Combinatorial optimization in telecommunications

Mauricio G. C. Resende

Algorithms & Optimization Research Dept.

AT&T Labs Research

Florham Park, New Jersey

mgcr@research.att.com

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

Talk given on August 6, 2003 at

COPPE/UFRJ – Rio de Janeiro, RJ - Brazil

Collaborators

- AT&T Labs Research
  - J. Abello, D. Applegate, D. S. Johnson, C. Lund, S. Phillips, N. Reingold, L. Resende, and M. Thorup
- Lucent Bell Labs
  - K. G. Ramakrishnan and P.M. Winkler
- Catholic U. of Rio de Janeiro
  - S. Canuto, S.L. Martins, and C. Ribeiro
- U. Federal Fluminense
  - C. Ribeiro
- U. of Napoli
  - P. Festa

Collaborators

- U. of Florida
  - P.M. Pardalos and C.A. Oliveira
- Fed. U. of Rio de Janeiro
  - A. Lucena
- Royal Institute of Technology (KTH)
  - M. Ericsson
- UNICAMP
  - L. Buriol and P. França
- Princeton U.
  - R. Werneck

Summary

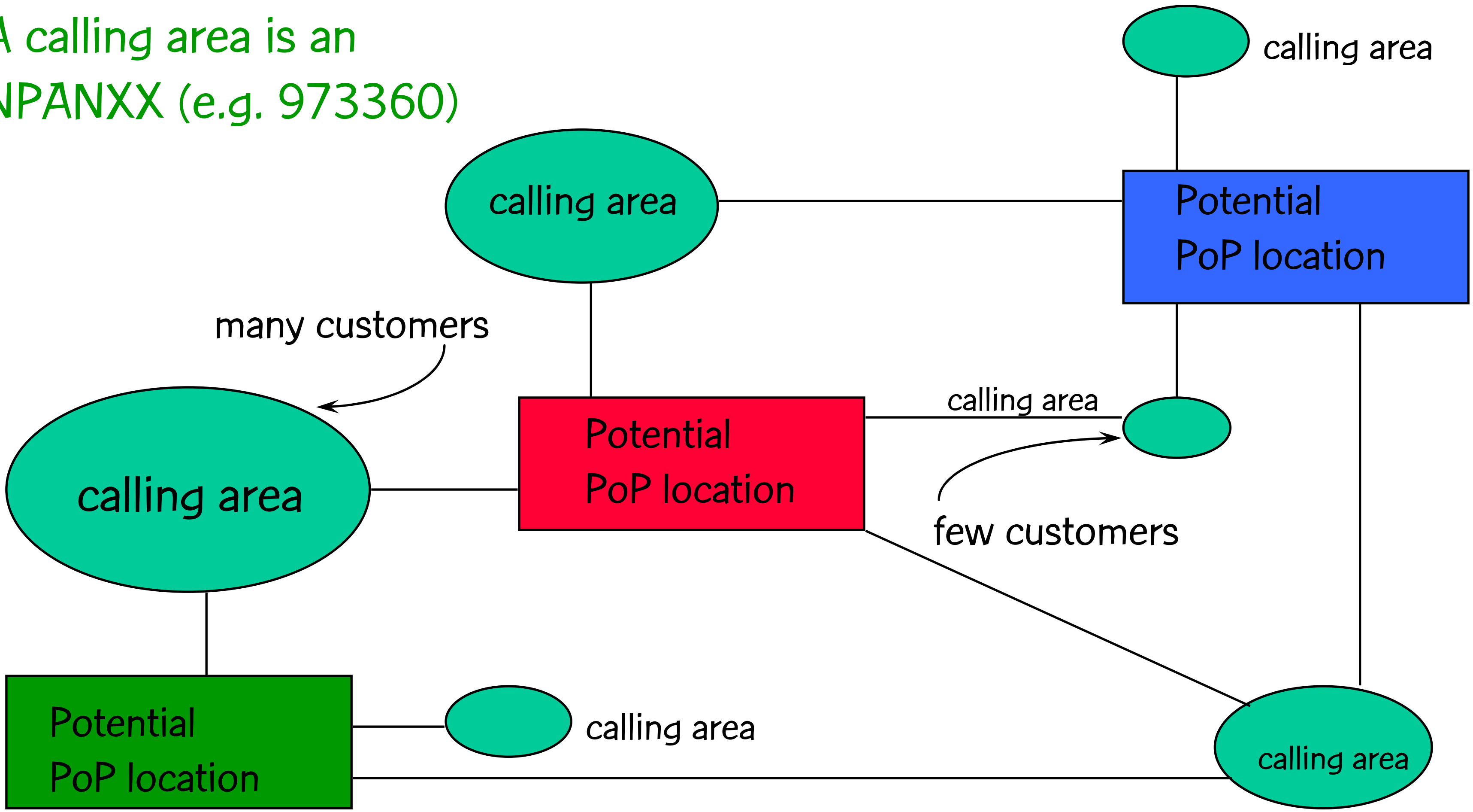
- modem location for internet service provider
  - metaheuristic & LP-based upper bounds
- design local access network
  - metaheuristic & LP-based lower bounds
- SONET ring network design
  - LP-based solution procedure
- offline routing of permanent virtual circuits
  - Metaheuristic
- OSPF routing
  - metaheuristic & LP-based lower bounds

Modem pool location for dial-up ISP access

- user dials up to a modem to access an internet service provider
- modem pools are located at PoPs (points of presence)
- users prefer making free local calls to access internet service

ISP access

A calling area is an NPANXX (e.g. 973360)



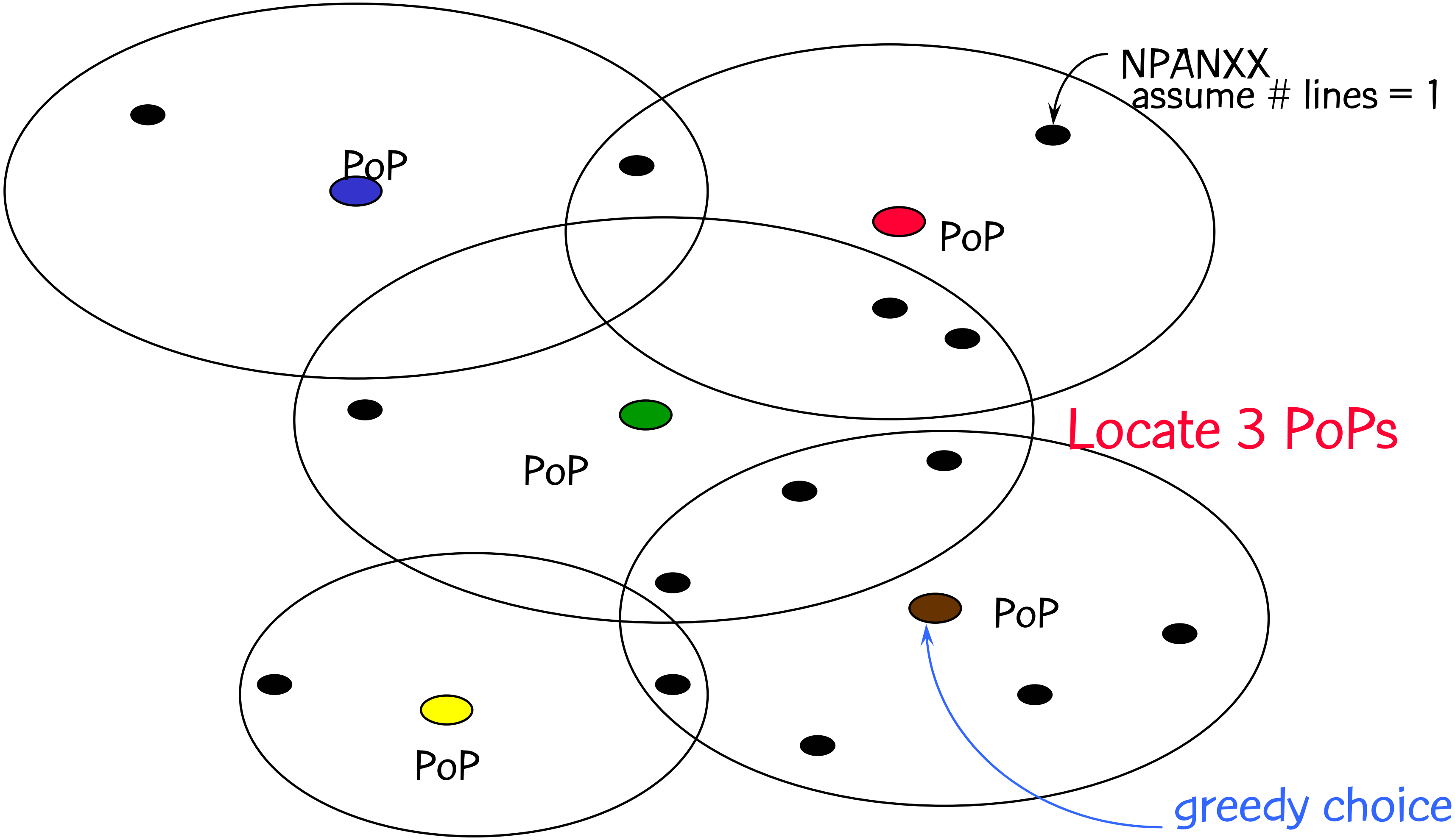
Location problem

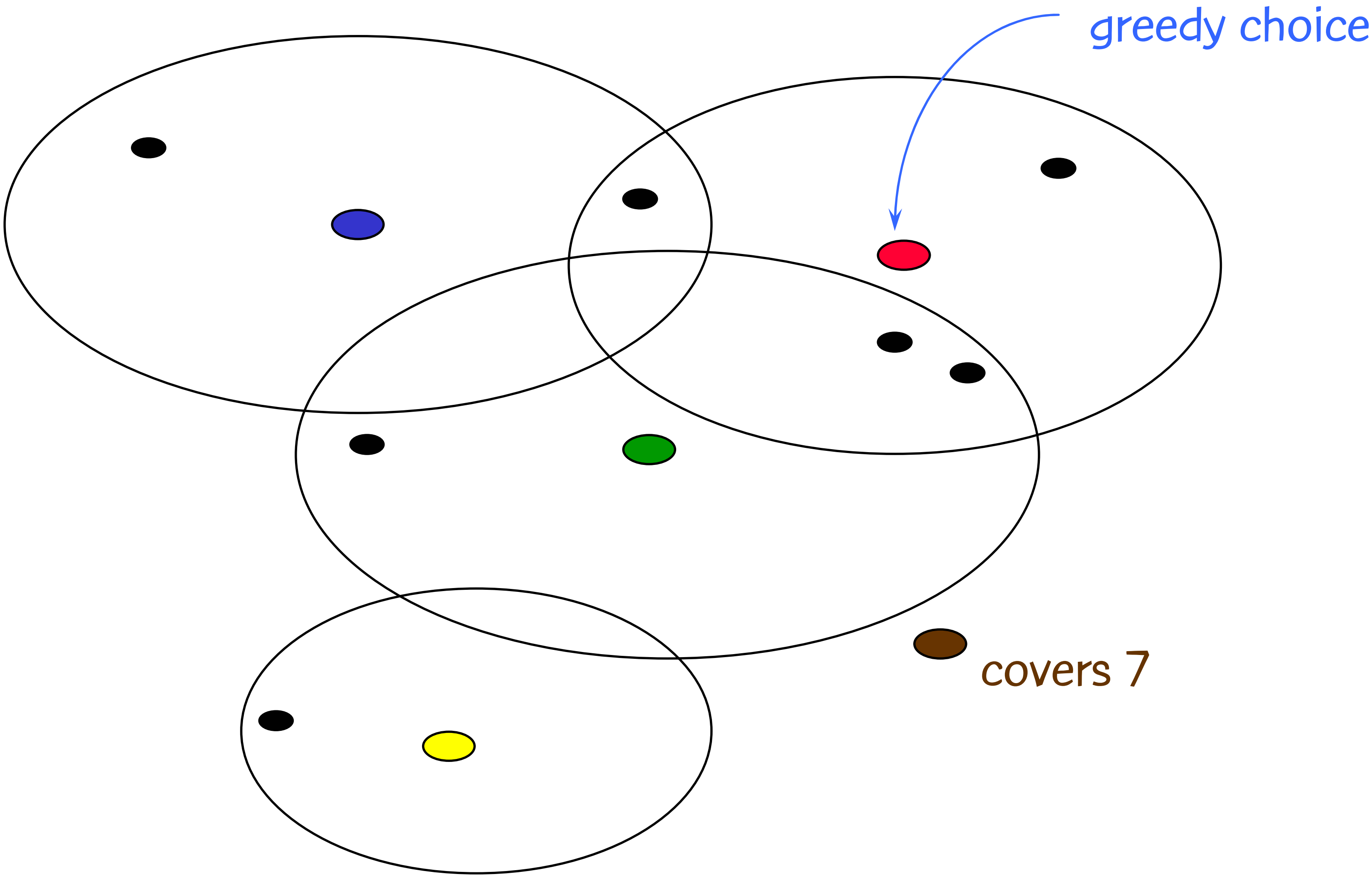
- maximize number of customers that can make free local calls to a PoP
- where to locate PoPs
  - fixed number of PoPs
  - choose from set of potential PoP locations

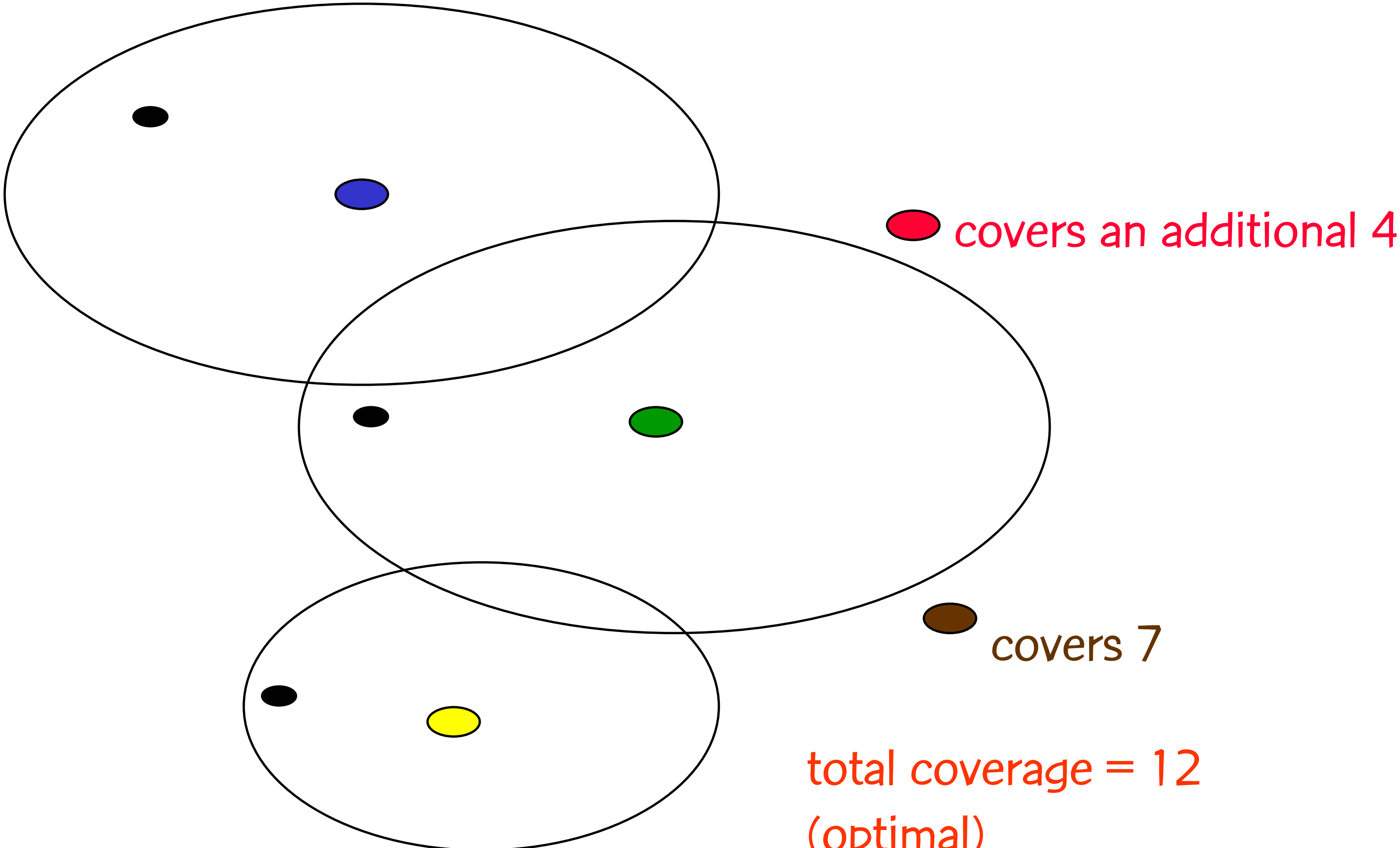
Typical size

- ~ 50,000 potential PoP locations
- ~ 50,000 calling areas (NPANXX)
- ~ 120 million residential lines
- + 255 PoPs to be located

Greedy algorithm

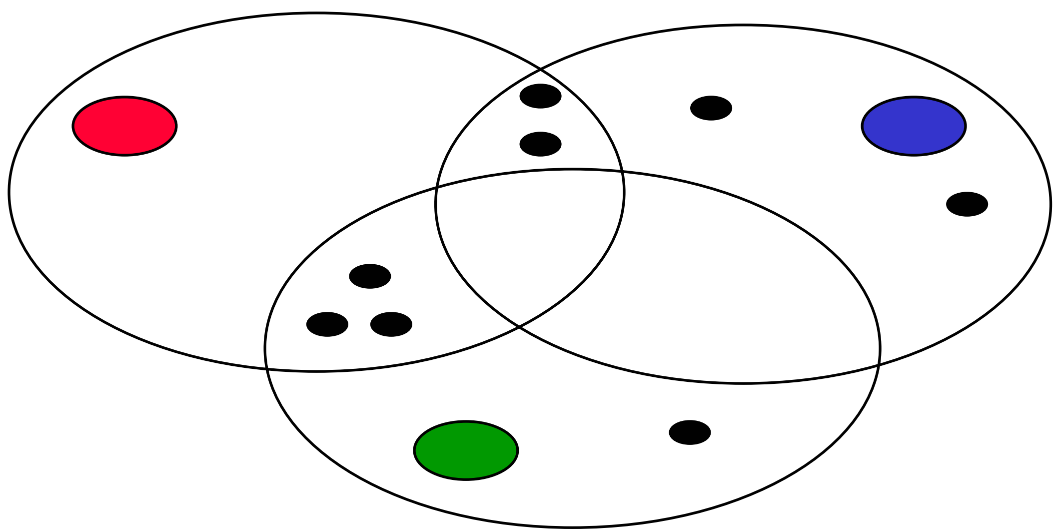



greedy choice
covers 7


covers an additional 4
covers 7
total coverage = 12
(optimal)

Greedy algorithm

- Greediness does not always lead to optimality:

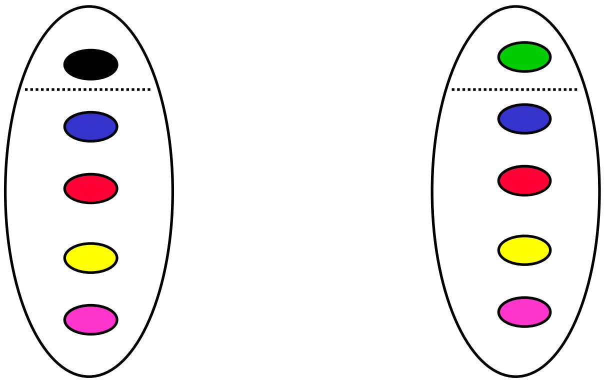


greedy solution: covers 7

nongreedy solution: covers 8

GRASP

- repeated applications of:
  - randomized greedy
    - select next PoP, at random, from quasi-greedy set of PoPs
  - local search
    - 2-exchange:



