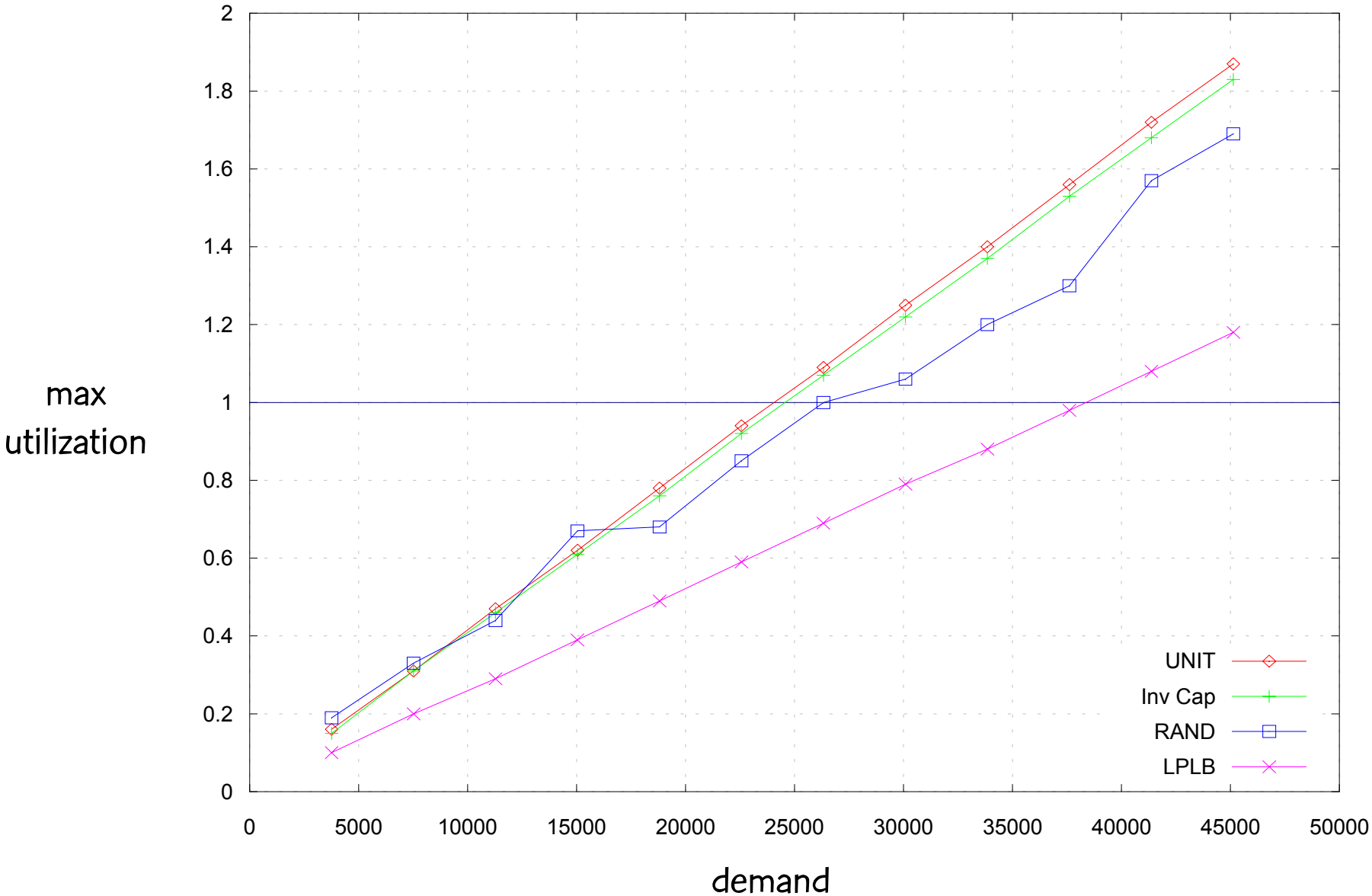
Normalized cost $\Phi^* = \Phi / \Phi_{uncap}$

- Fortz & Thorup (2000) show that:
- $1 \le \Phi^*_{opt} \le \Phi^*_{optOSPF} \le \Phi^*_{unitOSPF} < 5000$
- If $\Phi^* = 1$, then all loads are below 1/3 of capacity.
- If a packet follows a shortest path and if all arcs are exactly full, then $\Phi^* = 10\frac{2}{3}$
- Routing congests the network if $\Phi^* \ge 10\frac{2}{3}$

AT&T Worldnet backbone network (90 routers, 274 links)



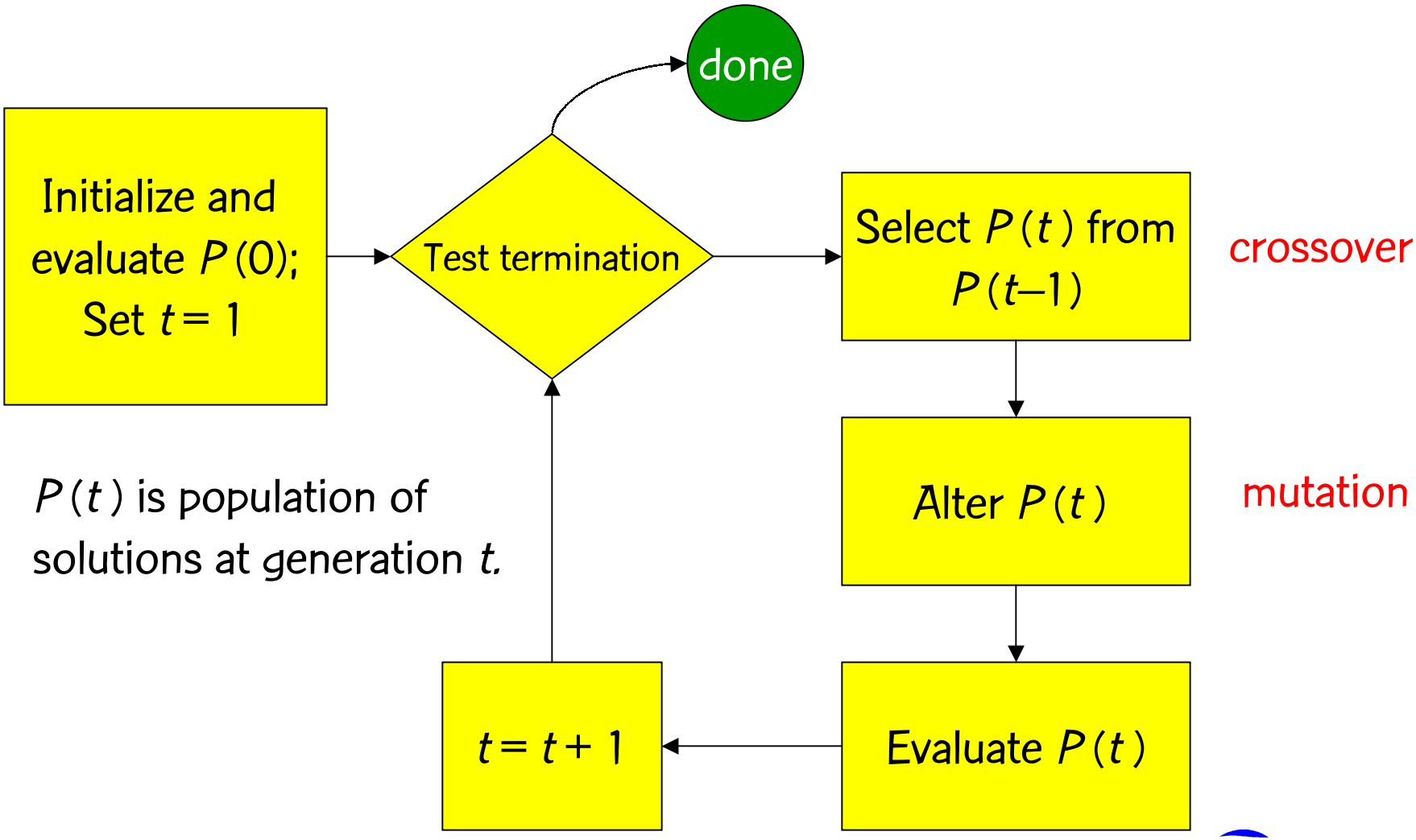
AT&T Worldnet backbone network (90 routers, 274 links)



Previous work

- Avoid multiple shortest paths (small networks):
  - Lin & Wang (1993): use Lagrangian relaxation and consider networks with up to 26 nodes.
  - Rodrigues & Ramakrishnan (1994): use local search with single descent and work on small networks with at most 16 nodes and 18 links.
  - Bley et at. (1998): use local search with single descent and consider small networks with up to 13 links.
- Allow multiple shortest paths (realistically sized networks):
  - Fortz & Thorup (2000): use tabu search type local search on large networks with up to 100 nodes and 503 links.
  - Ericcson, R., & Pardalos (2002): genetic algorithm

Genetic algorithms



Solution encoding

- A population consists of $nPop = 50$ integer weight arrays: $w = (w_1, w_2, \ldots, w_{|A|})$,

  where $w_a \in [1, w_{max} = 20]$
- All possible weight arrays correspond to feasible solutions.

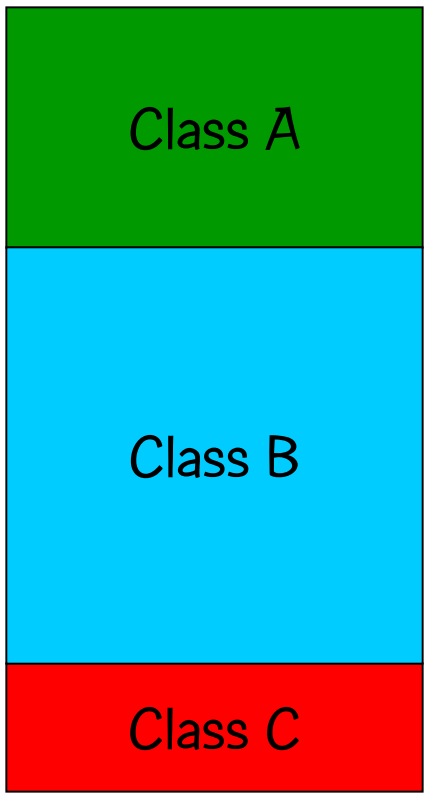
Initial population

- *nPop* solutions, with each weight randomly generated, uniformly in the interval $[1, w_{max}/3]$.

Solution evaluation

- For each demand pair (s,t), route using OSPF, computing demand pair loads $l_a^{s,t}$ on each link $a \in A$.
- Add up demand pair loads on each link $a \in A$, yielding total load l_a on link.
- Compute link congestion cost $\Phi_a(l_a)$ for each link $a \in A$.
- Add up costs: $\Phi = \Phi_1(l_1) + \Phi_2(l_2) + \ldots + \Phi_{|A|}(l_{|A|})$