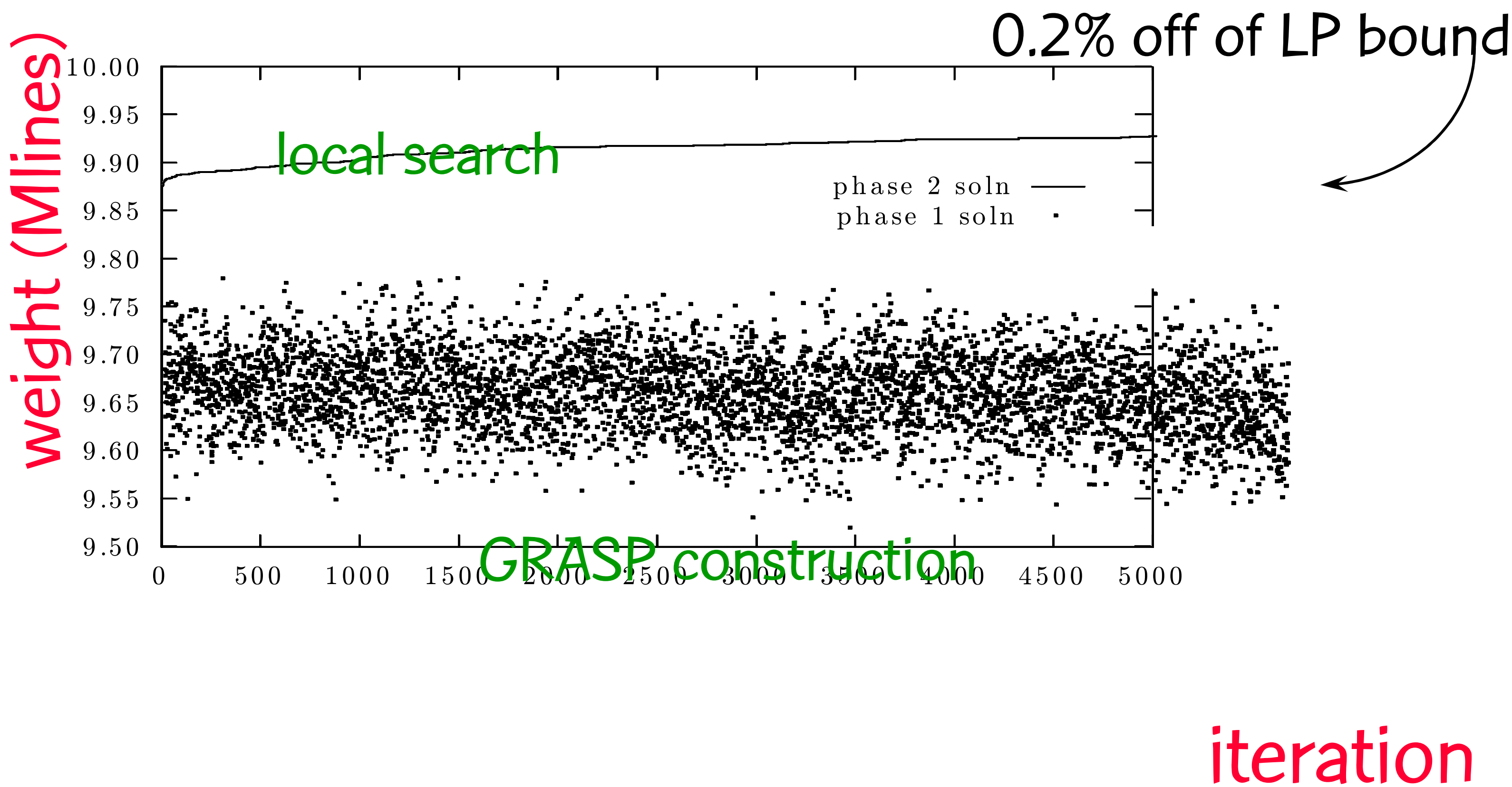
Solution X

Solution Y in neighborhood of X

LP upper bound

- location problem has integer programming formulation
- relaxing integrality constraints results in linear program
- cost of optimal solution of linear program is an upper bound on the maximum number of lines that can make a free call to some selected PoP

GRASP



- 27,521 potential PoP locations
- 18,419 calling areas
- 146 PoPs to be placed

poptool

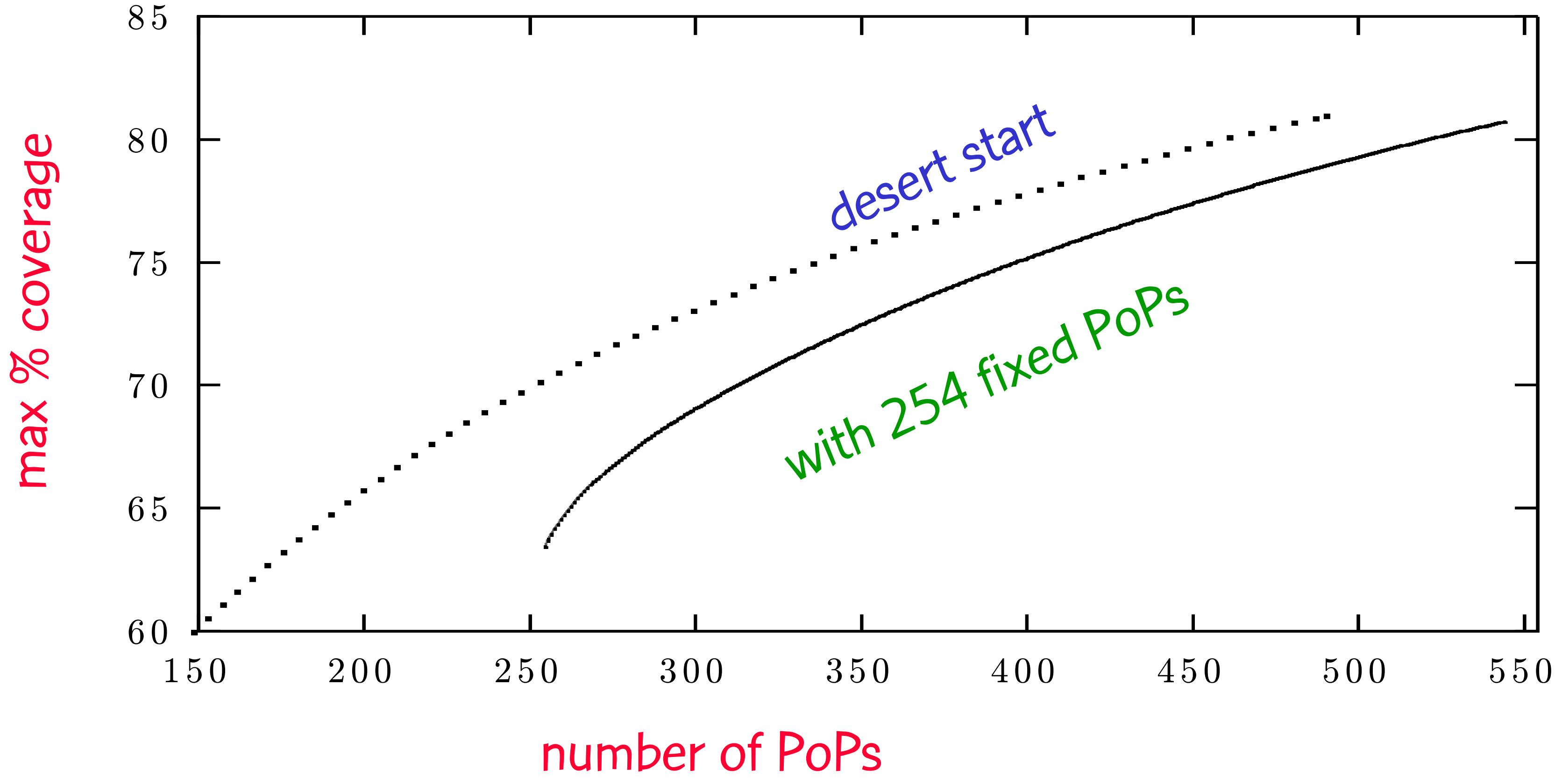
- poptool is a Unix tool that integrates the GRASP with the LP upper bound
- it has been used in a number of studies and applications by AT&T's WorldNet

Coverage determination

"WorldNet will go online with more than 200 PoPs, meaning that from the outset about 80% of all AT&T long-distance customers will have local dial-up access to the Internet." [Tom Evslin, VP for WorldNet Services, NY Times, February 28, 1996]

AT&T

Maximum coverage



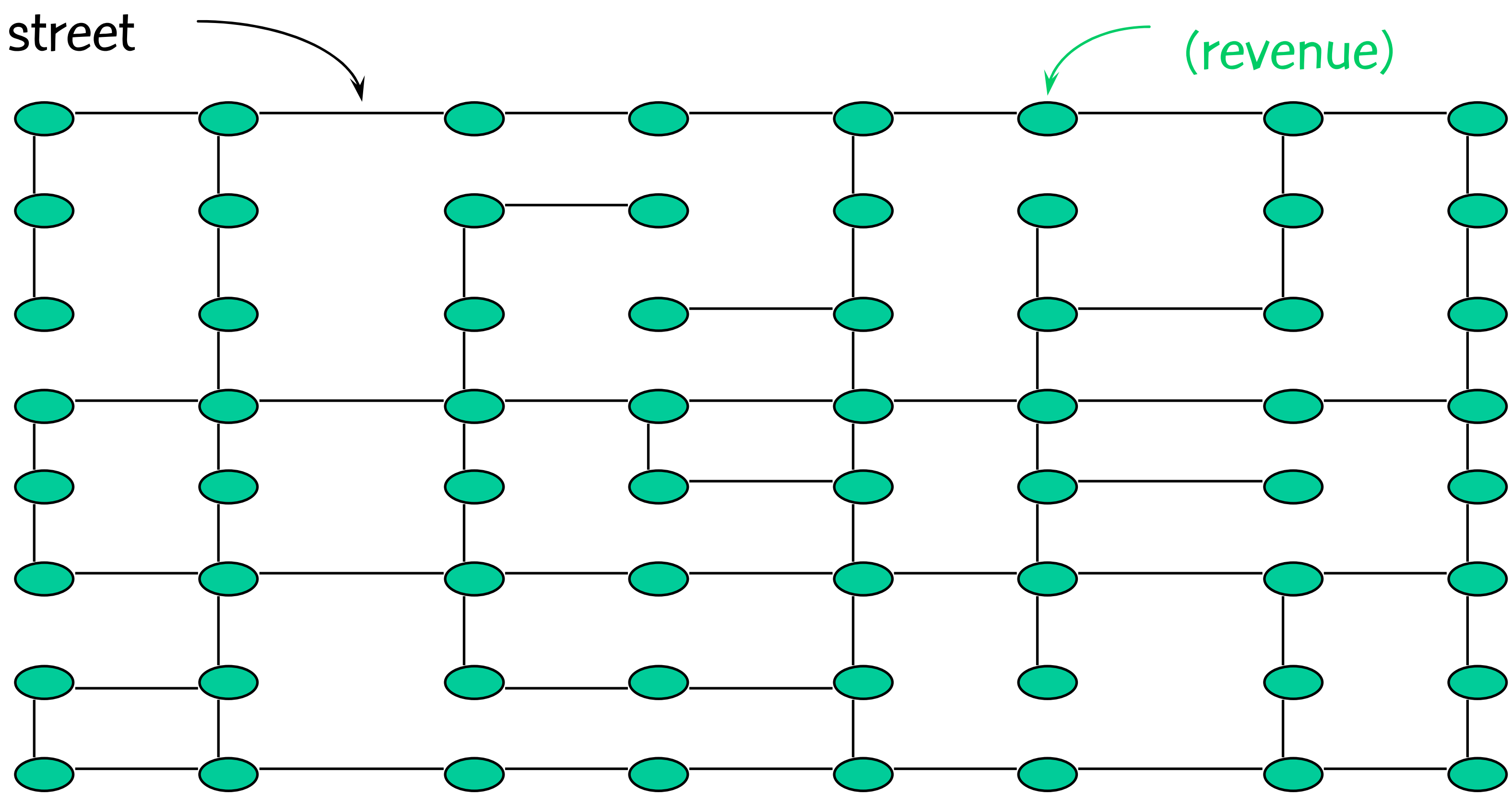
Summary

- modem location for internet service provider
  - metaheuristic & LP-based upper bounds
- design local access network
  - metaheuristic & LP-based lower bounds
- SONET ring network design
  - LP-based solution procedure
- offline routing of permanent virtual circuits
  - Metaheuristic
- OSPF routing
  - metaheuristic & LP-based lower bounds

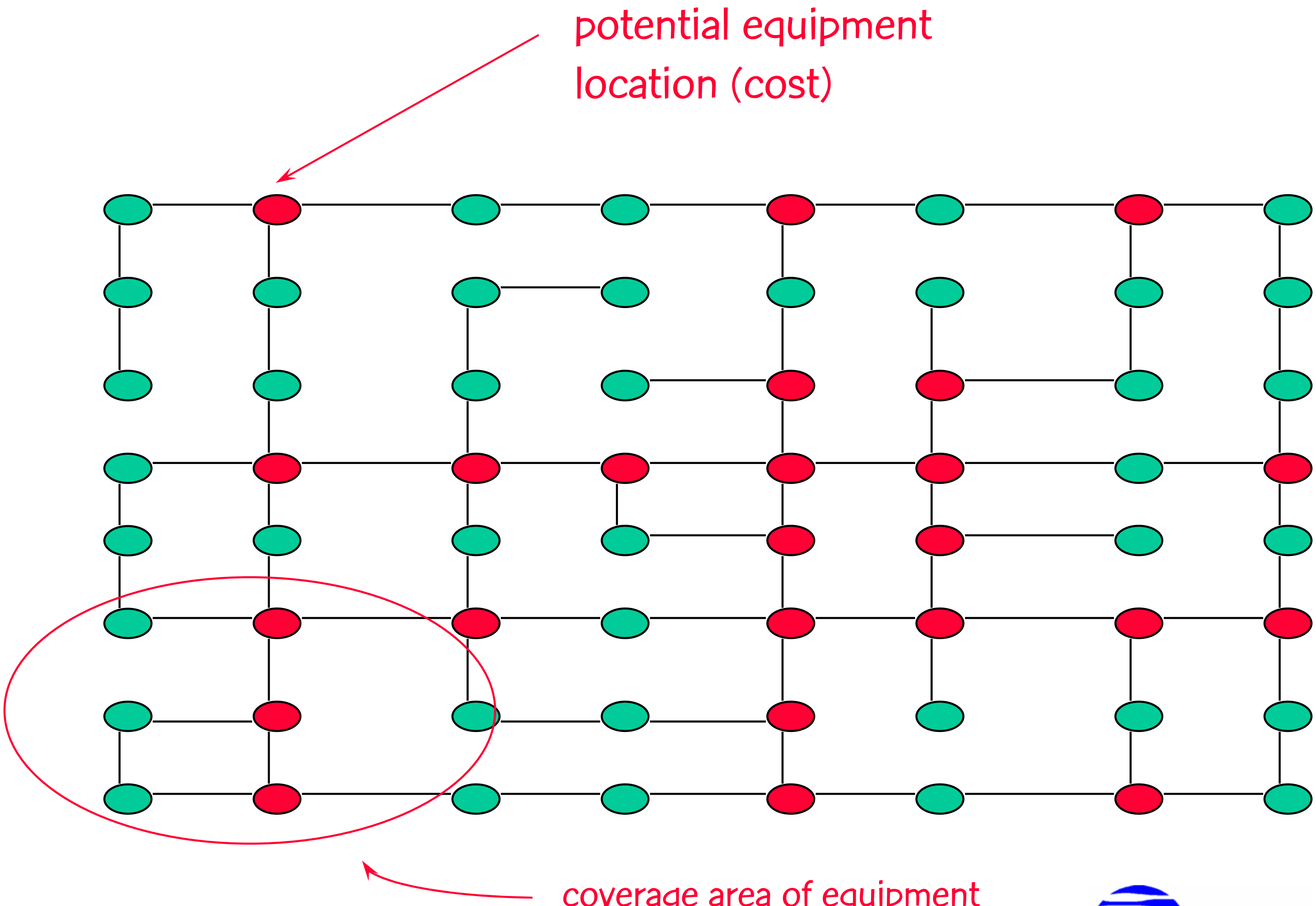
Local access network design

- Design a local access network taking into account tradeoff between:
  - cost of network
  - revenue potential of network

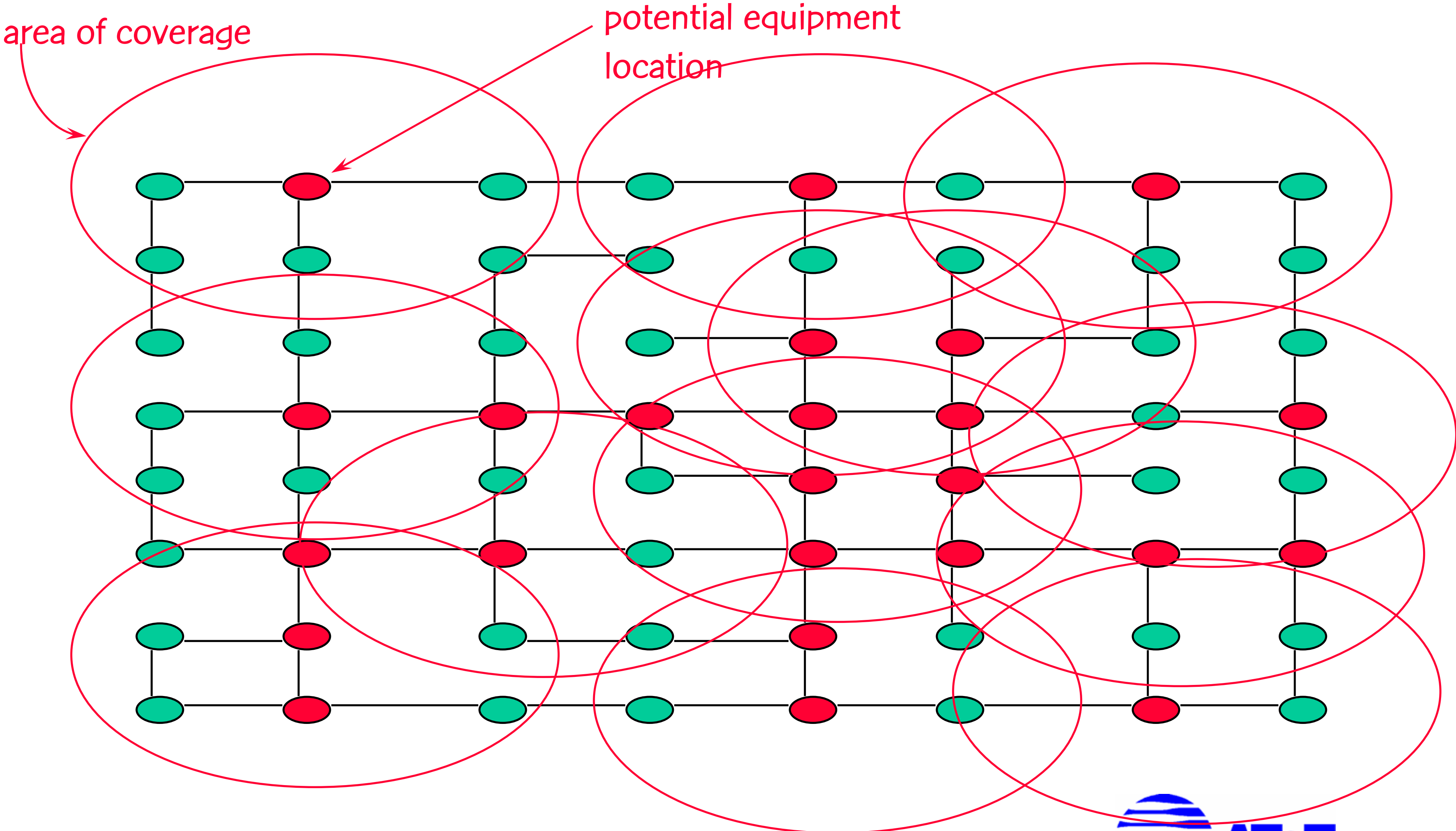
Local access network design



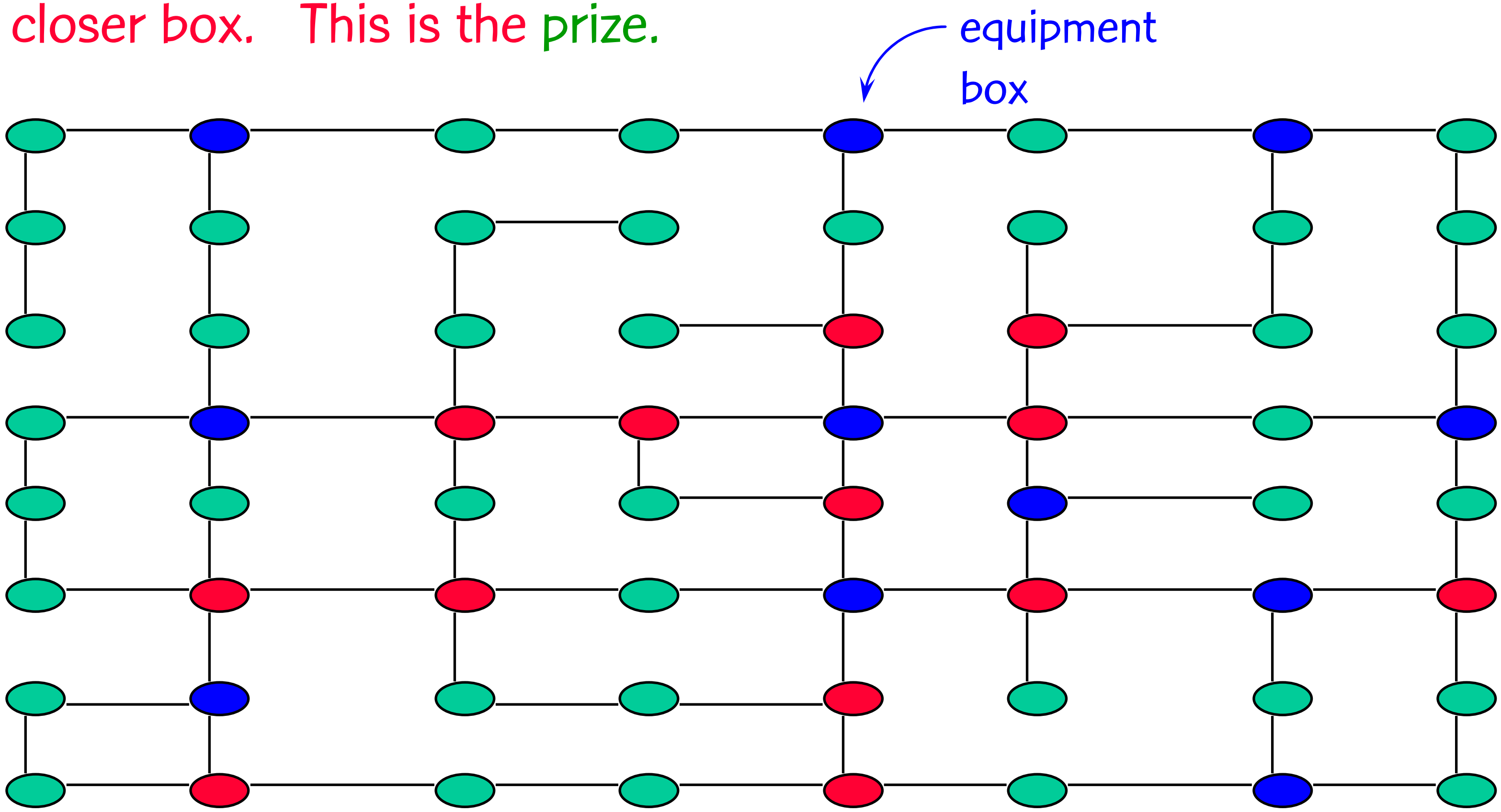
AT&T


potential equipment
location (cost)
coverage area of equipment

Locate p boxes to max revenue of covered residences.

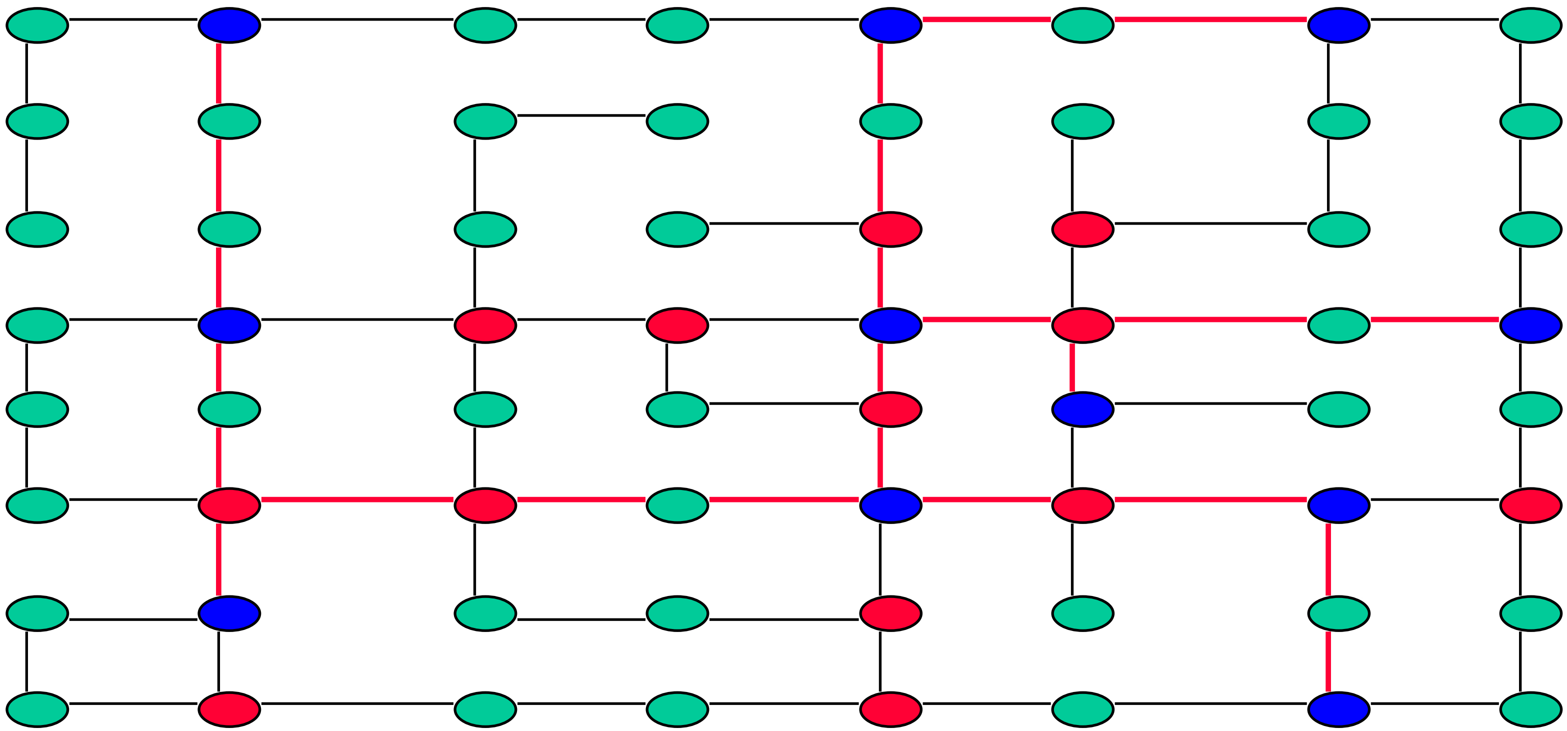


Assign to each equipment box the total revenue of residences it covers which for which there is no closer box. This is the prize.



Solve prize collecting Steiner tree problem

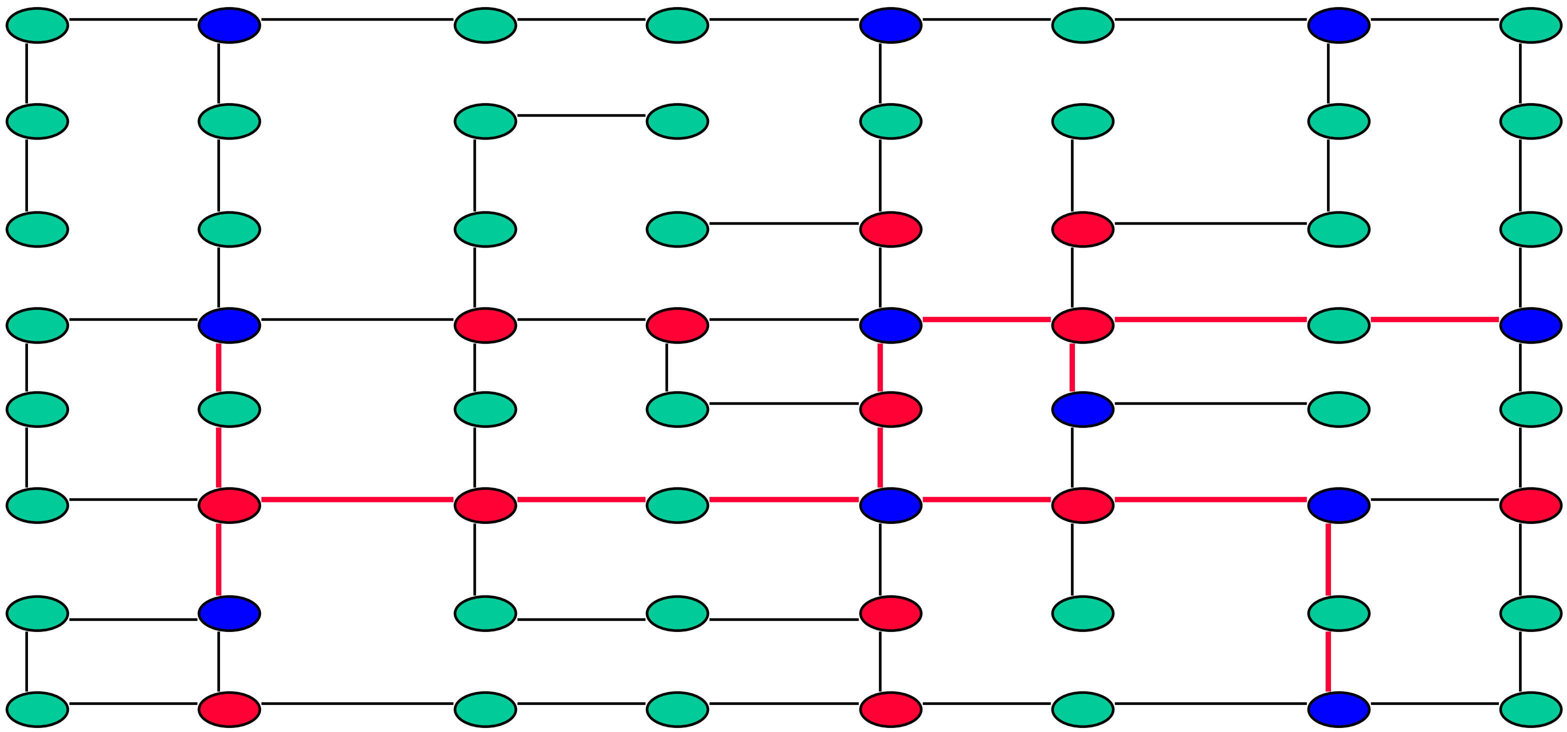
max prize collected minus edge cost



Here all prizes are collected.

Solve prize collecting Steiner tree problem

max prize collected minus edge cost



Here not all prizes are collected.

Solve prize collecting Steiner tree problem

- Typical dimension: 20,000 to 100,000 nodes.
- Use GRASP for max covering to locate equipment boxes
- Compute lower bounds with cutting planes algorithm of Lucena & Resende (Discrete Applied Math., 2003)
- Compute solutions (upper bounds) with local search with perturbations algorithm of Canuto, Resende, & Ribeiro (Networks, 2001)

Summary

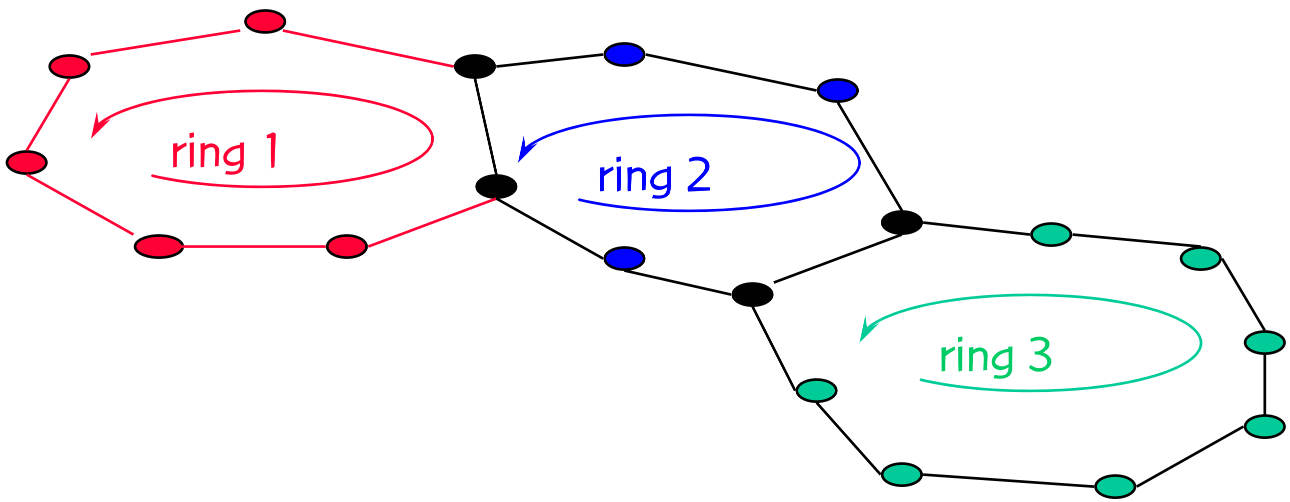
- modem location for internet service provider
  - metaheuristic & LP-based upper bounds
- design local access network
  - metaheuristic & LP-based lower bounds
- ▶ SONET ring network design
  - ▸ LP-based solution procedure
- offline routing of permanent virtual circuits
  - Metaheuristic
- OSPF routing
  - metaheuristic & LP-based lower bounds

SONET ring network design

- Businesses increasingly depend on continuous reliable communications.
  - For example: banks, brokerage houses, reservation systems, and credit card companies.
- Demand for survivable telecom networks with fast restoration capability is growing.
- Synchronous Optical Network (SONET) has enabled deployment of networks with a high level of service availability.
- SONET compatible equipment is capable of detecting problems with the signal and quickly (in less than a millisecond) react to reestablish communications.

SONET ring network design

- SONET is generally configured as a network of self-healing rings
  - each pair of demand points has at least two disjoint paths



SONET ring network design

- Telecommunications network can be represented by $G = (V, E)$, where:
  - V are vertices, each representing a large customer or a remote terminal (where low bandwidth traffic is aggregated) or a central office where switching takes place
  - E are edges or links, each representing a fiber cable connecting two nodes

SONET ring network design

- Demand between pairs of nodes (not all pairs have demands) is an estimate of the number of circuits needed to provide communications between that pair of nodes.
- Demand is given in units of DS3 (51.84 Mbits/sec).

SONET ring network design

- SONET equipment is configured on logical rings (a ring is a cycle in G)
- SONET network is a set of rings that covers the nodes of G and that allows the demand to be satisfied.
- Demand between pair of nodes is satisfied if bandwidth equal to the required number of DS3s is reserved on one or more paths between the pair of nodes, where paths traverse only nodes with SONET equipment.

SONET ring network design

- Equipment required:
  - add-drop multiplexer (ADM) at each node per unit of OC48 (48 DS3)
  - two dense wave division multiplexer (DWDM) per link per eight units of OC48 traversing that link
  - optical amplifier (OA) every 75 miles per OC48
  - signal regenerator (REGEN) every 225 miles per OC48
  - Digital cross-connect system (DCS) at each demand node and each interring node (one for each 3 units of DS3)

SONET ring network design

- Rings are selected from a set of predetermined rings (candidate rings).
- Design problem: given nodes, links, demands, and set of candidate rings, find a minimum cost SONET ring network using only rings from candidate set such that resulting equipment and fiber links have sufficient capacity to satisfy the demands.

Multicommodity flows

- Point-to-point demands are commodities that flow on the network sharing link and node resources.
- Ring size is a function of the maximum capacity over all links in the ring.
- Costs are linear functions of ring, link, and node capacities.
- Objective: move demand between demand pairs only on links that are part of at least one ring.
- Find optimal flows, and ring and link capacities.

LP variables

- $y(r,k)$ = demand of commodity k unloaded at sink node k from ring r
- $z(r, i, k)$ = demand of commodity k loaded on ring r at node i
- $f(r, k, i, j)$ = flow of commodity k on ring r directed from node i to node j
- $x(r, s, n, k)$ = crossover flow at node n of commodity k from ring r to ring s
- $u(r)$ = size of ring r
- $w(l)$ = size of link l

LP constraints

- Demand unloaded at k must satisfy demand of k.
- Demand loaded at n must satisfy demand of n.
- Flow conservation is done at ring level.
- Sum of bi-directional flows on links on ring determine size of ring.
- Ring sizes are limited by link sizes.
- Ring sizes are bounded from above.