Population partitioning



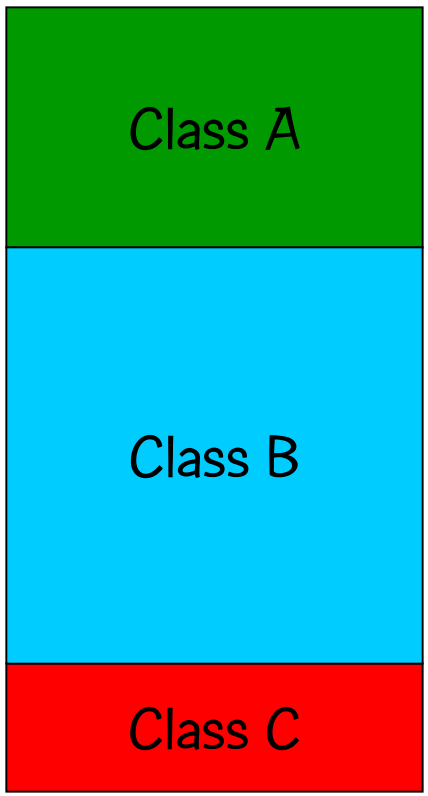
25% most fit

5% least fit

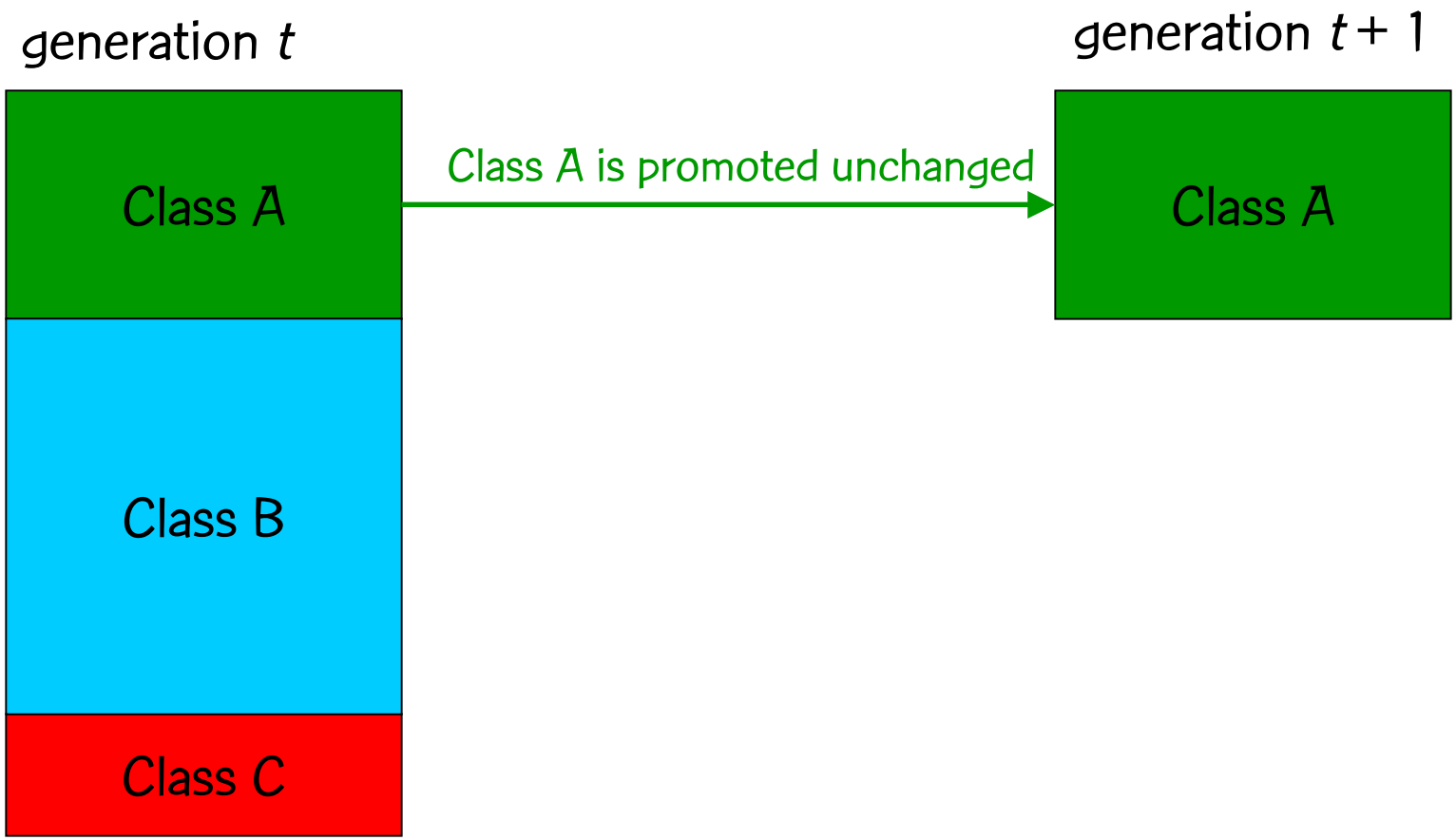
Population is sorted according to solution value Φ and solutions are classified into three categories.