LP cost function

- total cost is sum of
  - total ring cost
  - total link cost
  - total cross-connect cost
  - total loading cost
  - total unloading cost

Heuristic solution of integer program

- IP is too large for current IP packages
- Use a GRASP rounding scheme to produce feasible integer solutions from the optimal LP solution
  - idea: ring sizes may be fractional
    - round up each fractional ring size
    - enough slack may be produced to allow one or more rings to be reduced by one or more units

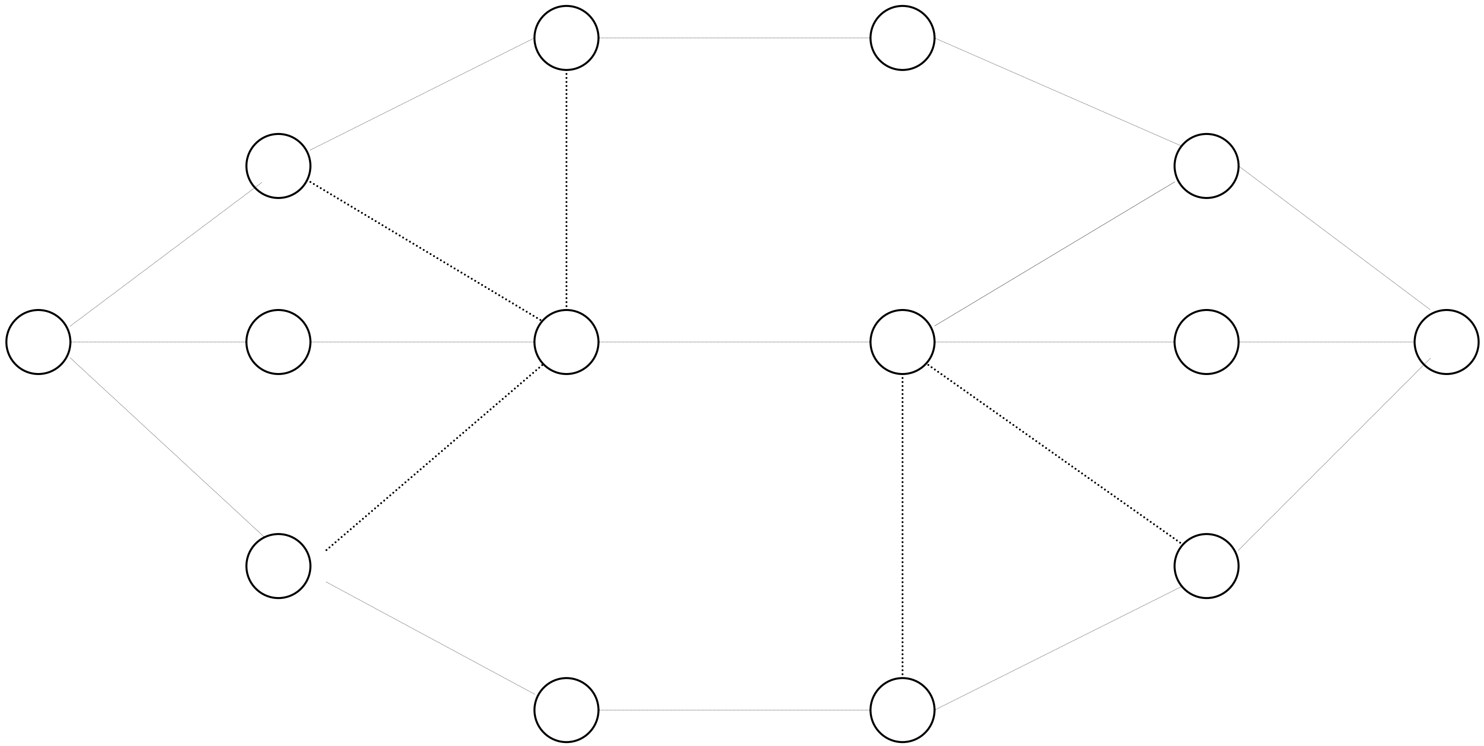
Summary

- modem location for internet service provider
  - metaheuristic & LP-based upper bounds
- design local access network
  - metaheuristic & LP-based lower bounds
- SONET ring network design
  - LP-based solution procedure
- ▶ offline routing of permanent virtual circuits
  - ▸ Metaheuristic
- OSPF routing
  - metaheuristic & LP-based lower bounds

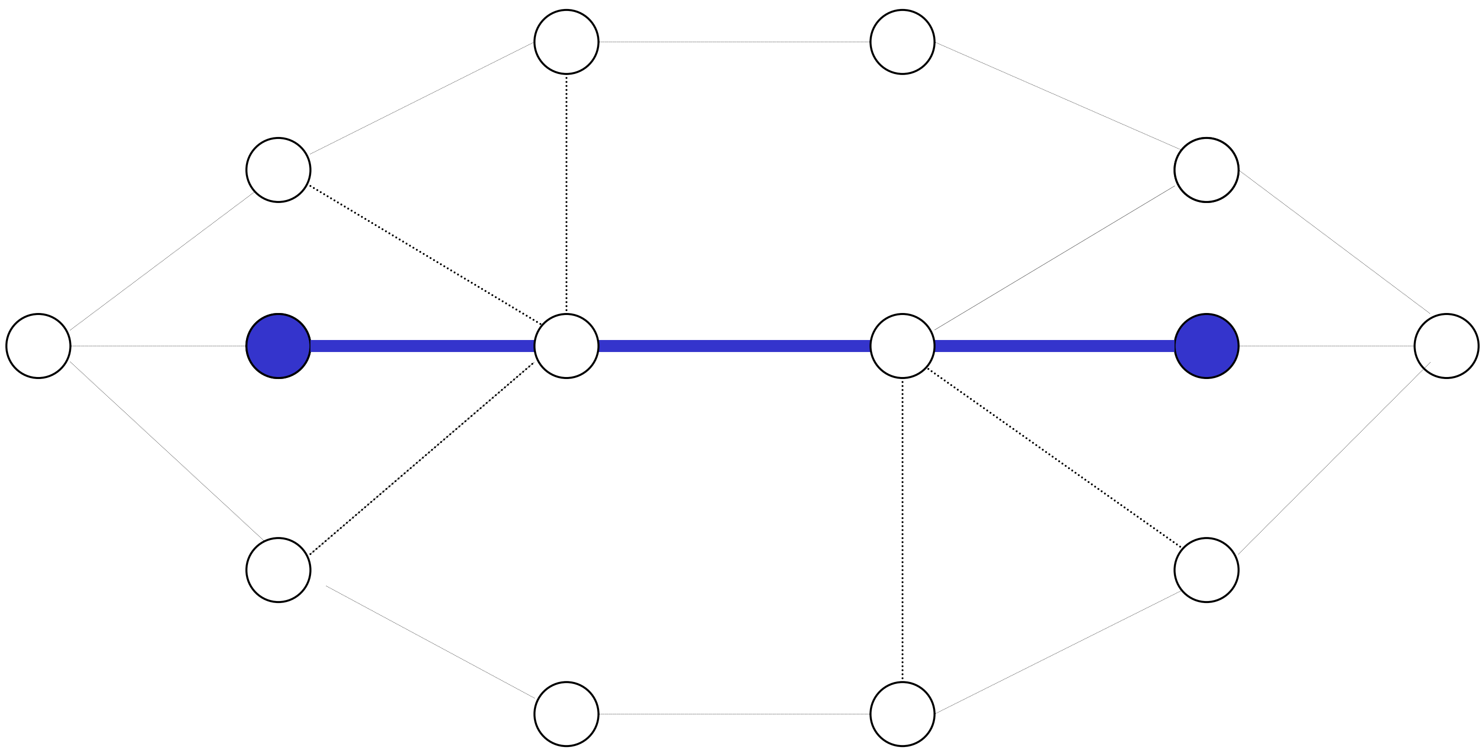
Routing Frame Relay Permanent Virtual Circuits (PVC)

- Frame relay (FR) service
  - provides virtual private networks to customers
  - by provisioning a set of permanent (long-term) virtual circuits (PVC) between customer endpoints on the backbone network
- Provisioning of PVCs
  - routing is done either automatically by switch or by network designer without any knowledge of future requests
  - over time these decisions cause inefficiencies in network and occasional rerouting of PVCs is needed

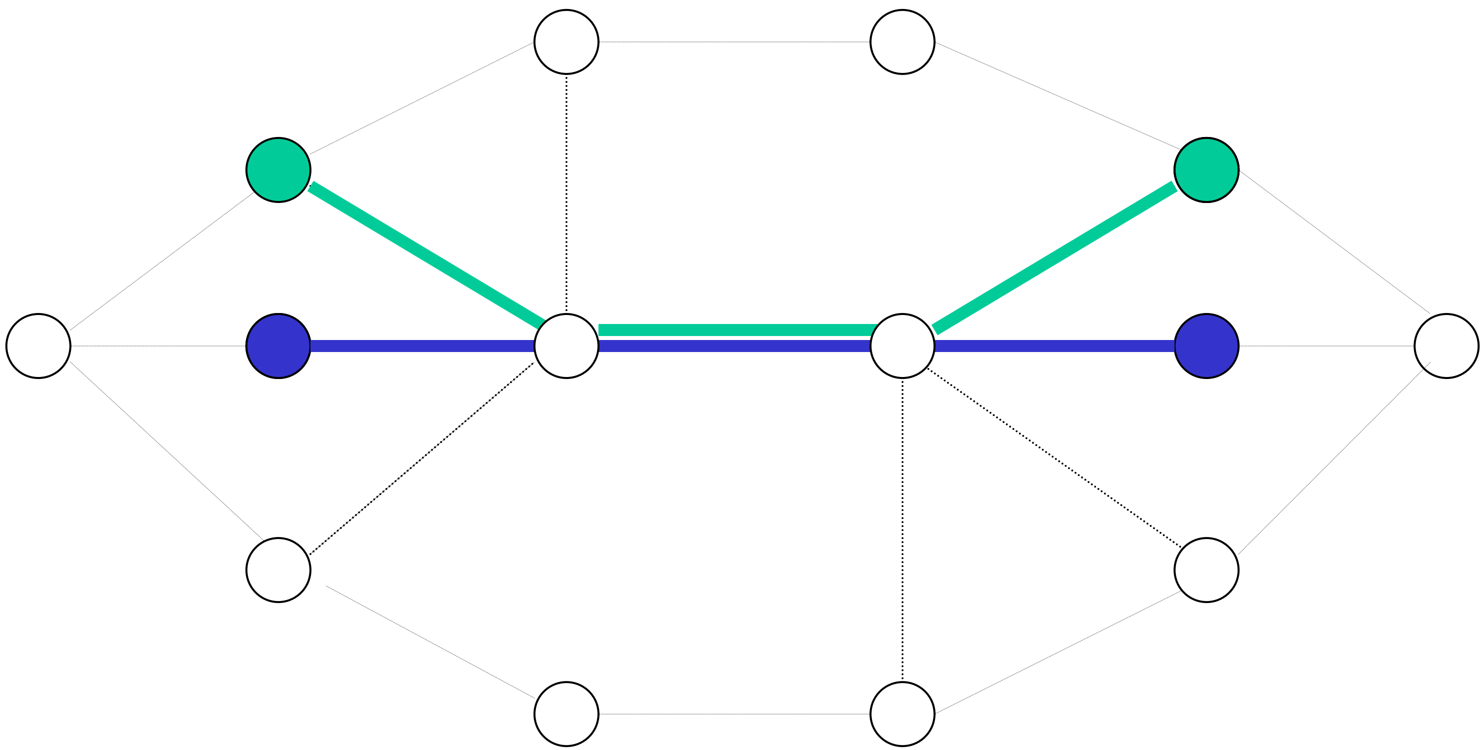
PVC routing: example



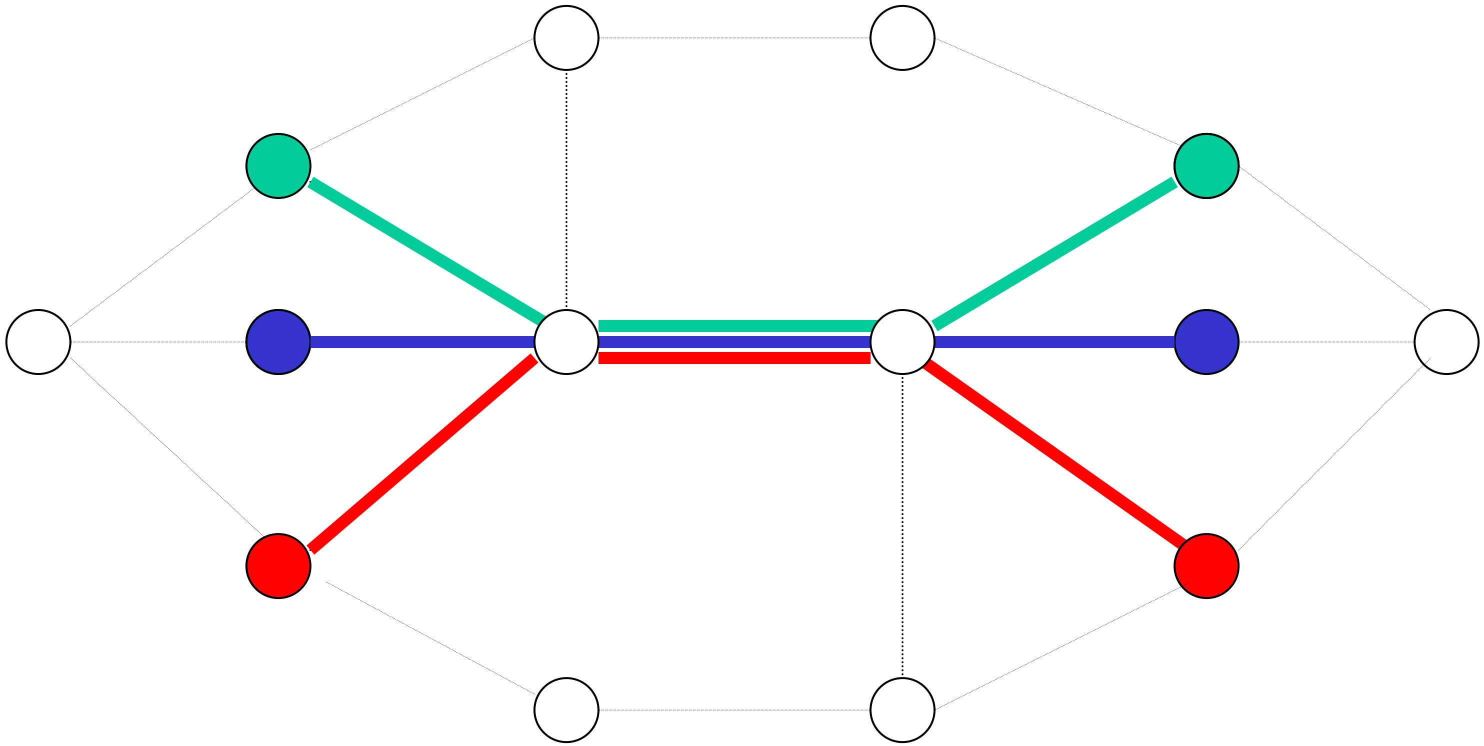
PVC routing: example



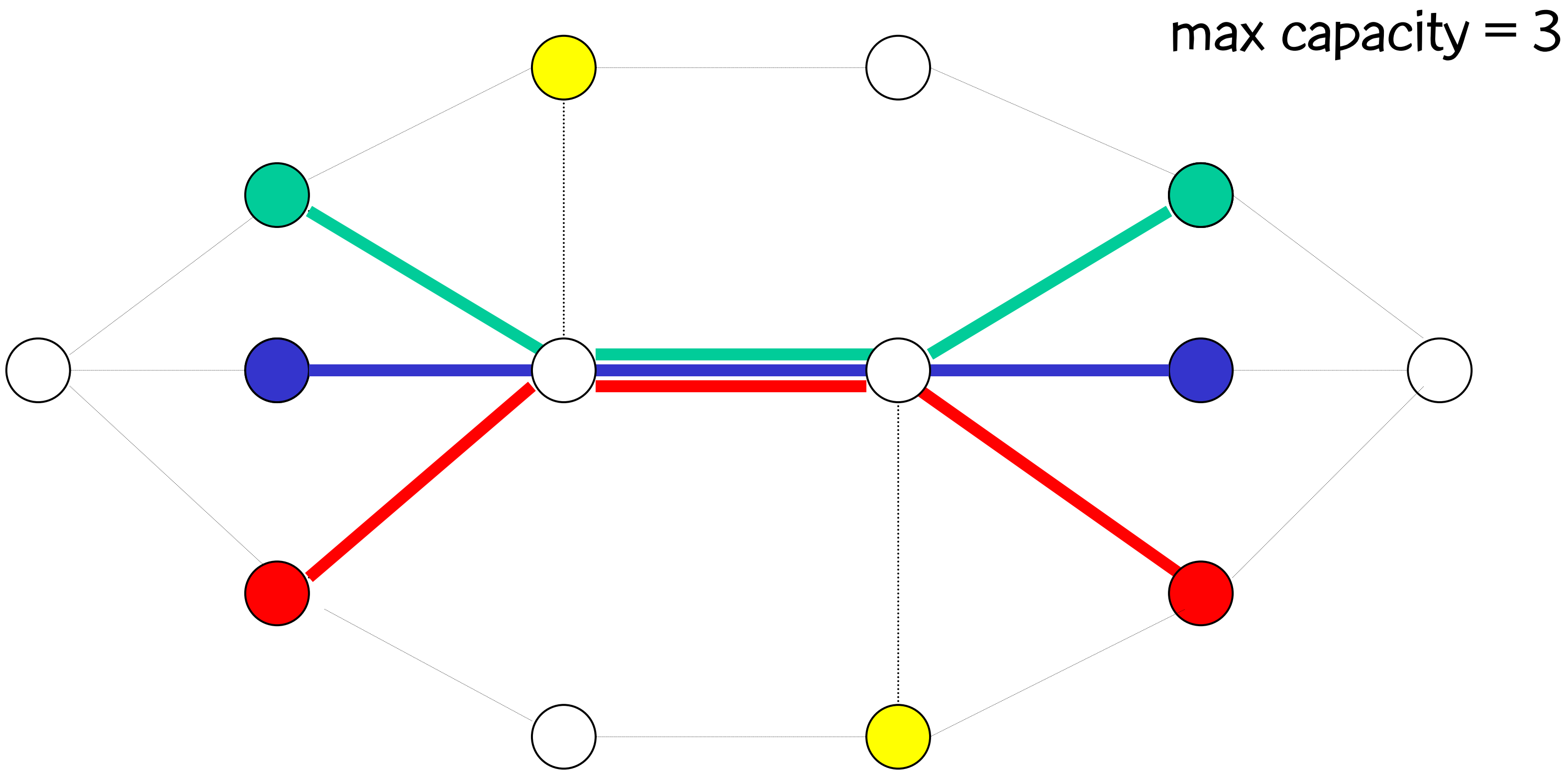
PVC routing: example



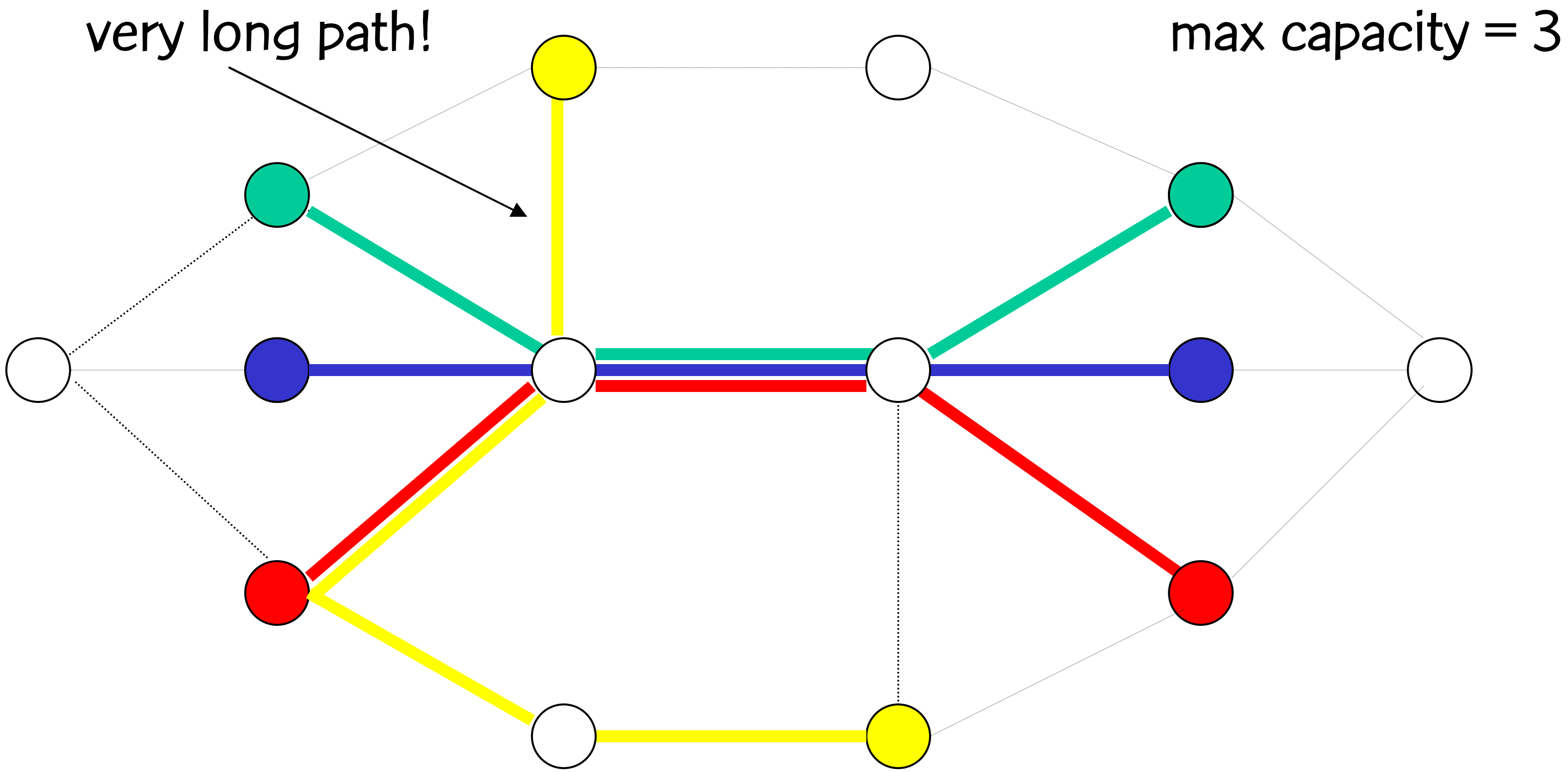
PVC routing: example



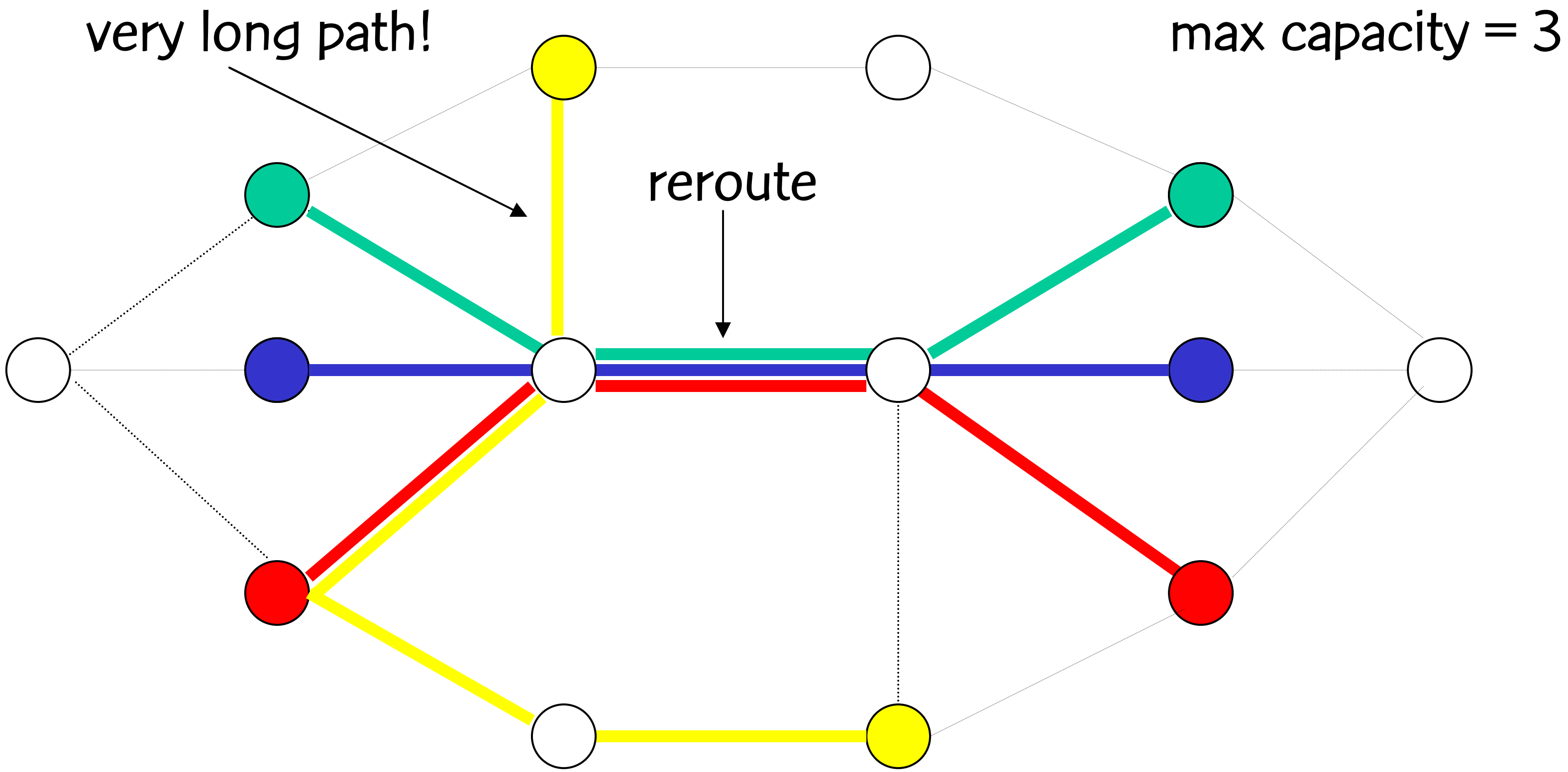
PVC routing: example



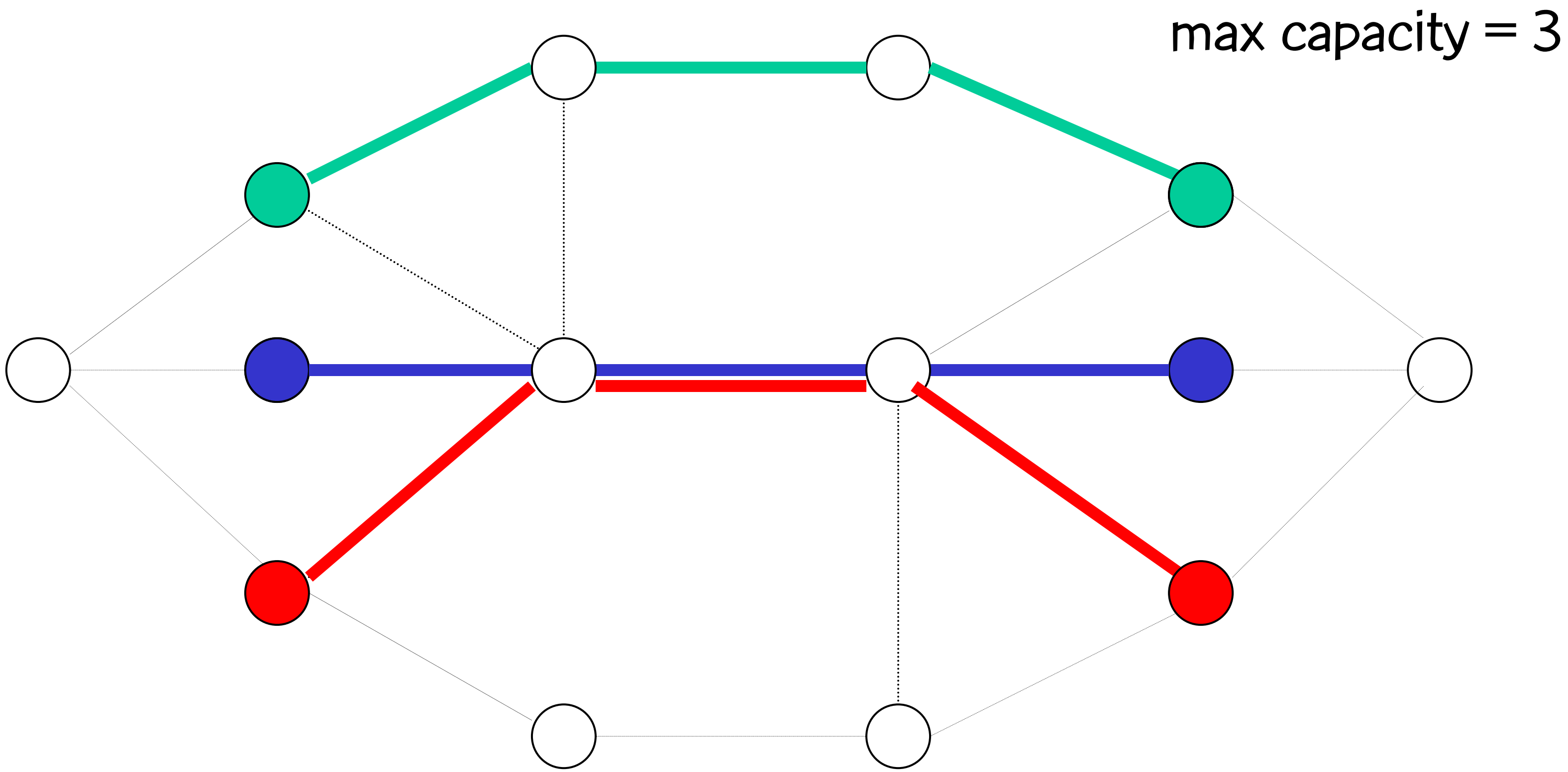
PVC routing: example



PVC routing: example



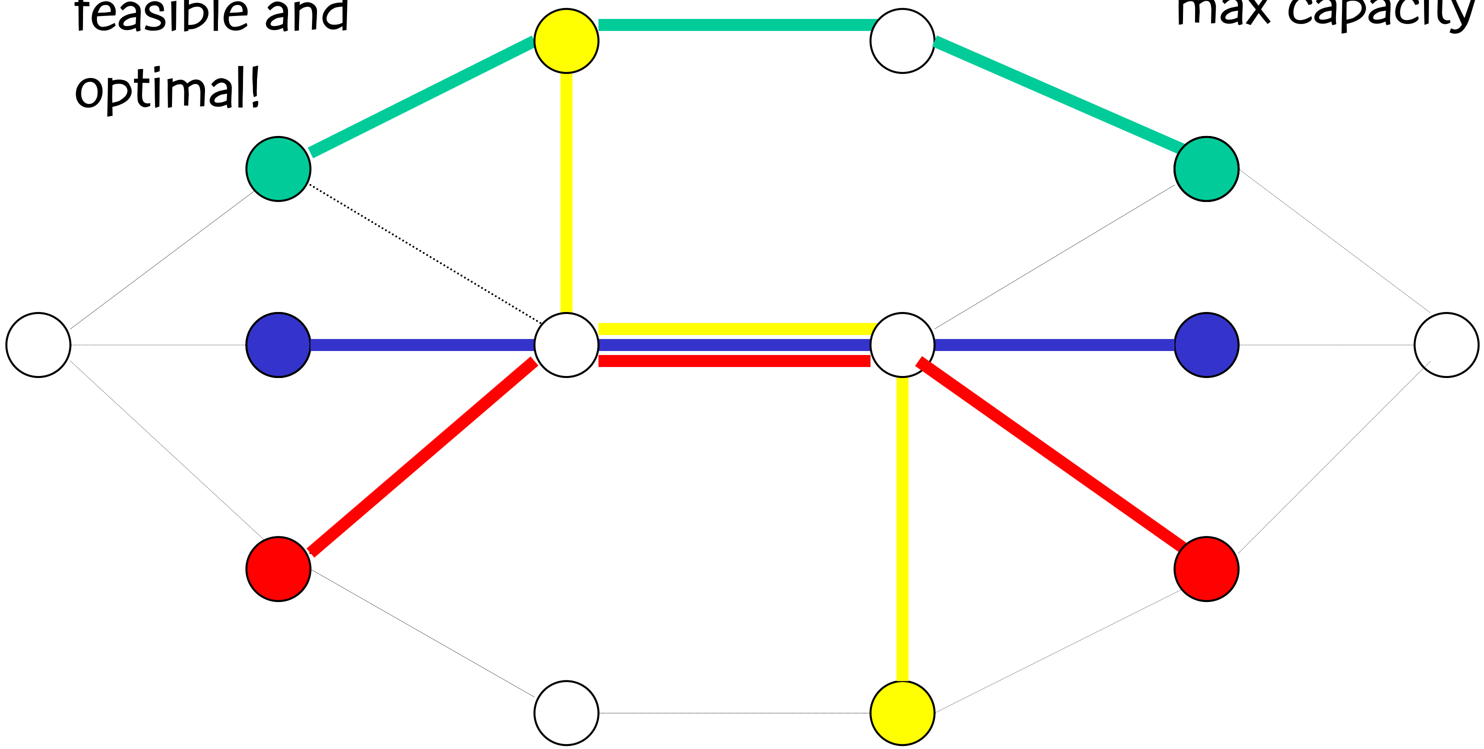
PVC routing: example



PVC routing: example

feasible and optimal!

max capacity = 3



Routing Frame Relay Permanent Virtual Circuits (PVC)

- one approach is to order PVCs and apply algorithm on FR switch to reroute
  - however, taking advantage of factors not considered by FR switch routing algorithm may lead to greater efficiency of network resource utilization
  - FR switch algorithm is typically fast since it is also used to reroute in case of switch or trunk failures
  - this can be traded off for improved network resource utilization when routing off-line

FR PVC Routing Problem

- given undirected FR network $G = (V, E)$, where
  - V denotes n backbone nodes (FR switches)
  - E denotes m trunks connecting backbone nodes
- for each trunk $e = (i,j)$ let
  - $b(e)$ be the bandwidth (max kbits/sec rate) of trunk e
  - $c(e)$ be the max number of PVCs that can be routed on trunk e
  - $d(e)$ be the propagation and hopping delay associated with trunk e

FR PVC Routing Problem

- list of demands (or commodities $K = \{1,...,p\}$) is defined by
  - origin - destination pairs
  - $r(p)$ - effective bandwidth requirement (forward, backward, overbooking) for PVC p
- objective is to minimize
  - delays
  - network load unbalance
- subject to
  - technological constraints

FR PVC Routing (bandwidth packing) Problem

- route for PVC (o, d) is
  - sequence of adjacent trunks
  - first trunk originates in node o
  - last trunk terminates in node d
- set of routing assignments is feasible if for all trunks e
  - total PVC bandwidth requirements routed on e does exceed $b(e)$
  - number of PVCs routed on e is not greater than $c(e)$

Mathematical programming formulation

$$\min \phi(x) = \sum_{(i,j)\in E, i<j} \phi_{i,j}(x^1_{i,j},\ldots,x^p_{i,j},x^1_{j,i},\ldots,x^k_{j,i})$$

subject to

$$\sum_{k\in K} r_k(x^k_{i,j} + x^k_{j,i}) \le b_{i,j}, \quad \forall\ (i,j)\in E, i<j$$

$$\sum_{k\in K} (x^k_{i,j} + x^k_{j,i}) \le c_{i,j}, \quad \forall\ (i,j)\in E, i<j$$

$$\sum_{(i,j)\in E} x^k_{i,j} - \sum_{(i,j)\in E} x^k_{j,i} = \begin{cases} 1, \text{ if } i\in V \text{ is source for } k\in K \\ -1, \text{ if } i\in V \text{ is destination for } k\in K \\ 0, \text{ otherwise} \end{cases}$$

$$x^k_{i,j} \in \{0,1\}, \quad \forall (i,j)\in E, \forall k\in K.$$

$x^k_{i,j}$ = 1, iff trunk (i,j) is used to route PVC k.

Cost function

- Linear combination of
  - delay component
  - load balancing component

- Delay component: $$d_{i,j}\sum_{k\in K}\rho_k(x_{i,j}^k + x_{j,i}^k)$$

Cost function: Load balancing component

- We use the measure of Fortz & Thorup (2000) to compute congestion:

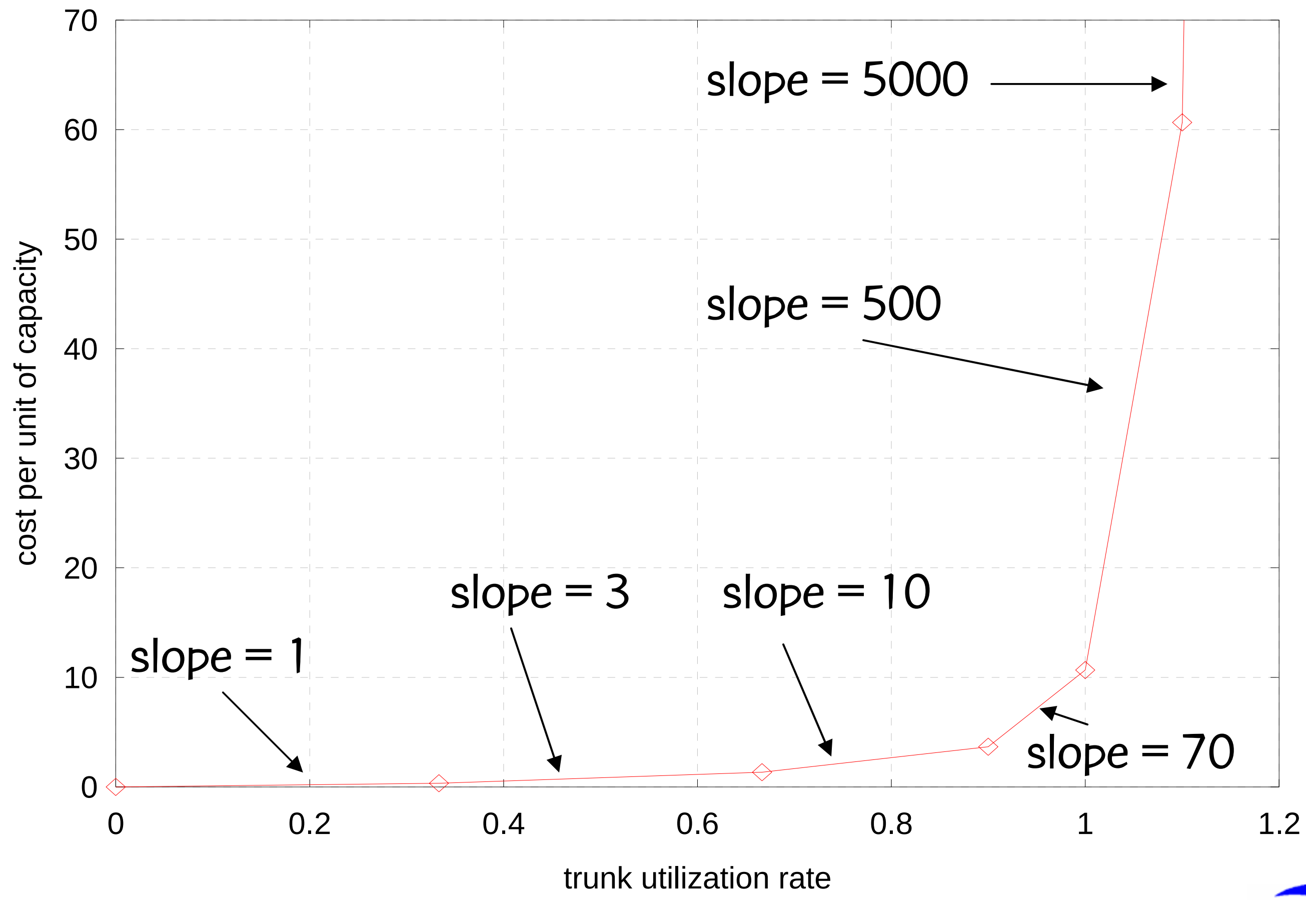
$$\Phi = \Phi_1(l_1) + \Phi_2(l_2) + \ldots + \Phi_{|E|}(l_{|E|})$$

where l_a is the load on link $e \in E$,

$\Phi_e(l_e)$ is piecewise linear and convex,

$\Phi_e(0) = 0$, for all $e \in E$.