Population dynamics

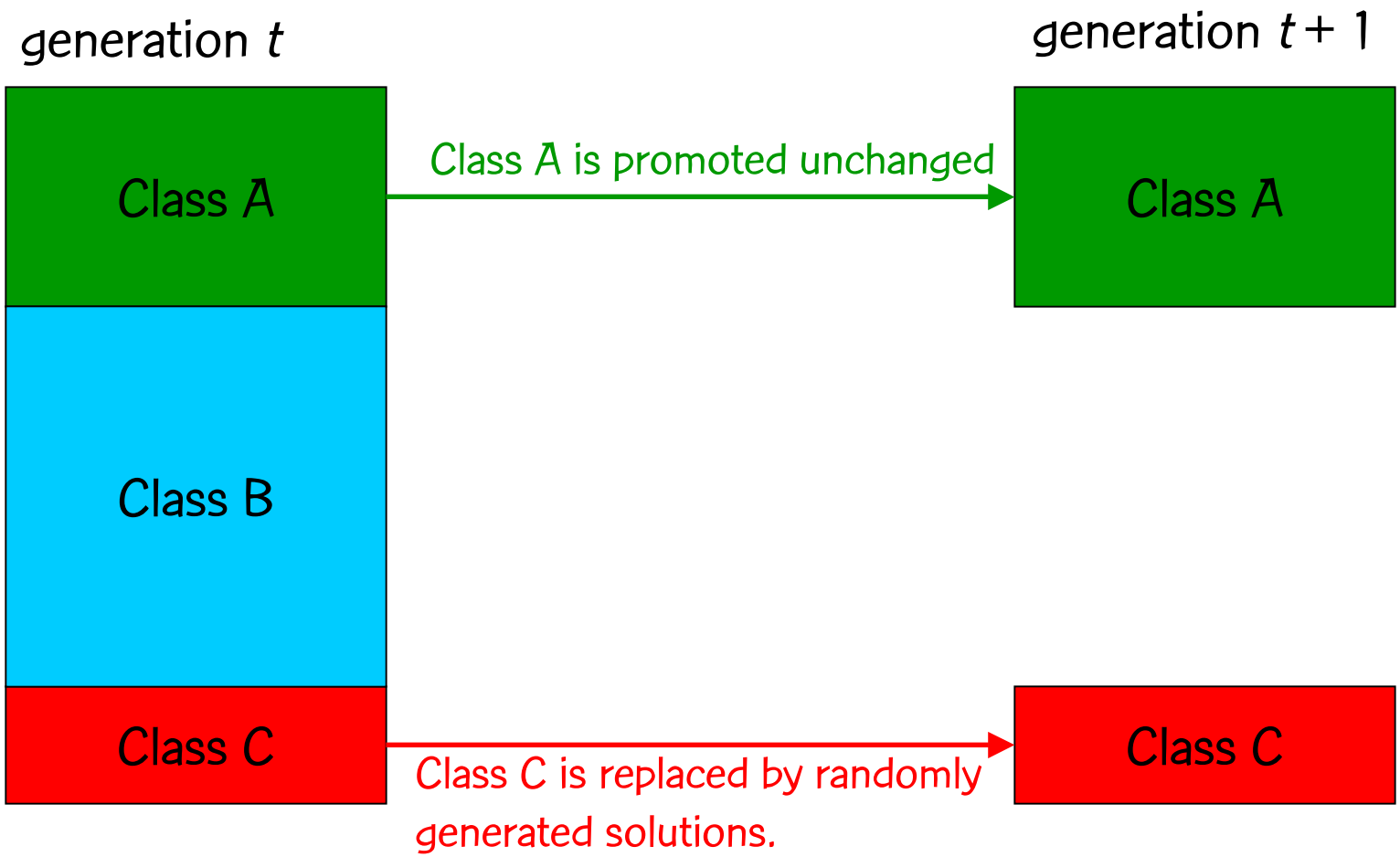
generation *t*



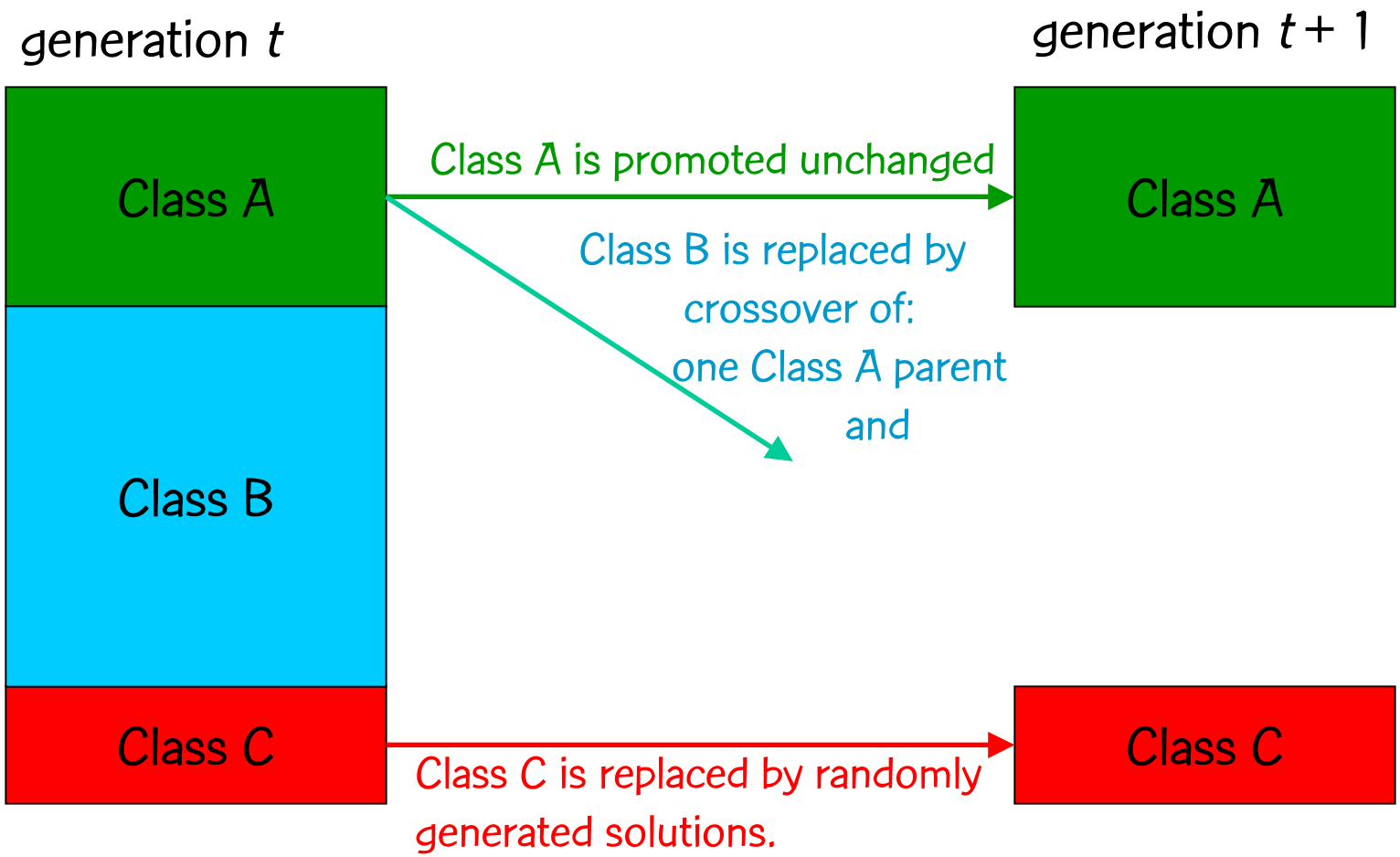
Population dynamics



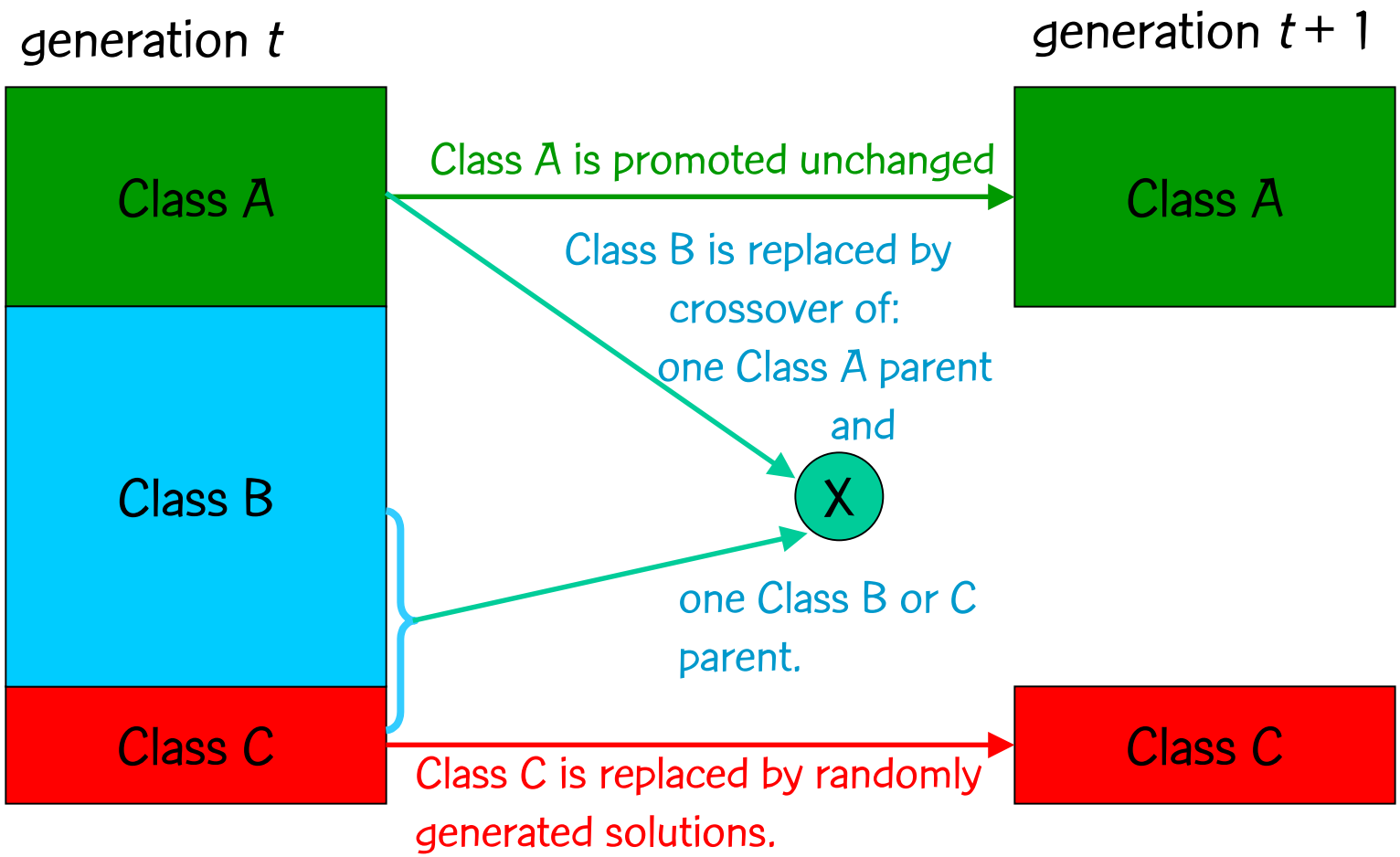
Population dynamics



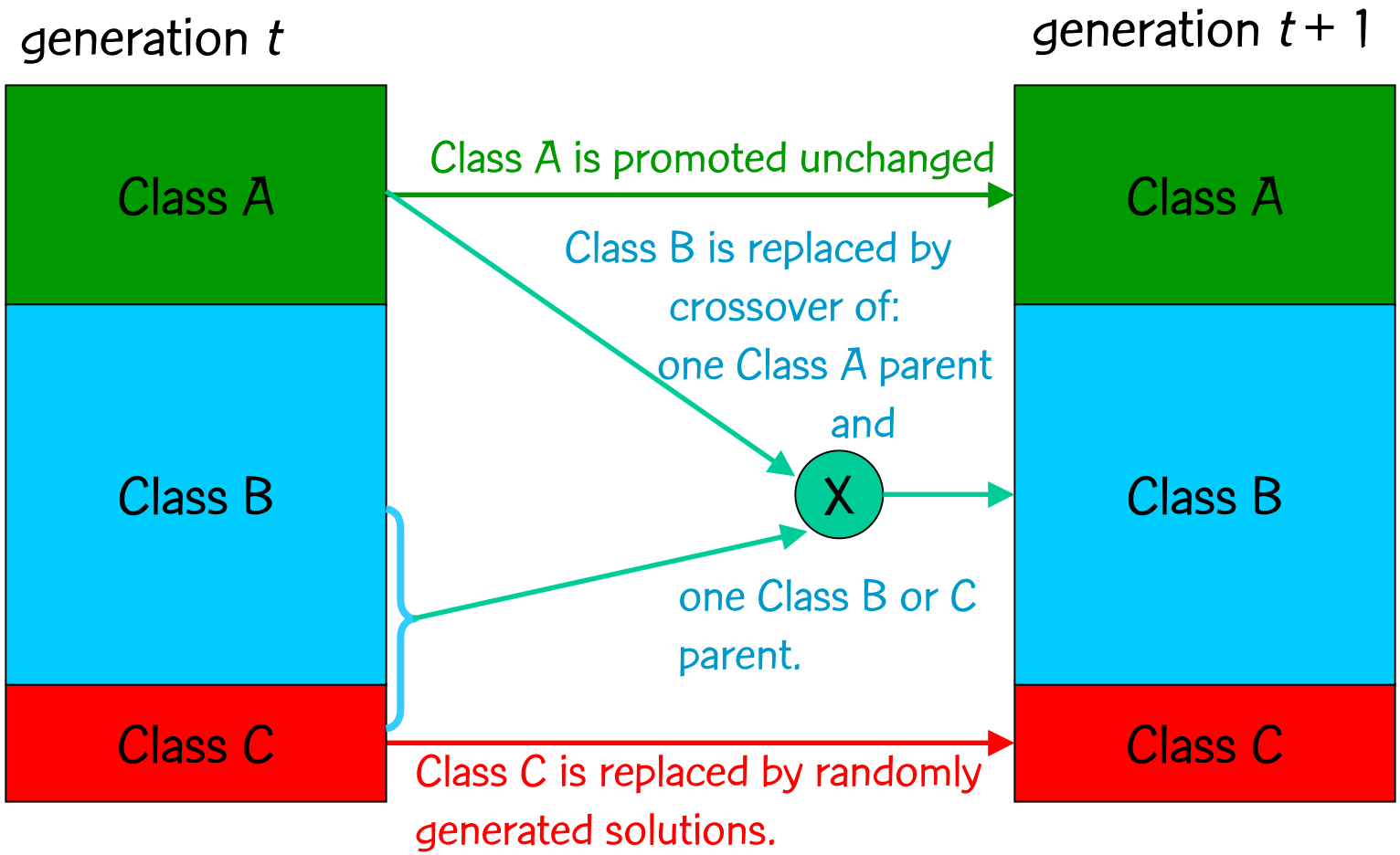
Population dynamics



Population dynamics



Population dynamics



Parent selection

- Parents are chosen at random:
  - one parent from Class A (elite).
  - one parent from Class B or C (non-elite).
- Reselection is allowed, i.e. parents can breed more than once per generation.
- Better individuals are more likely to reproduce.

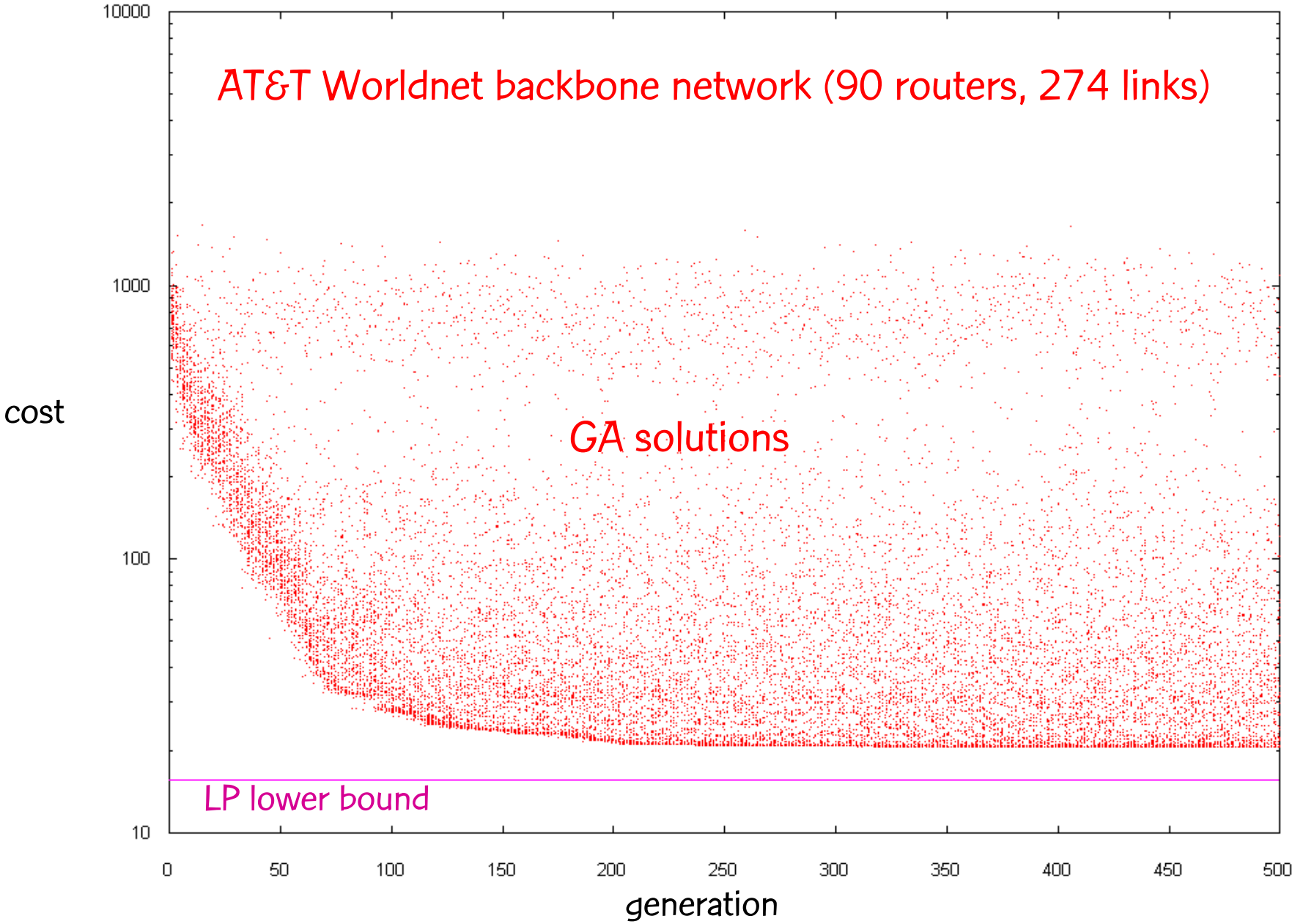
Crossover with random keys (Bean, 1994)

Crossover combines elite parent p_1 with non-elite parent p_2 to produce child c:

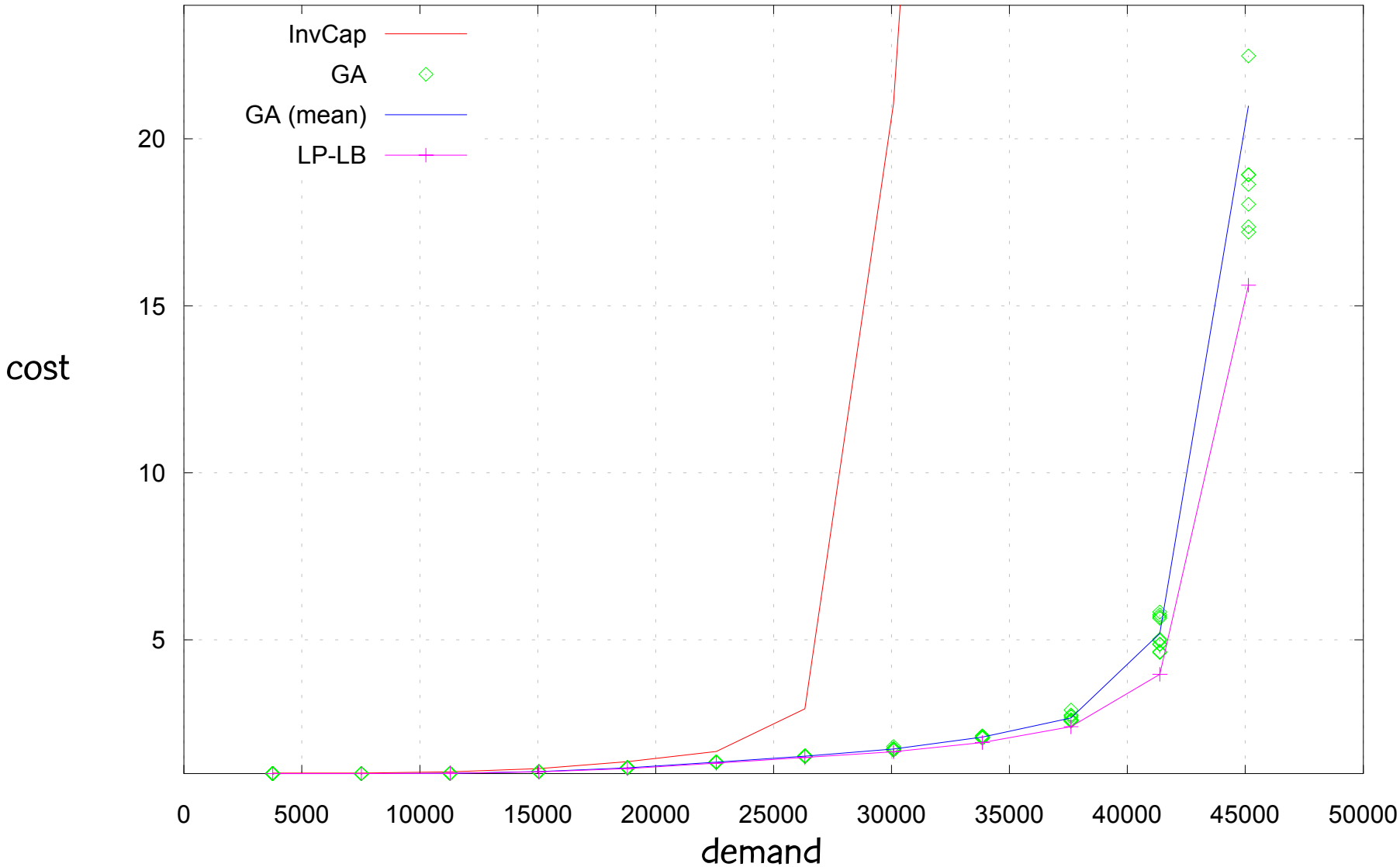
With small probability child has single gene mutation.

Child is more likely to inherit gene of elite parent.

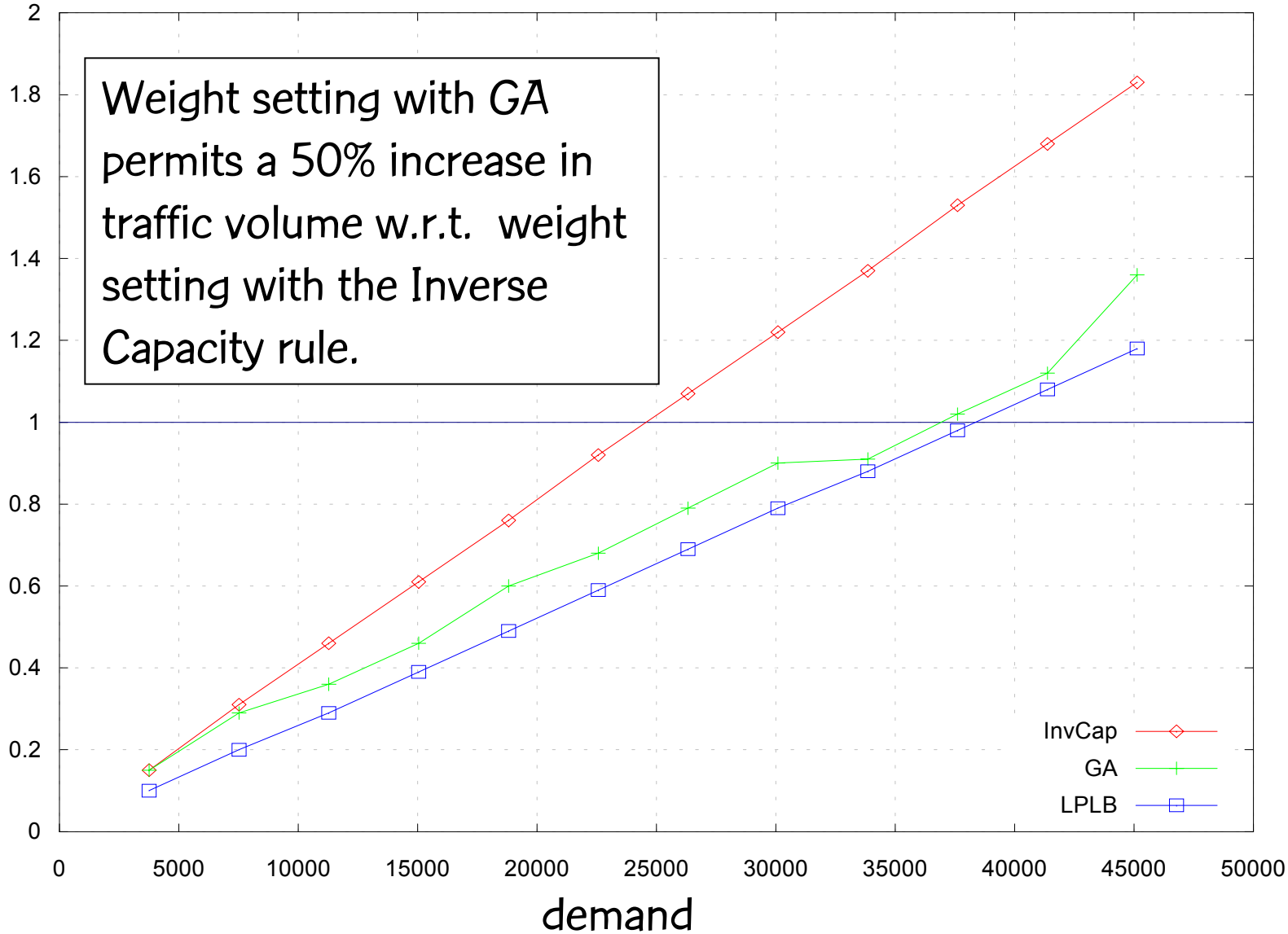
```
for all genes i = 1,2,..., |A| do
    if rrandom[0,1] < 0.01 then
        c[i] = irandom[1, w_max]
    else if rrandom[0,1] < 0.7 then
        c[i] = p1[i]
    else c[i] = p2[i]
end
```


AT&T Worldnet backbone network (90 routers, 274 links)
GA solutions
LP lower bound
cost
generation
10000
1000
100
10
0
50
100
150
200
250
300
350
400
450
500

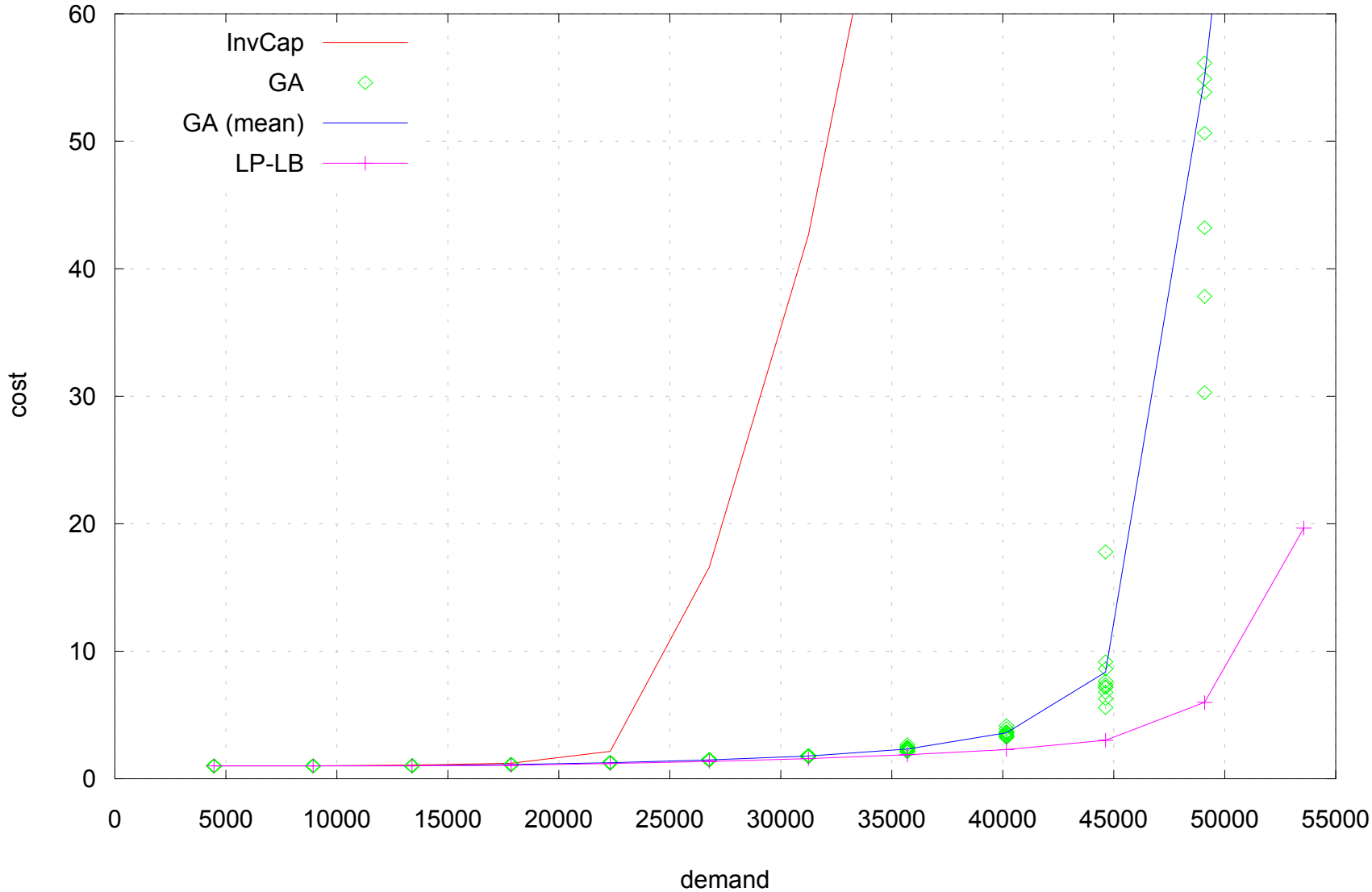
AT&T Worldnet backbone network (90 routers, 274 links)



AT&T Worldnet backbone network (90 routers, 274 links)



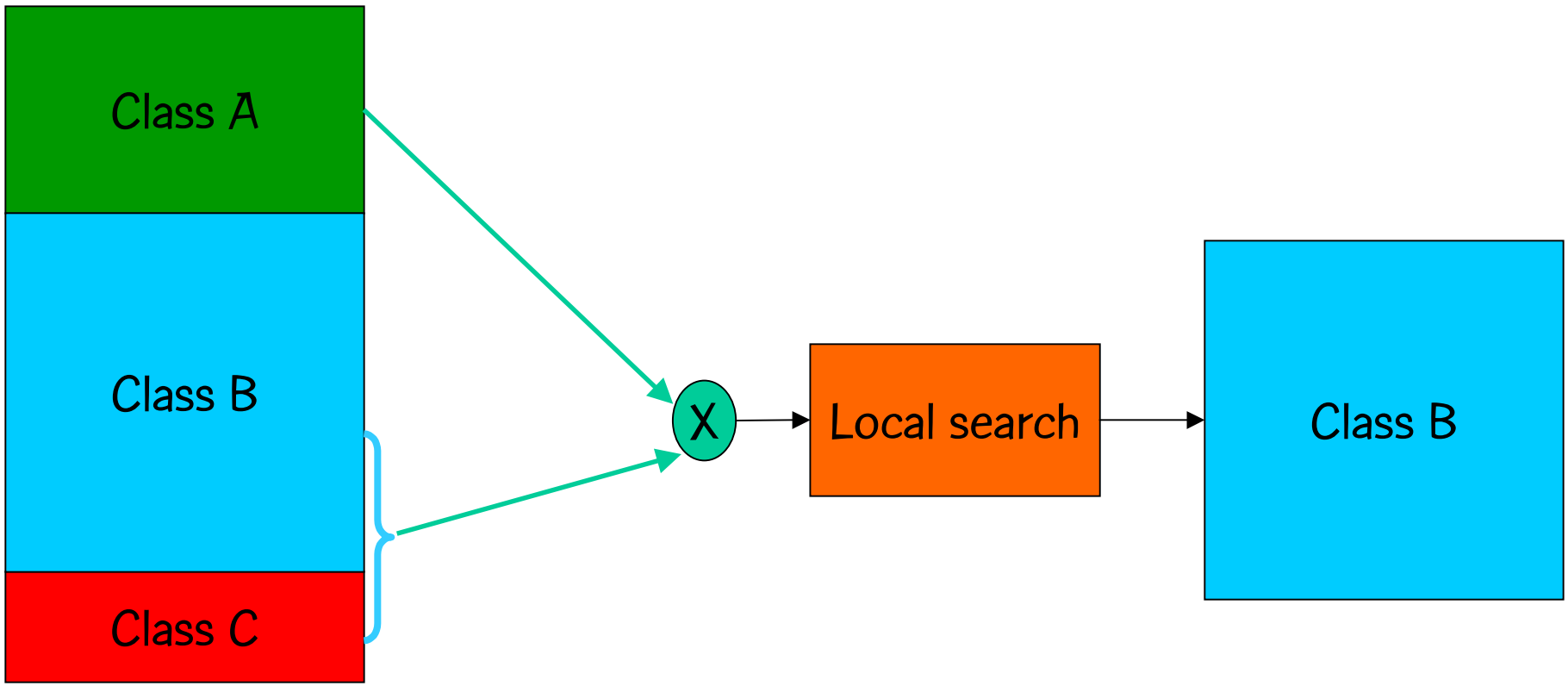
Rand50a: random graph with 50 nodes and 245 arcs.