Piecewise linear and convex $\Phi_e(l_e)$ link congestion measure



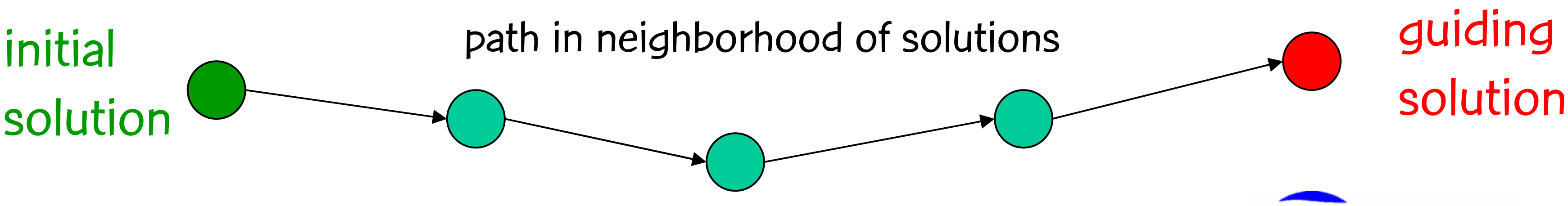
$(l_a \div c_a)$

Solution method

- GRASP
  - Construct by choosing unrouted pair, biasing in favor of high bandwidth requirement. Use shortest path routing using as edge distance the incremental cost associated with routing r_k additional units of demand on edge (i, j).
  - Local search: for each PVC $k \in K$, remove r_k units of flow from each edge in its current route, compute incremental edge weights, and reroute.

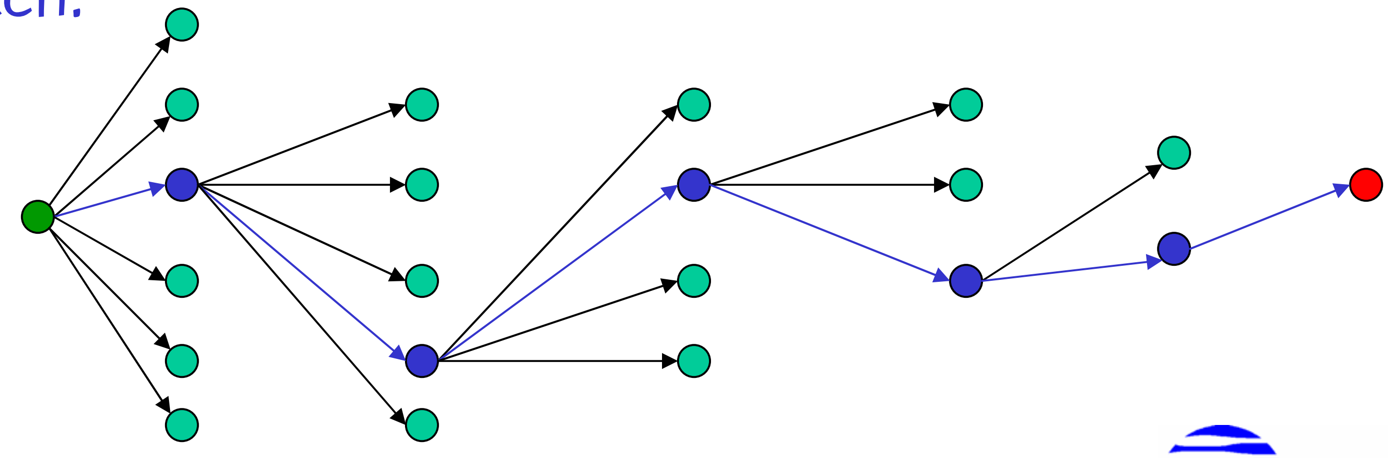
Path relinking

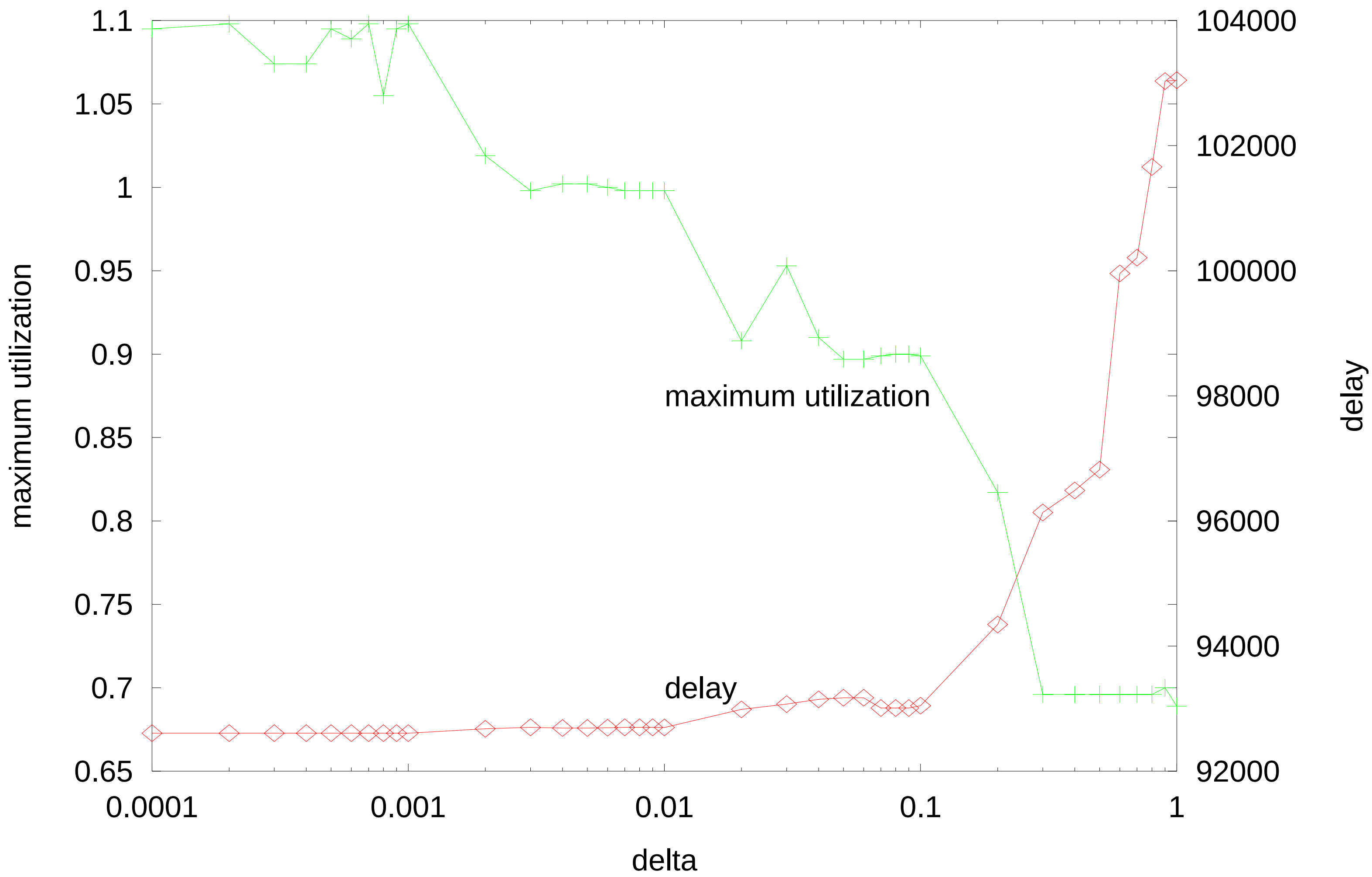
- Introduced in context of tabu search in Glover & Laguna (1997):
  - Approach to integrate intensification & diversification in search.
- Consists in exploring trajectories that connect high quality solutions.



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 analyzed and best move is taken.





maximum utilization
delay
delta
maximum utilization
delay
1.1
1.05
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
104000
102000
100000
98000
96000
94000
92000
0.0001
0.001
0.01
0.1
1

AT&T

Summary

- modem location for internet service provider
  - metaheuristic & LP-based upper bounds
- design local access network
  - metaheuristic & LP-based lower bounds
- SONET ring network design
  - LP-based solution procedure
- offline routing of permanent virtual circuits
  - Metaheuristic
- ▶ OSPF routing
  - Metaheuristic & lower bounds

Internet traffic engineering

- Internet traffic has been doubling each year [Coffman & Odlyzko, 2001]
- In the1995-96 period, there was a doubling of traffic each three months!
  - Web browsers were introduced.
- Increasingly heavy traffic (due to video, voice, etc.) will raise the requirements of the Internet of tomorrow.

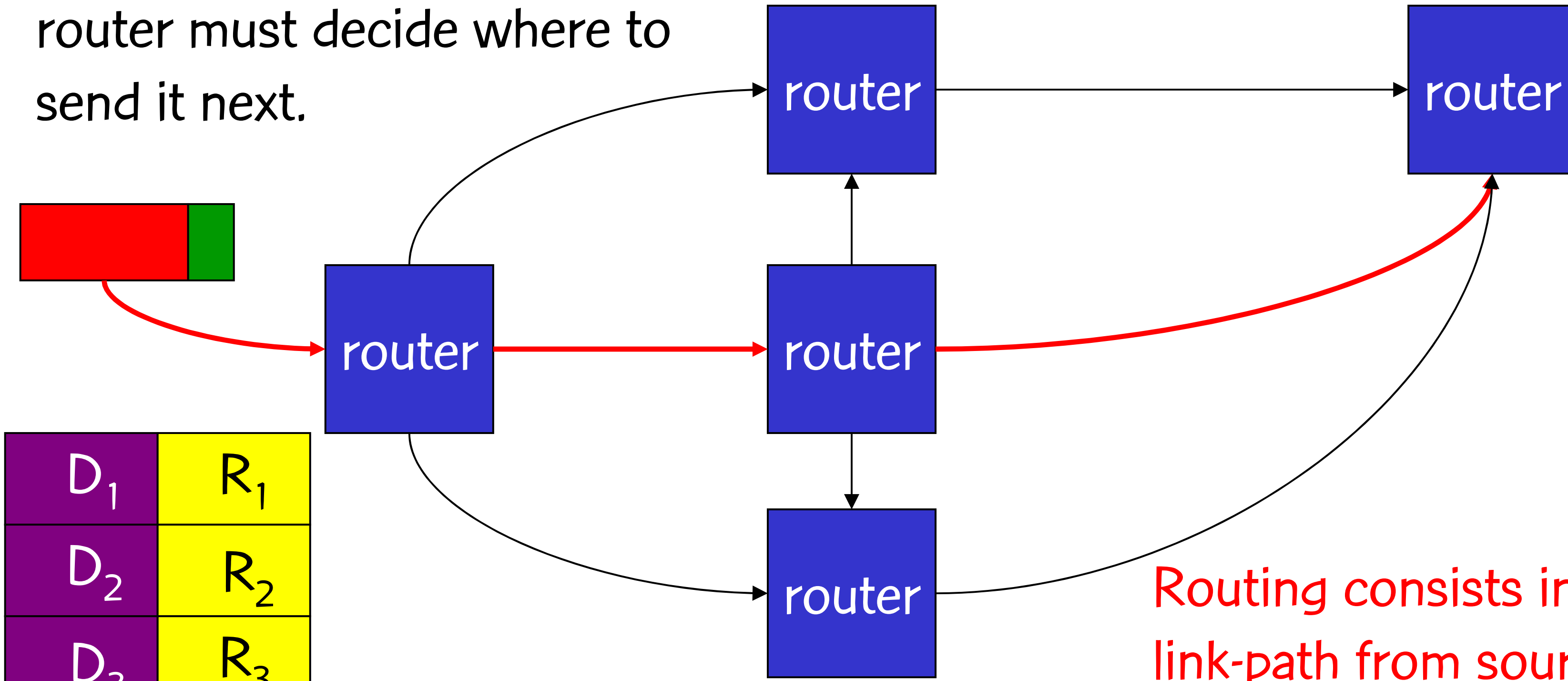
Internet traffic engineering

- 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.



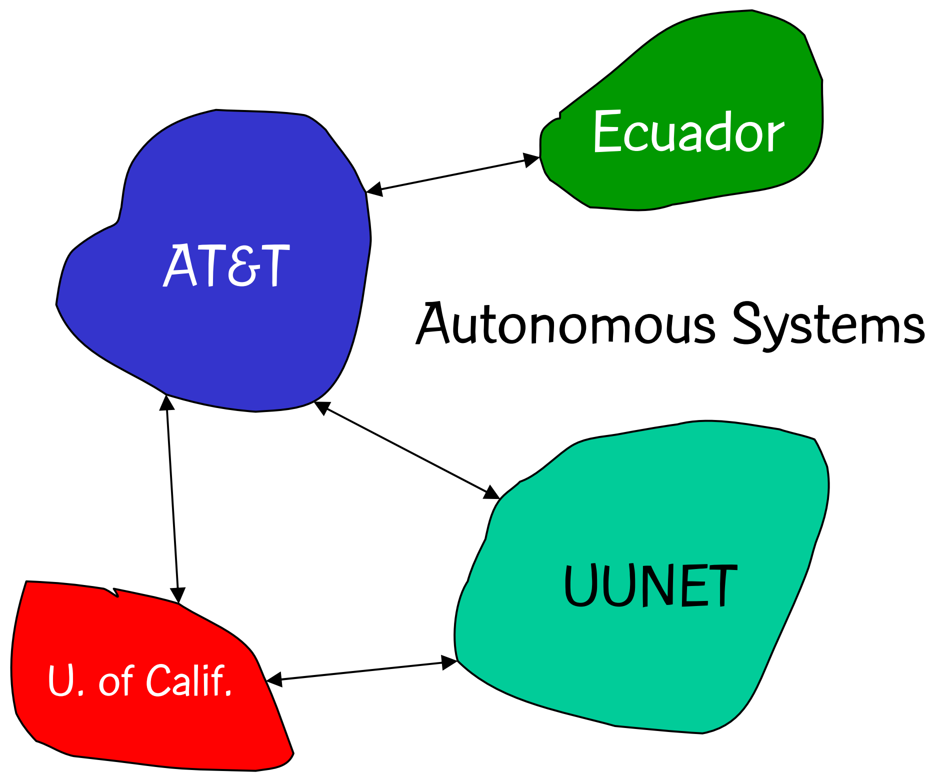
D_1	R_1
D_2	R_2
D_3	R_3
D_4	R_4

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.



AT&T

OSPF routing

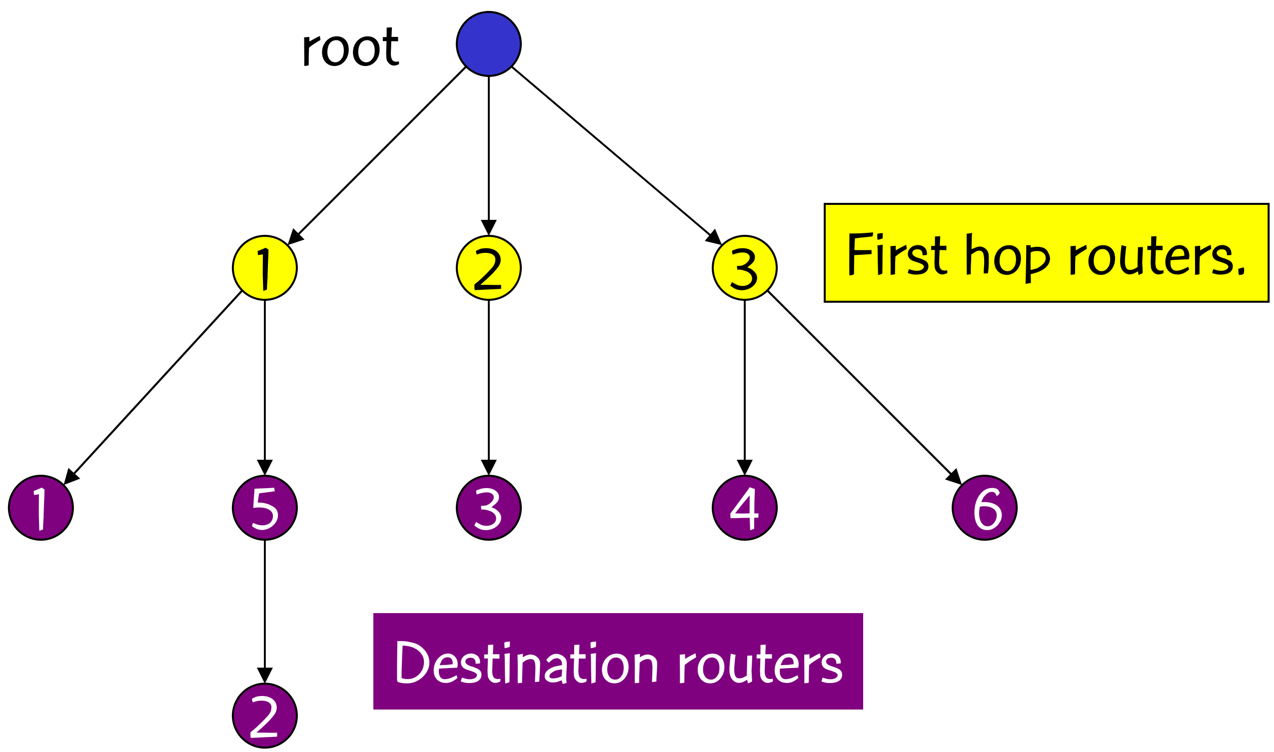
- Assign an integer weight $\in [1, w_{max}]$ to each link in AS. In general, $w_{max} = 65535 = 2^{16}-1$.
- Each router computes tree of shortest weight paths to all other routers in the AS, with itself as the root, using Dijkstra's algorithm.

OSPF routing

Routing table 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

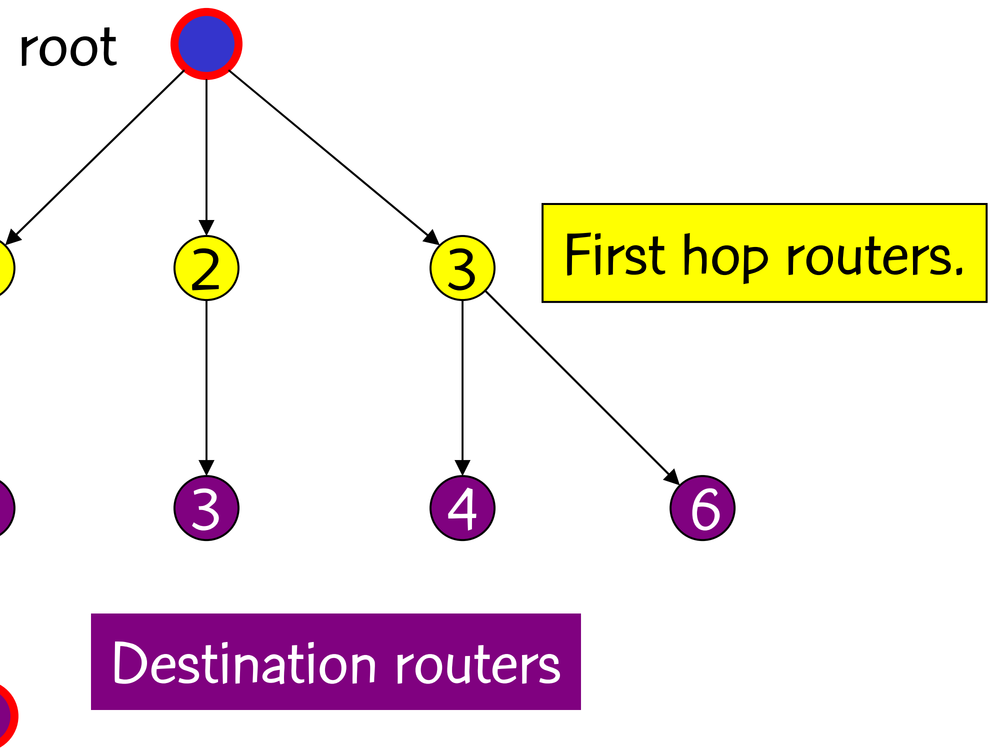


OSPF routing

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

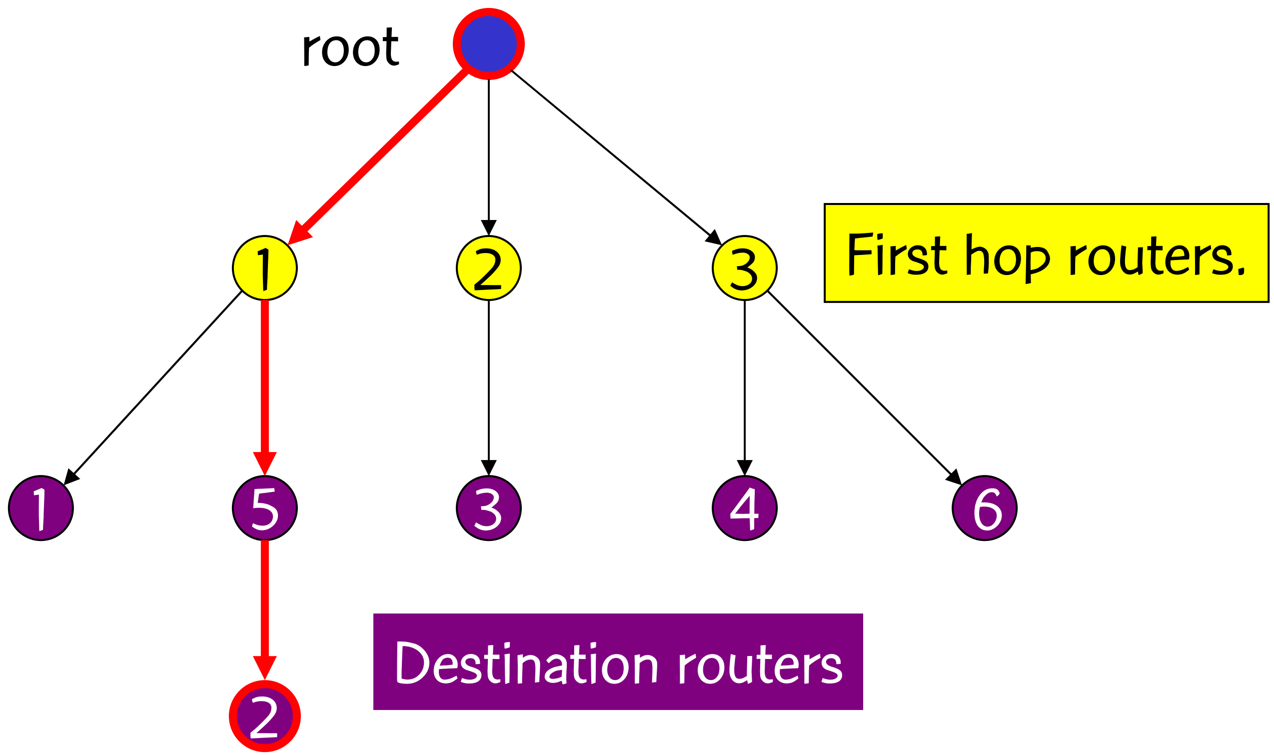


OSPF routing

Routing table

D_1	R_1
D_2	R_1
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.

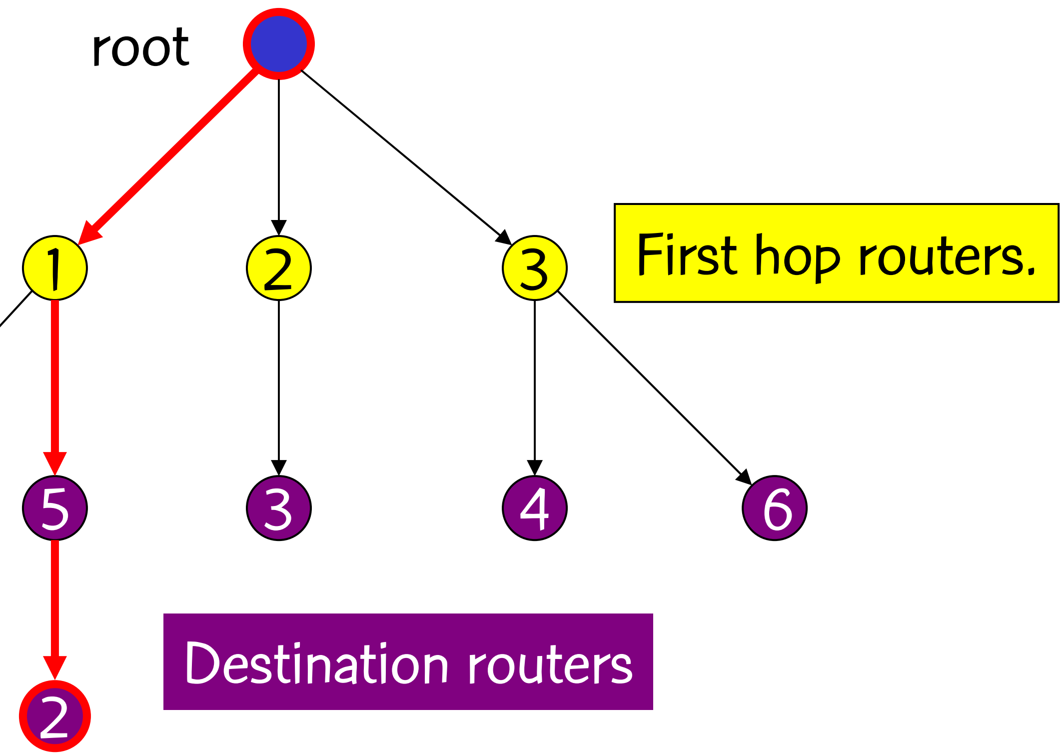


OSPF routing

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

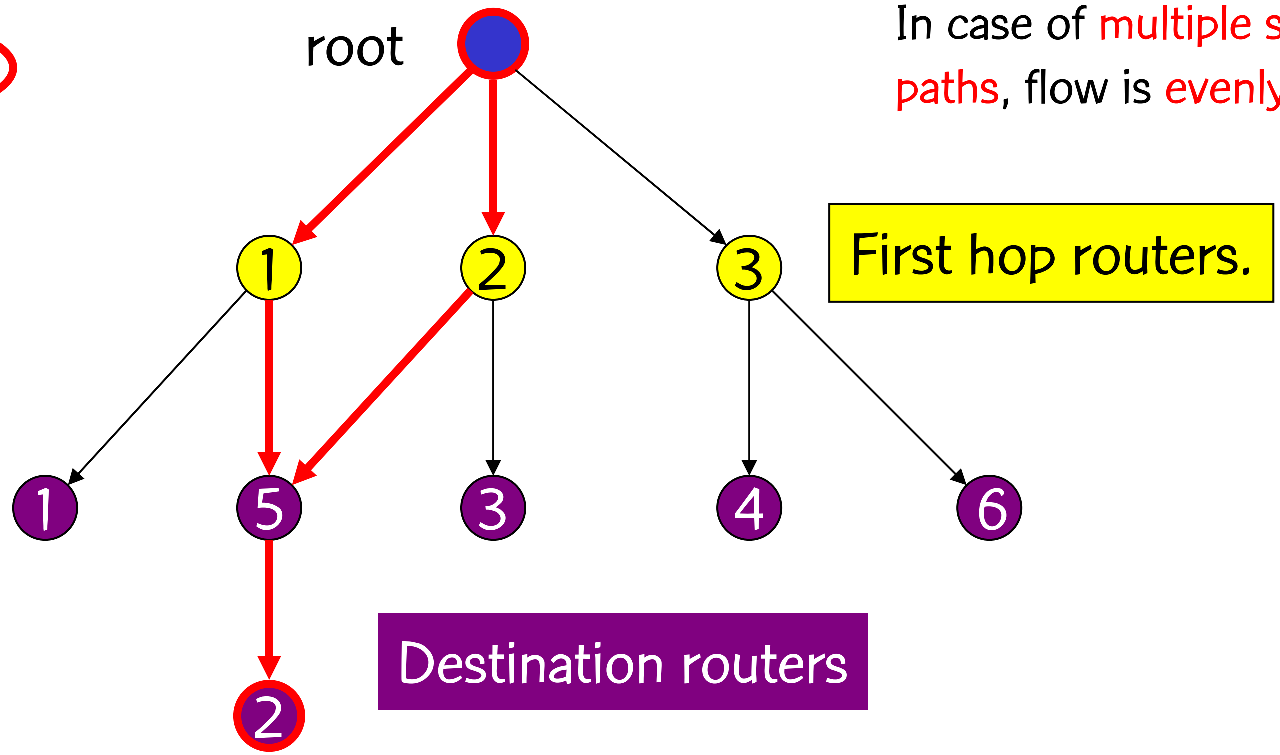


OSPF routing

Routing table

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:

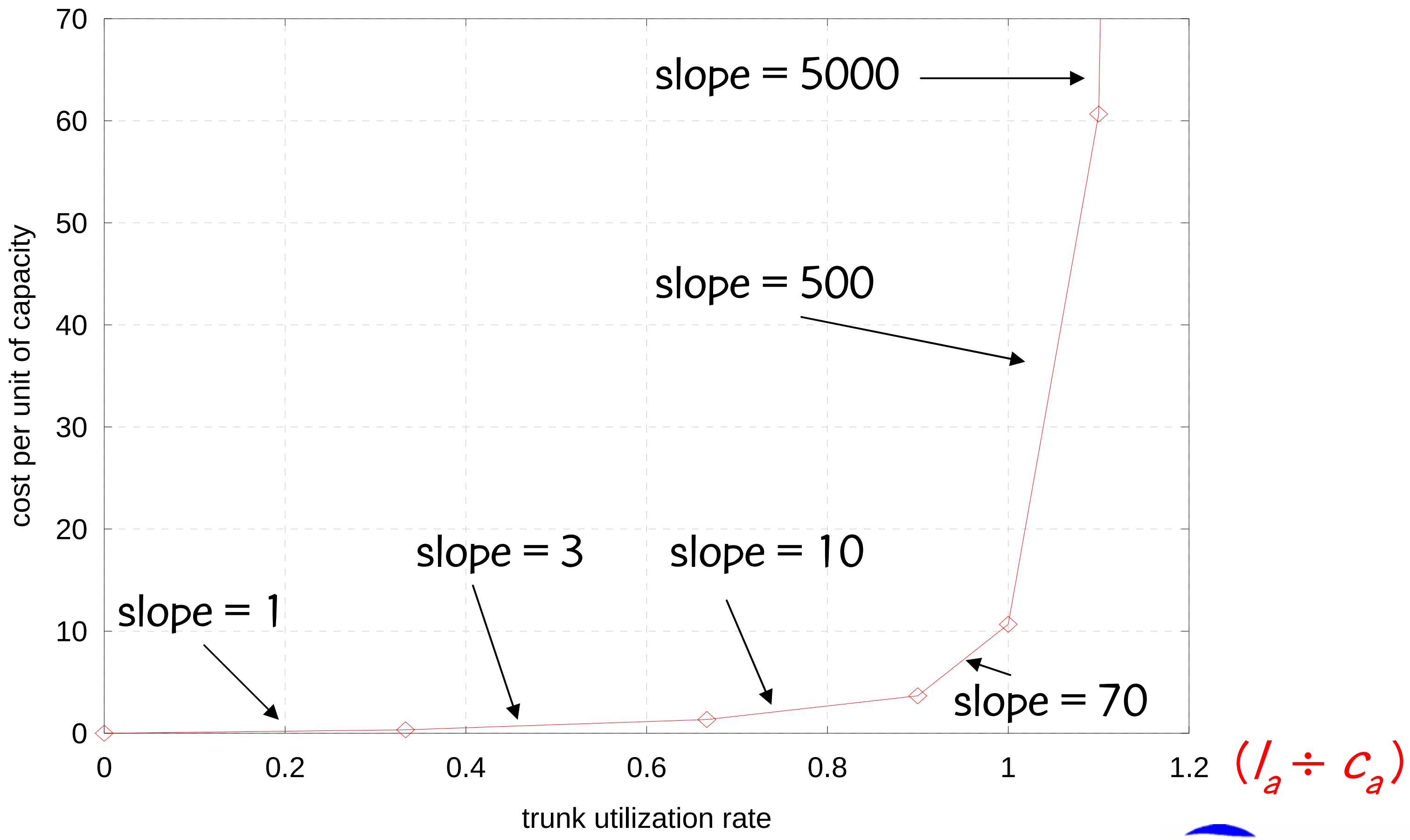
$$\Phi = \Phi_1(l_1) + \Phi_2(l_2) + \ldots + \Phi_{|A|}(l_{|A|})$$

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 $\Phi_a(l_a)$ 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 $\Phi_{uncap} = \sum_{(s,t) \in N \times N} d_{s,t} h_{s,t}$

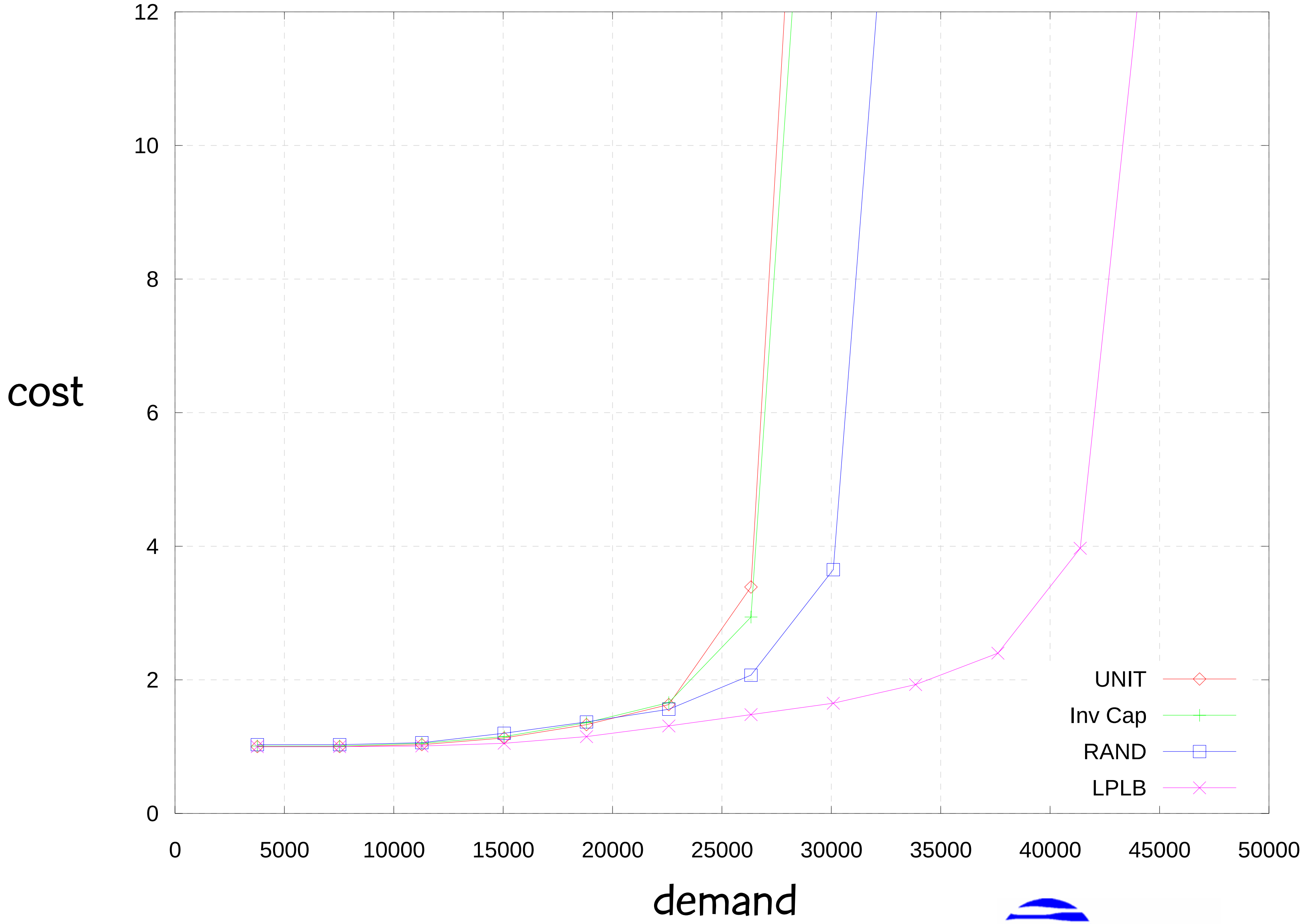
Total load if all traffic goes along unit weight shortest paths.

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

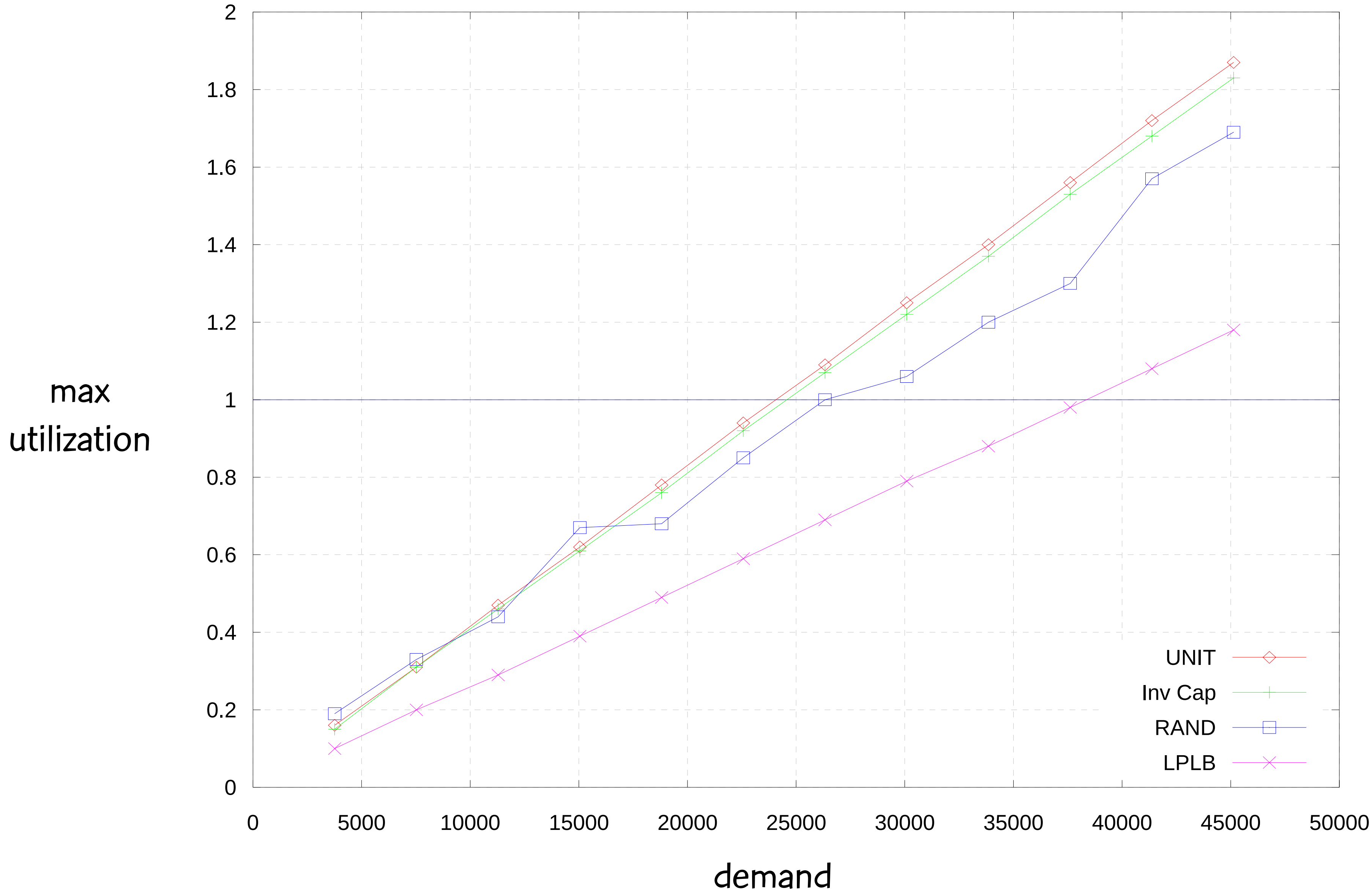
Normalized cost $\Phi^* = \Phi / \Phi_{uncap}$