1 hour run

AT&T

Optimized crossover = crossover + local search



Fast local search

- Let A^* be the set of five arcs $a \in A$ having largest Φ_a values.
- Scan arcs $a \in A^*$ from largest to smallest Φ_a:
  - Increase arc weight, one unit at a time, in the range $[w_a , w_a + \lceil (w_{max} - w_a)/4 \rceil]$
  - If total cost Φ is reduced, restart local search.

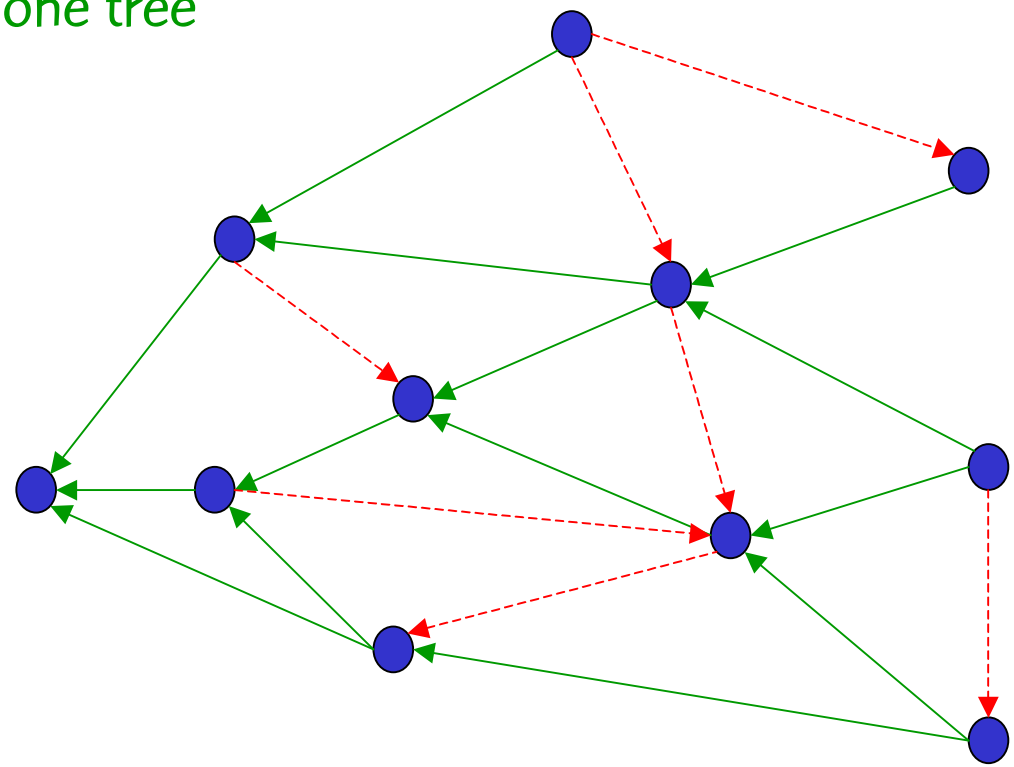
Dynamic shortest path

- In local search, when arc weight increases, shortest path trees:
  - may change completely (rarely do)
  - may remain unchanged (e.g. arc not in a tree)
  - may change partially
    - Few trees change
    - Small portion of tree changes

} Does not make sense to recompute trees from scratch.

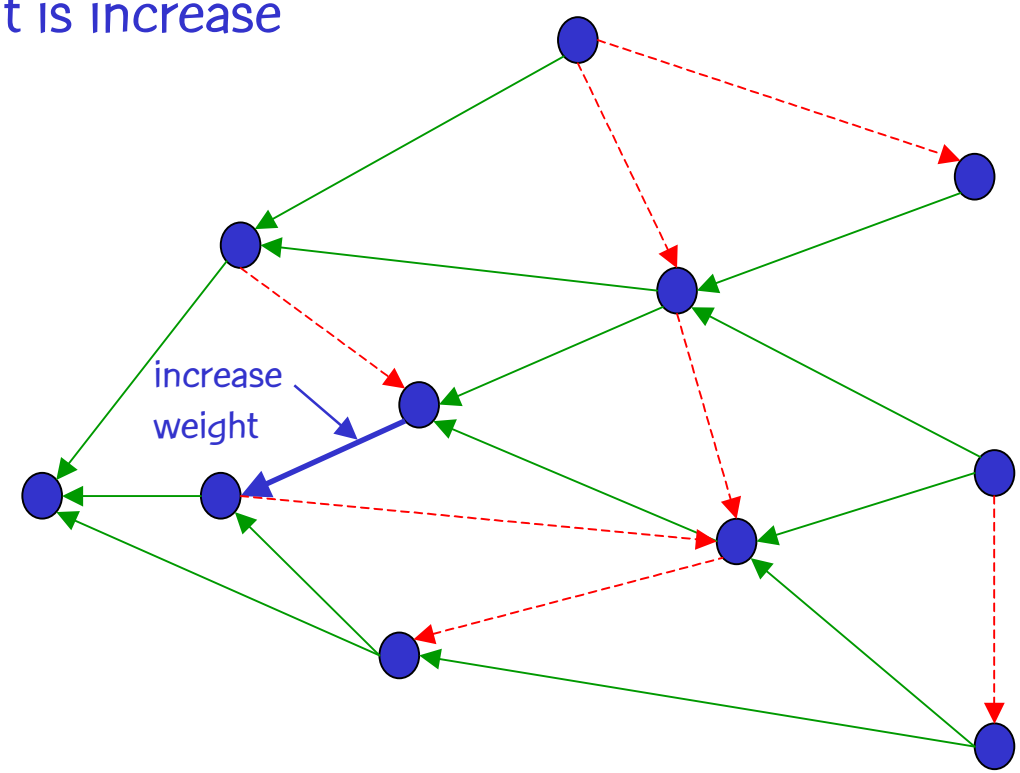
Dynamic shortest path

Consider one tree at a time.



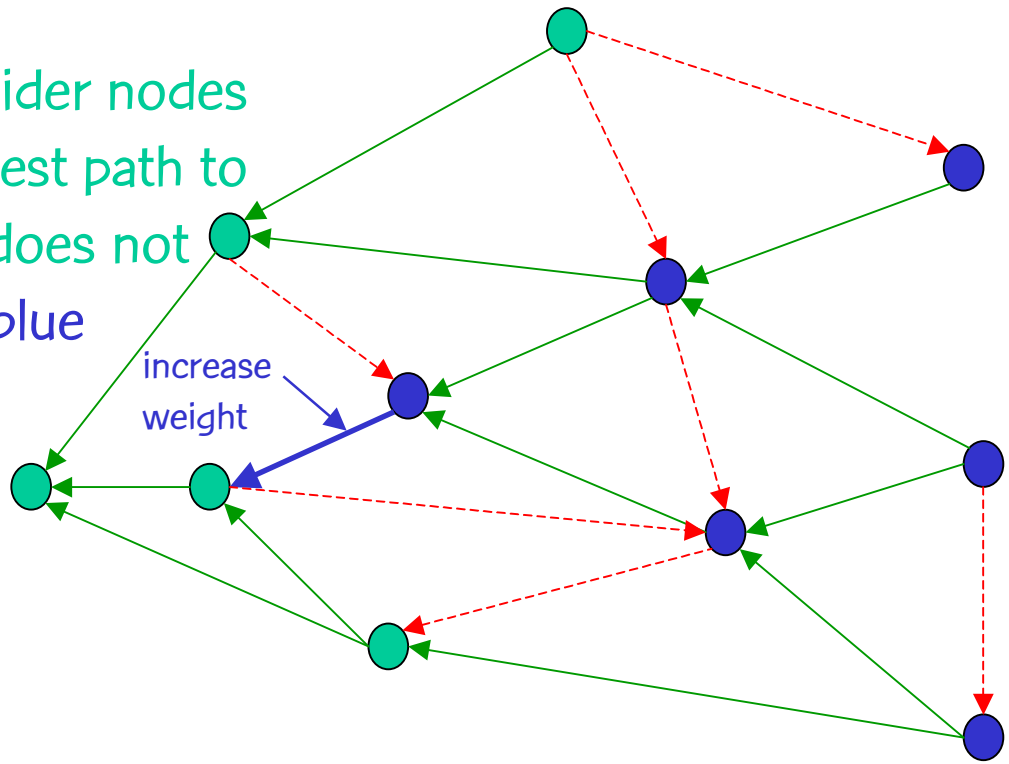
Dynamic shortest path

Arc weight is increase by 1.

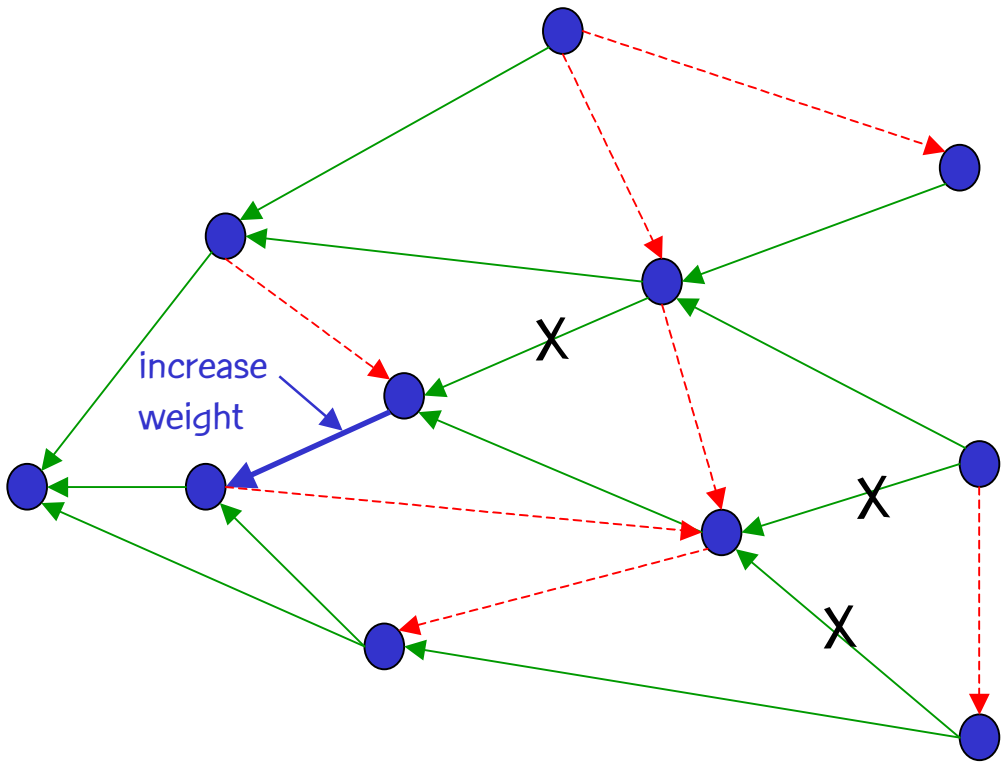


Dynamic shortest path

Do not consider nodes whose shortest path to destination does not go through blue arc.

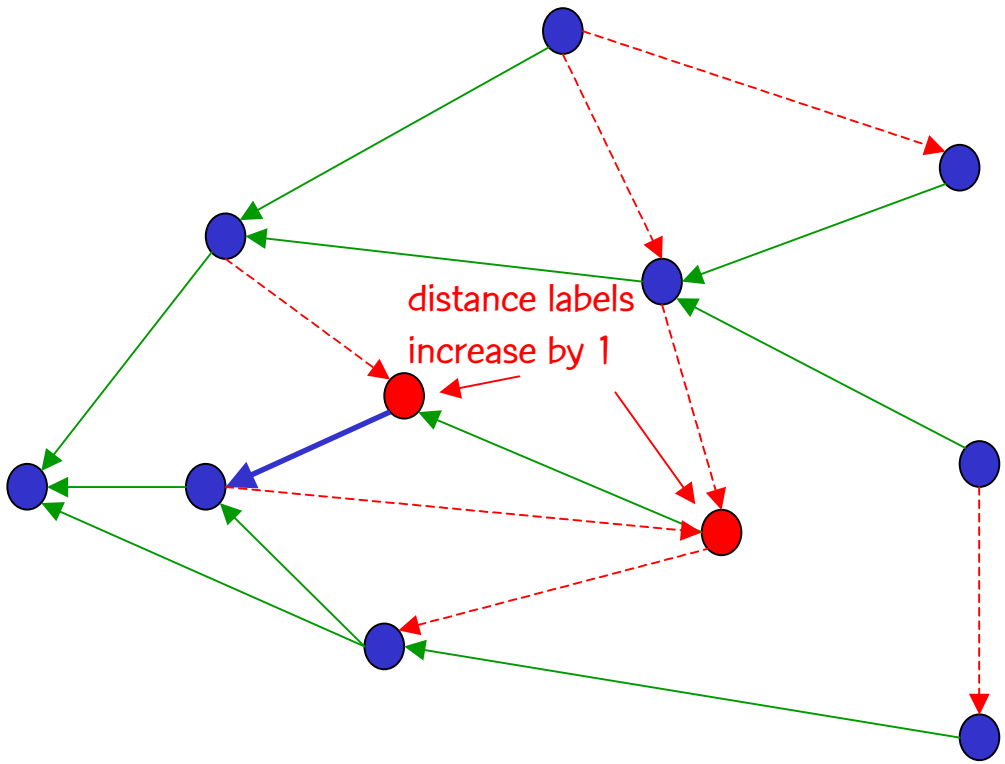


Dynamic shortest path



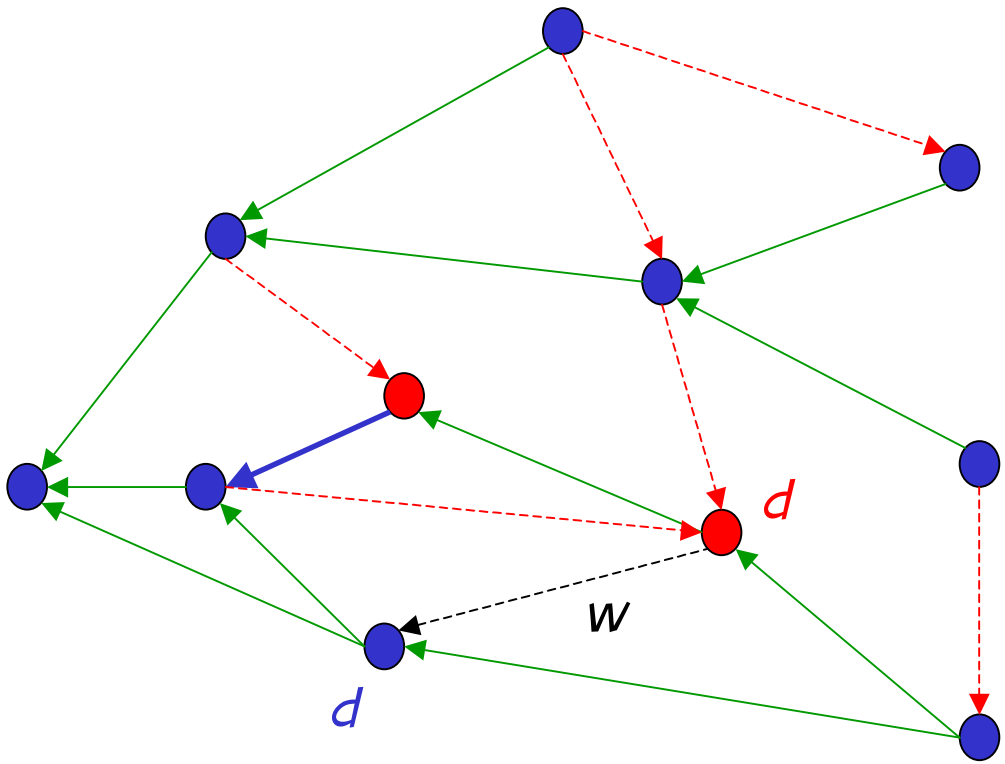
Arc (u,v) is removed from tree since alternative paths from node u to the destination node exist.

Dynamic shortest path

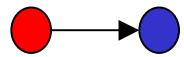


Shortest paths from red nodes must traverse blue arc.

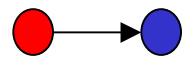
Dynamic shortest path



Test all arcs of type

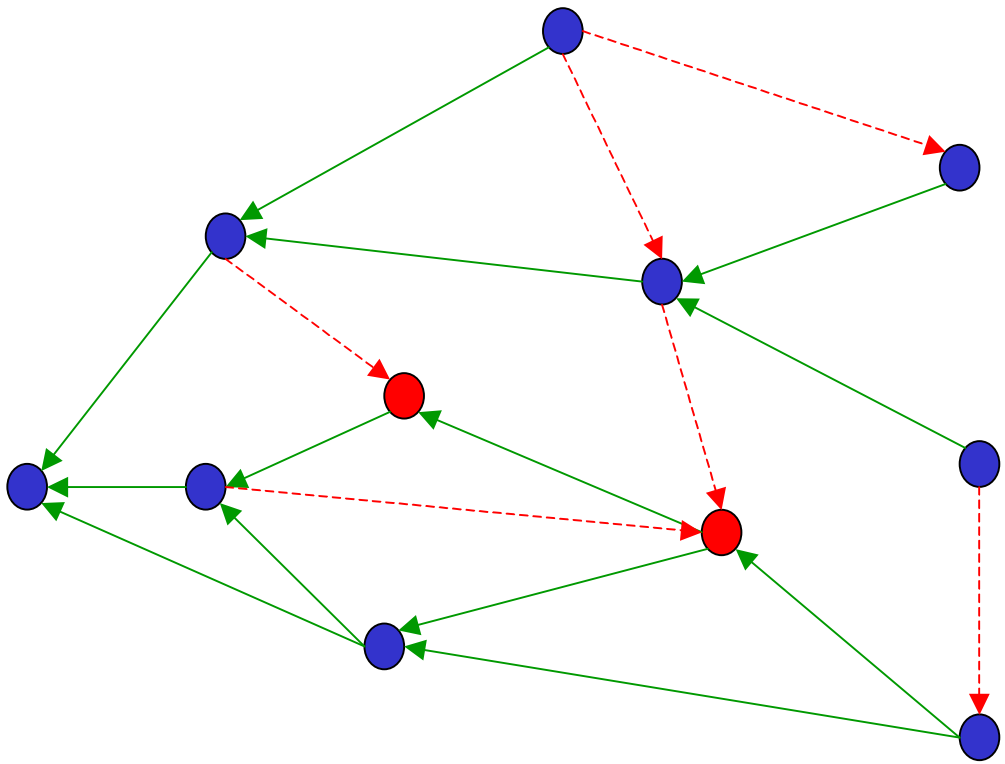


If $d - d = w$, then



enters tree.

Dynamic shortest path

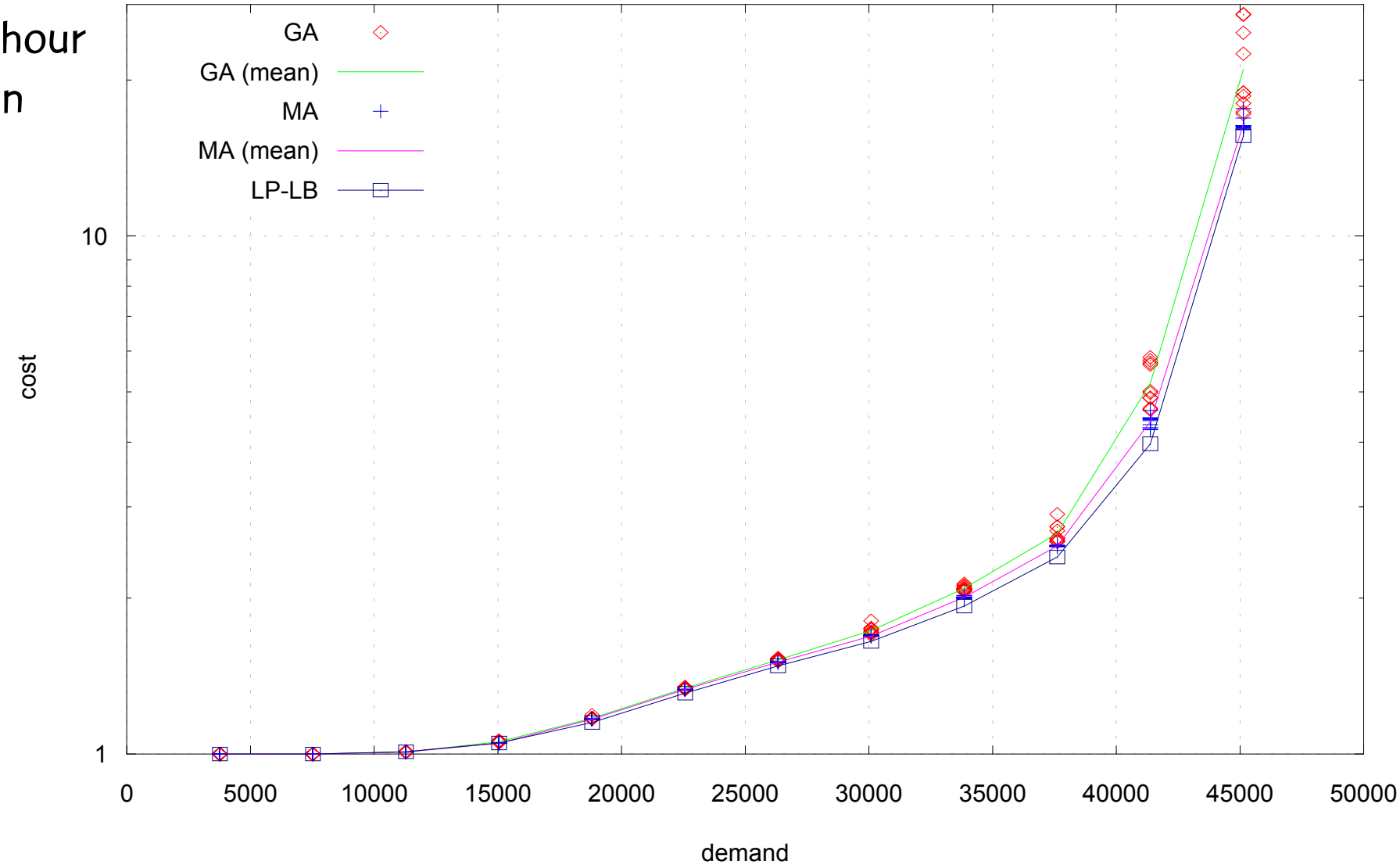


Dynamic shortest path

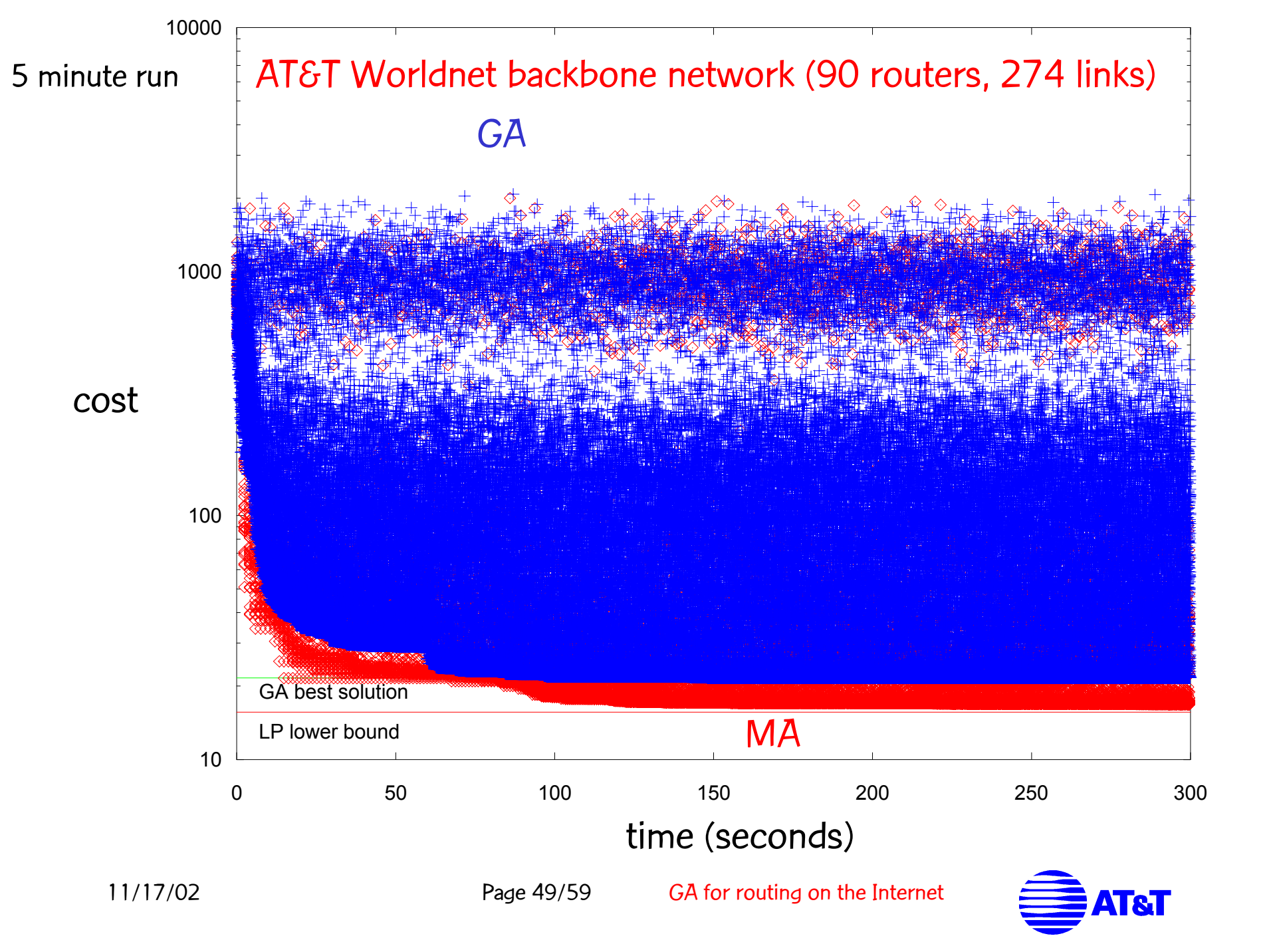
- Ramalingam & Reps (1996) allow arbitrary arc weight change.
- We specialized the Ramalingam & Reps algorithm for unit arc weight change.
  - Avoid use of heaps
  - Achieve a factor of 2~3 speedup w.r.t. Ramalingam & Reps on these test problems

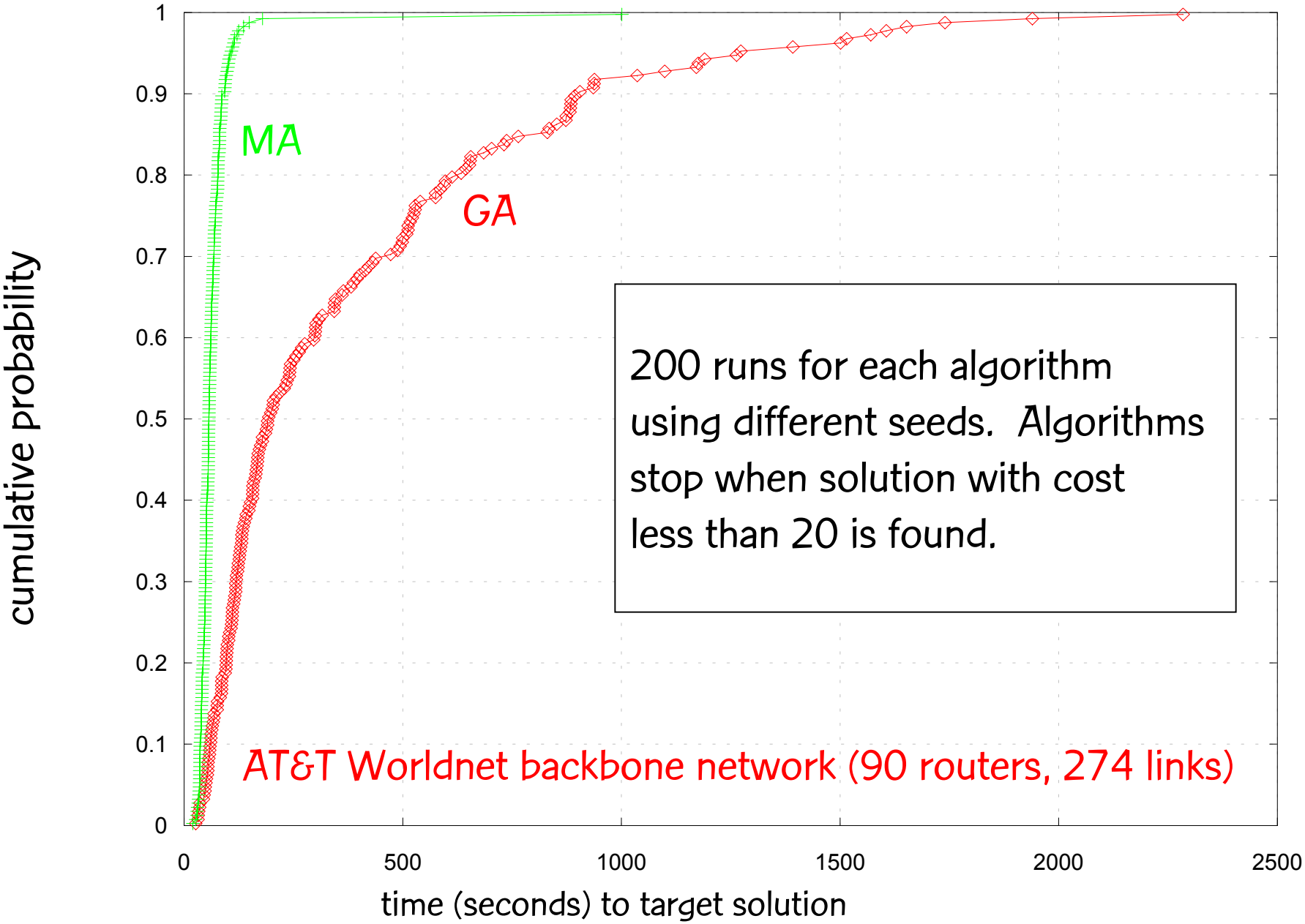
AT&T Worldnet backbone network (90 routers, 274 links)

1 hour run



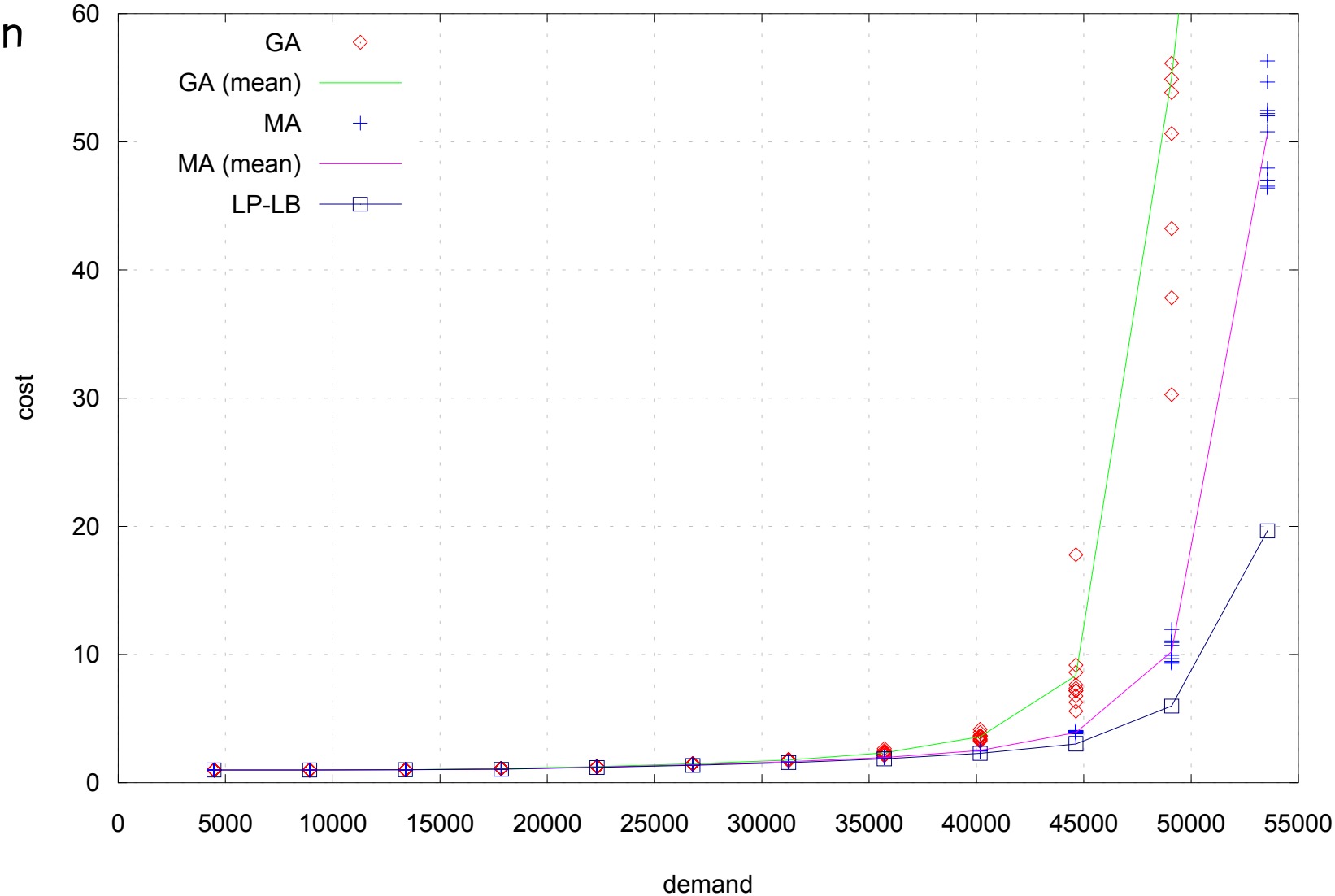
AT&T


5 minute run
AT&T Worldnet backbone network (90 routers, 274 links)
GA
MA
cost
10000
1000
100
10
GA best solution
LP lower bound
0
50
100
150
200
250
300
time (seconds)


cumulative probability
time (seconds) to target solution
MA
GA
200 runs for each algorithm using different seeds. Algorithms stop when solution with cost less than 20 is found.
AT&T Worldnet backbone network (90 routers, 274 links)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
500
1000
1500
2000
2500

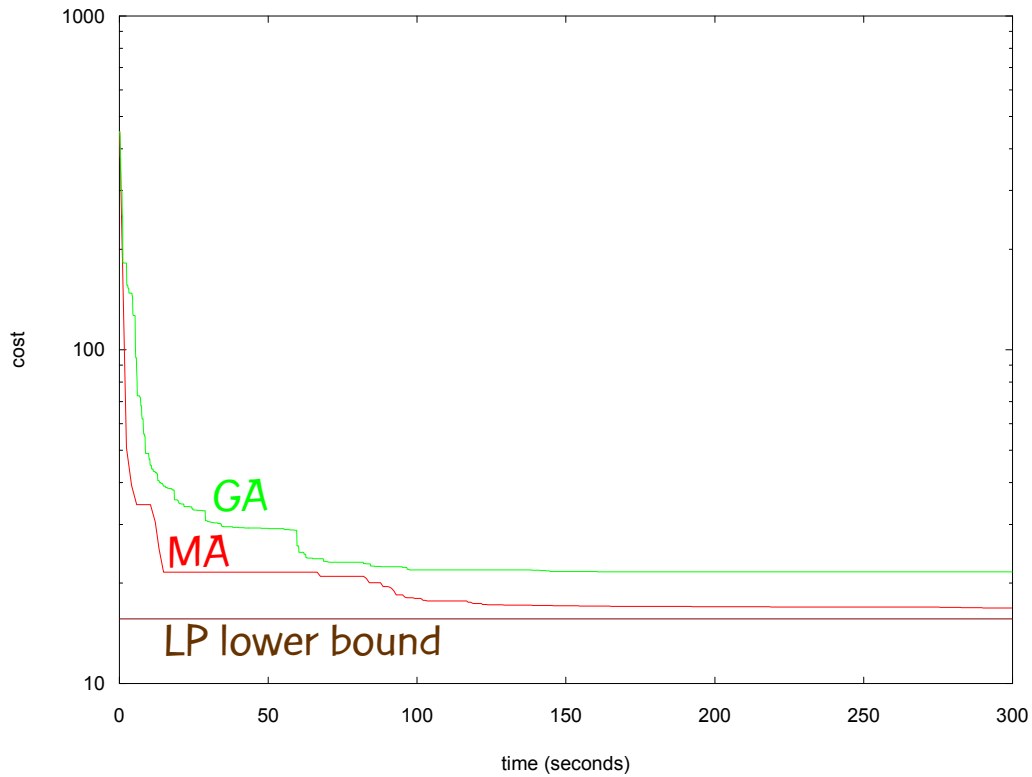
Rand50a: random graph with 50 nodes and 245 arcs.

1 hour run

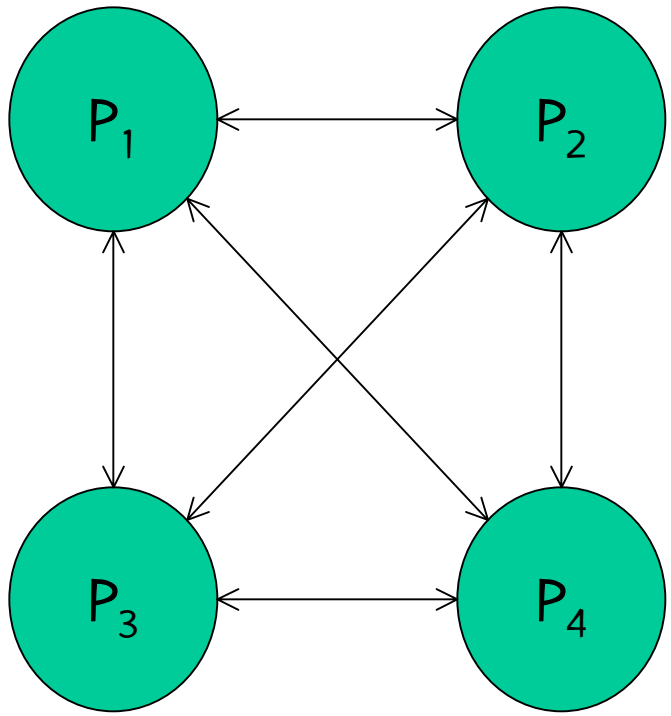


Remark

- Memetic algorithm (MA) improves over pure genetic algorithm (GA) in two ways:
  - Finds solutions faster
  - Finds better solutions

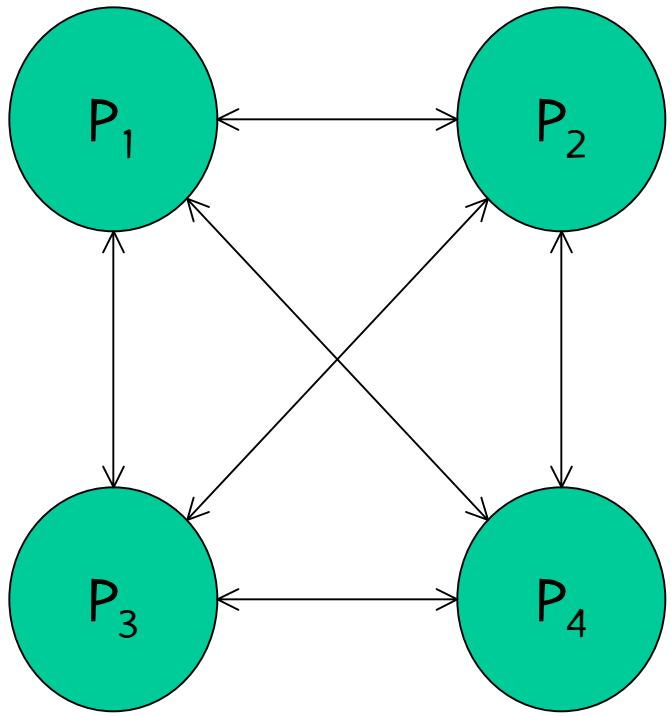


Collaborative parallel implementation



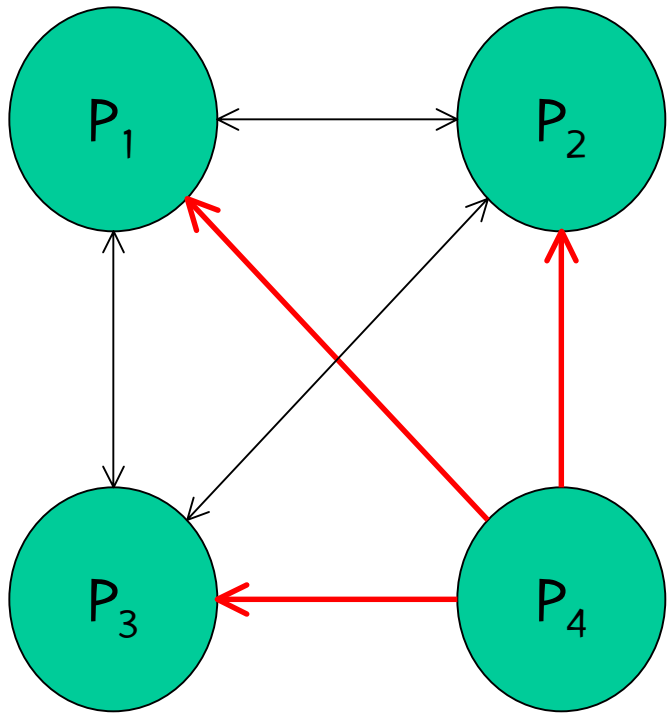
MPI: Message Passing Interface

Collaborative parallel implementation



If P_4 finds a new incumbent solution.

Collaborative parallel implementation

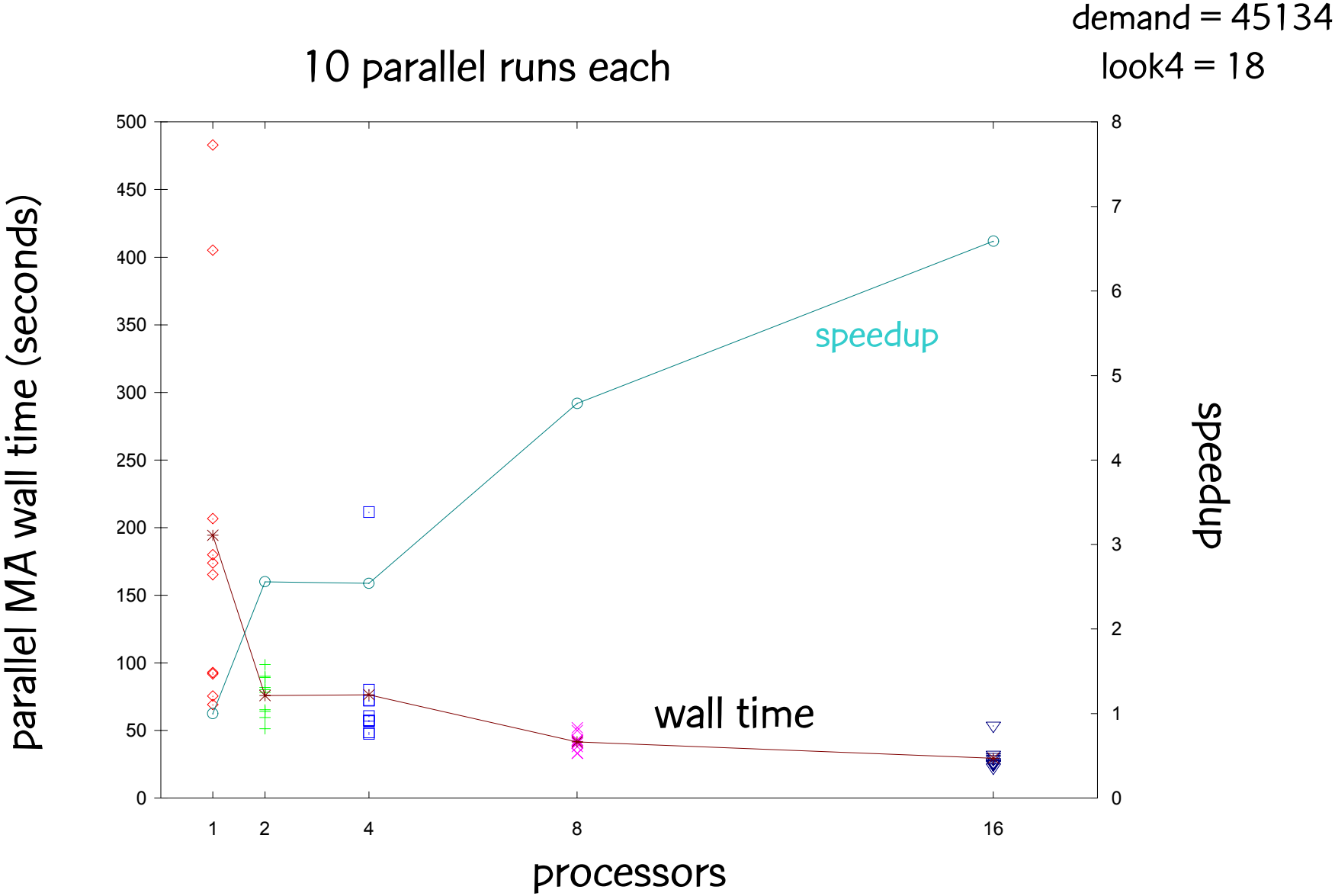


If P_4 finds a new incumbent solution.
Incumbent solution is broadcast to P_1, P_2, P_3.

AT&T Worldnet backbone network (90 routers, 274 links)

demand = 45134

look4 = 18



Extensions

- Network design: Minimize total multiplicity of links to guarantee traffic flow subject to single link failures.
- Routing: Minimize maximum utilization subject to single link and router failures.
- Server placement: Locate minimum number of cache servers on network for multi-cast of streaming video.

Concluding remarks

- Slides of this talk can be downloaded from: http://www.research.att.com/~mgcr
- Paper on GA: M. Ericsson, M.G.C. Resende, and P.M. Pardalos, "A genetic algorithm for the weight setting problem in OSPF routing," J. Comb. Opt., vol. 6, pp. 299-333, 2002.
- Soon at http://www.research.att.com/~mgcr, papers on:
  - GA with opt. crossover (L.S. Buriol, M.G.C. Resende, C.C. Ribeiro, and M. Thorup)
  - Speeding up dynamic shortest path algorithms (L.S. Buriol, M.G.C. Resende, and M. Thorup)

My coauthors

Photos from their homepages.

Luciana Buriol
UNICAMP, Brazil &
AT&T Labs Research

Celso Ribeiro
PUC-Rio, Brazil

Panos Pardalos
U. of Florida

Mikkel Thorup
AT&T Labs Research

?

Märten Ericsson
KTH, Sweden

AT&T