- Fortz & Thorup (2000) show that:
- $1 \leq \Phi^*_{opt} \leq \Phi^*_{optOSPF} \leq \Phi^*_{unitOSPF} < 5000$
- 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 $\Phi^* \geq 10\frac{2}{3}$

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

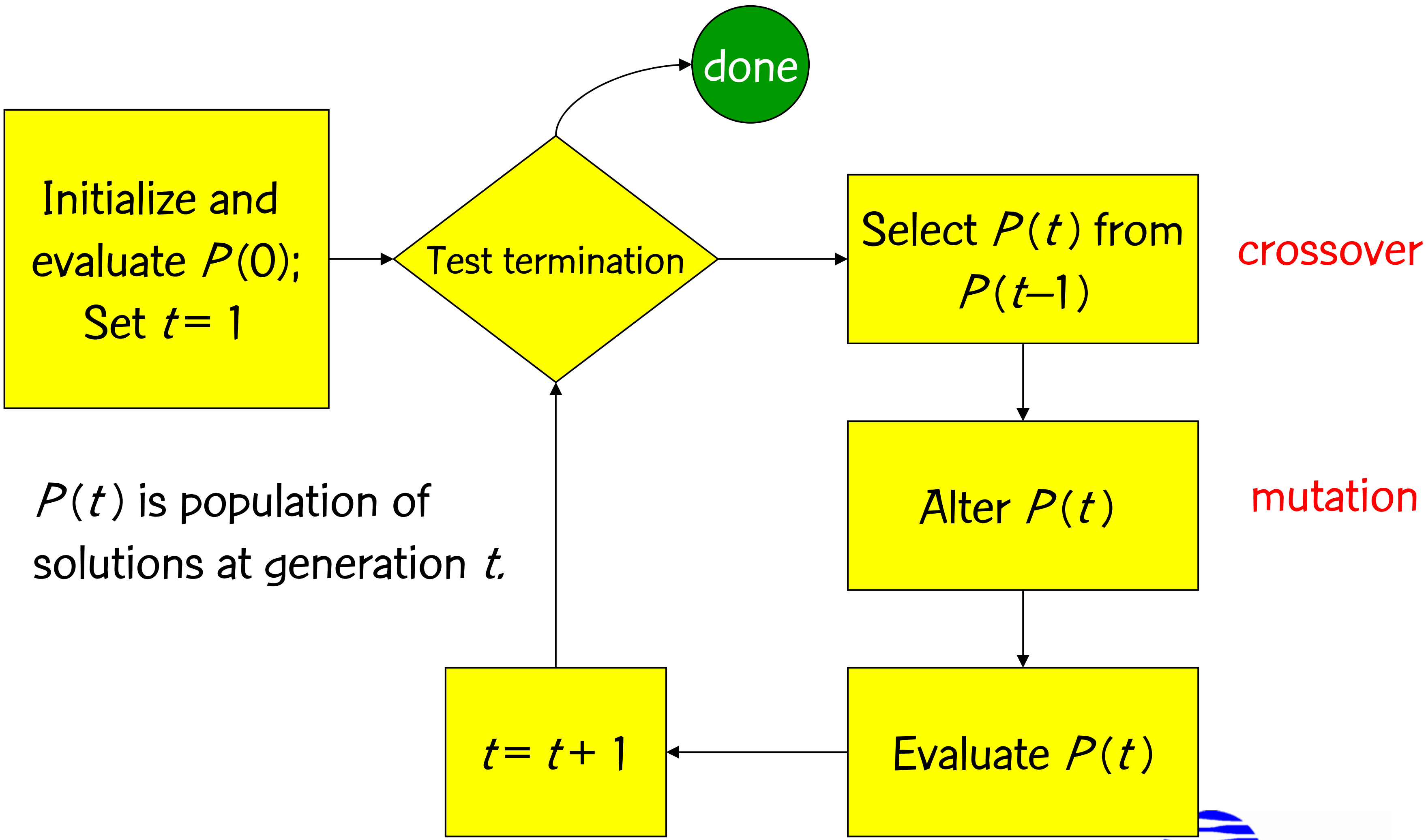


AT&T Worldnet backbone network (90 routers, 274 links)



AT&T

Genetic algorithms



$P(t)$ is population of solutions at generation t.

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.

AT&T

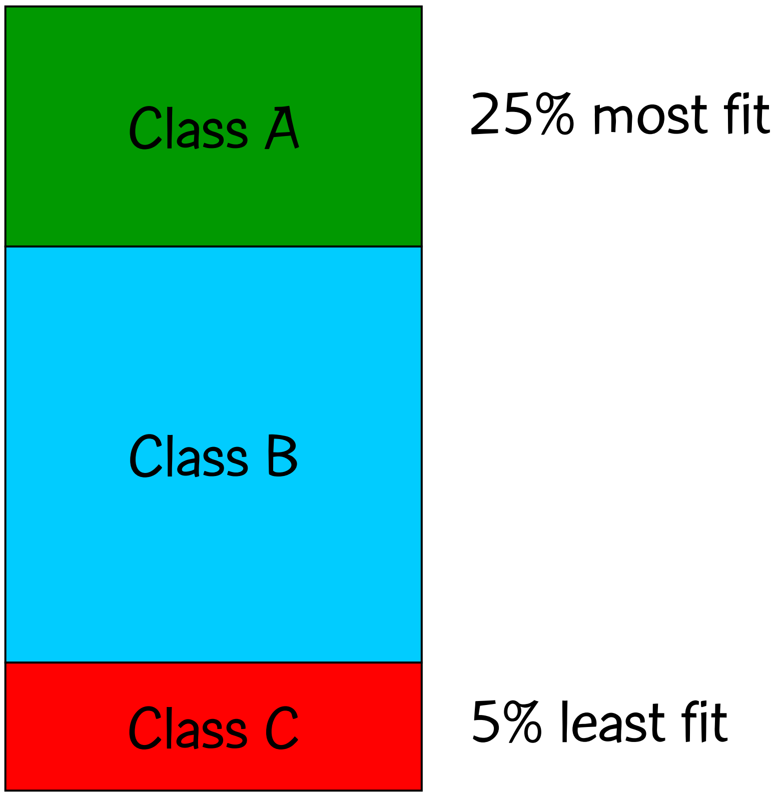
Initial population

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

Solution evaluation

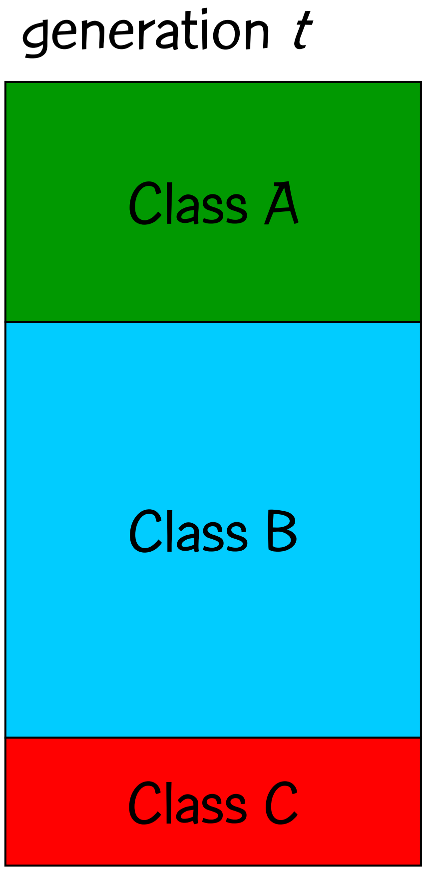
- For each demand pair (s,t), route using OSPF, computing demand pair loads $l_a^{s,t}$ on each link $a \in A$.
- Add up demand pair loads on each link $a \in A$, yielding total load l_a on link.
- Compute link congestion cost $\Phi_a(l_a)$ for each link $a \in A$.
- Add up costs: $\Phi = \Phi_1(l_1)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

Population partitioning

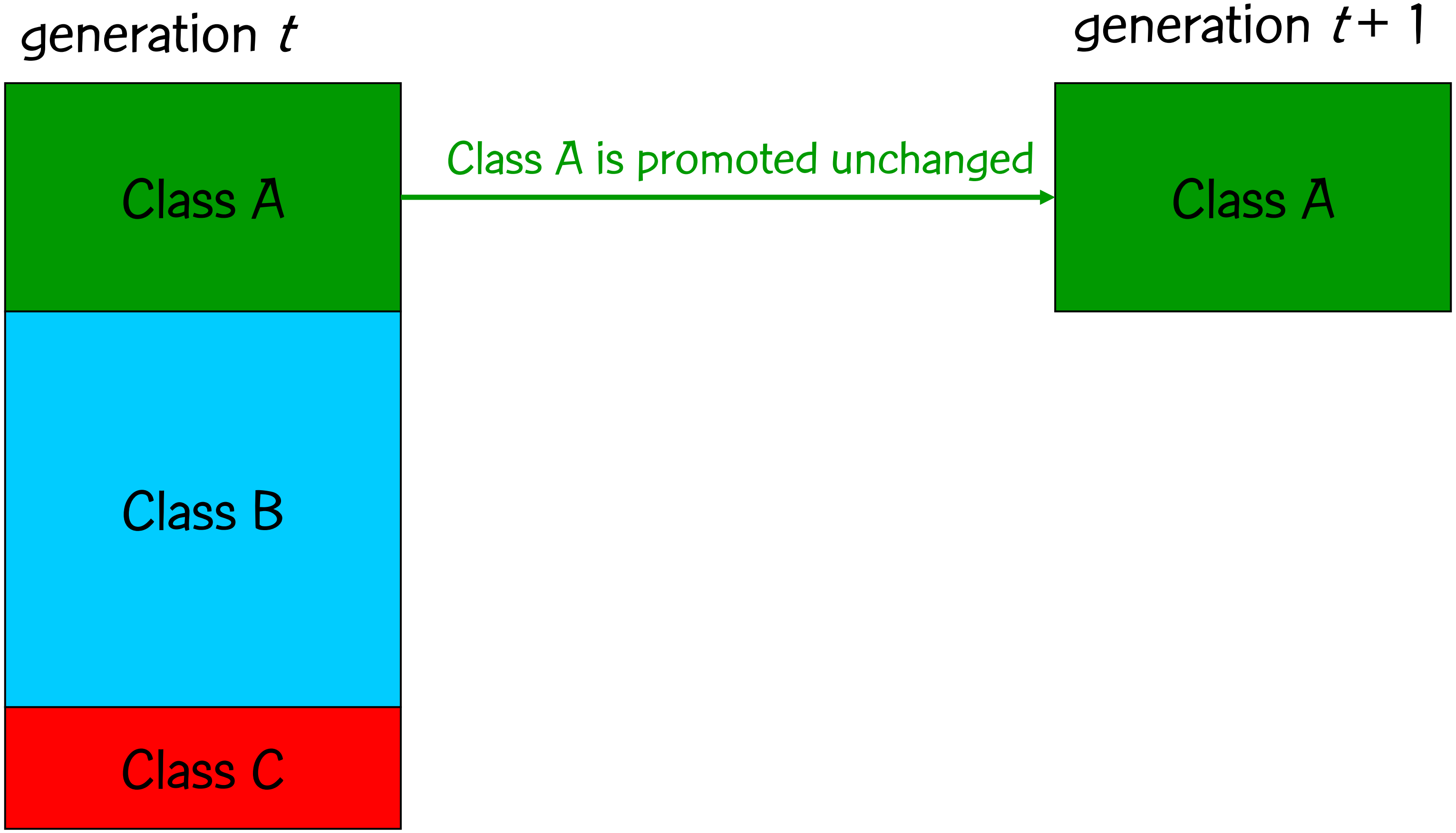


Population is sorted according to solution value Φ and solutions are classified into three categories.

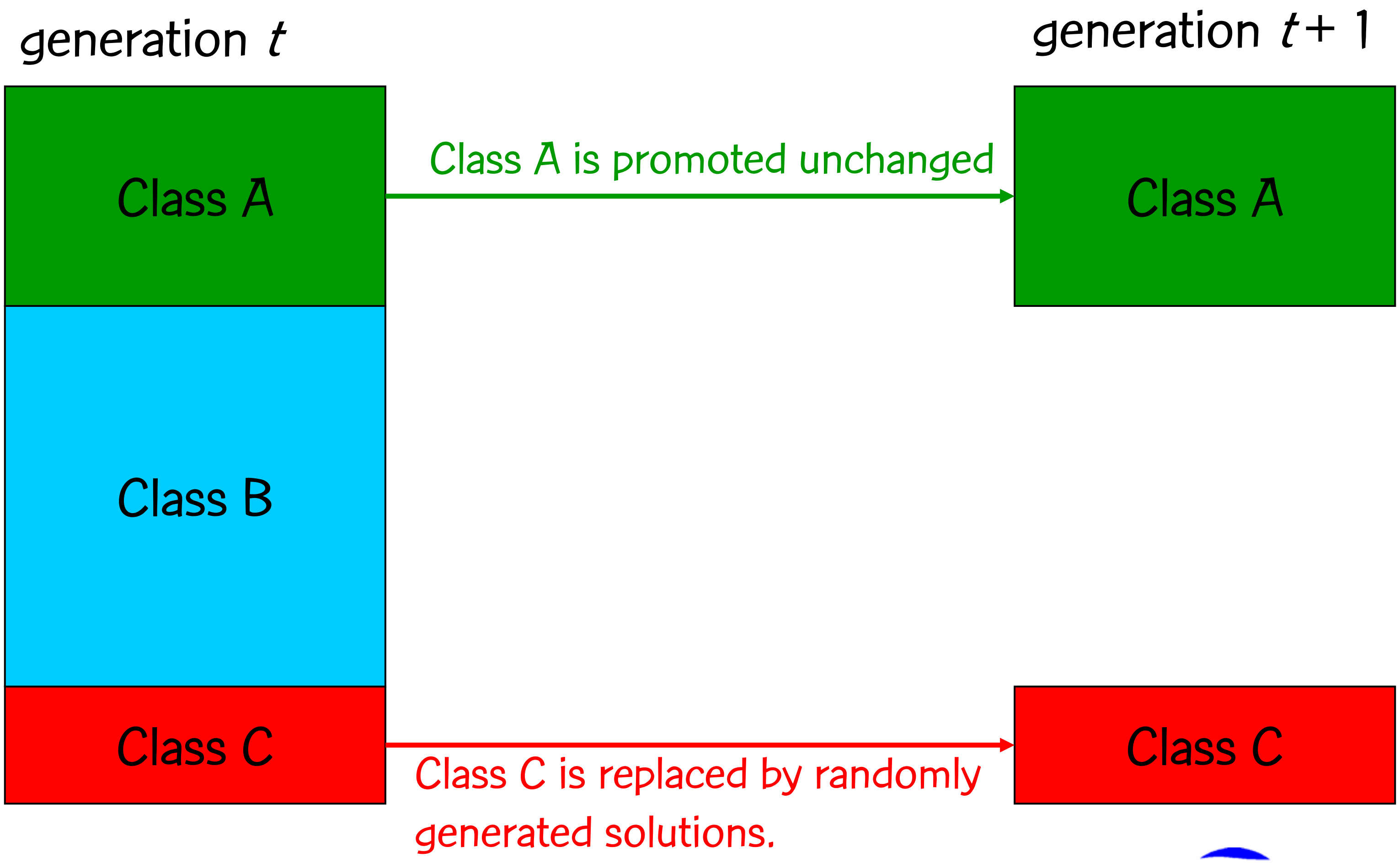
Population dynamics



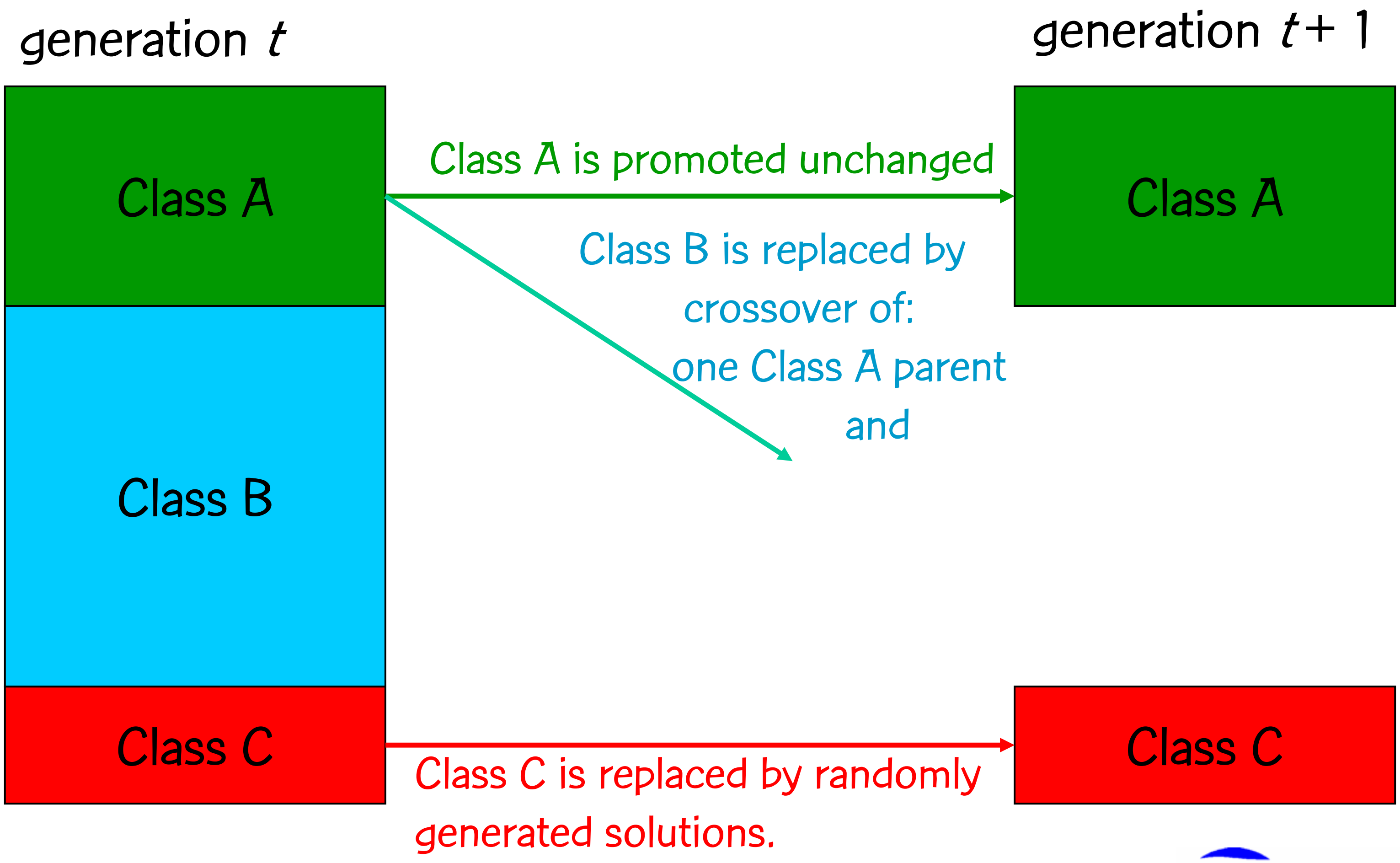
Population dynamics



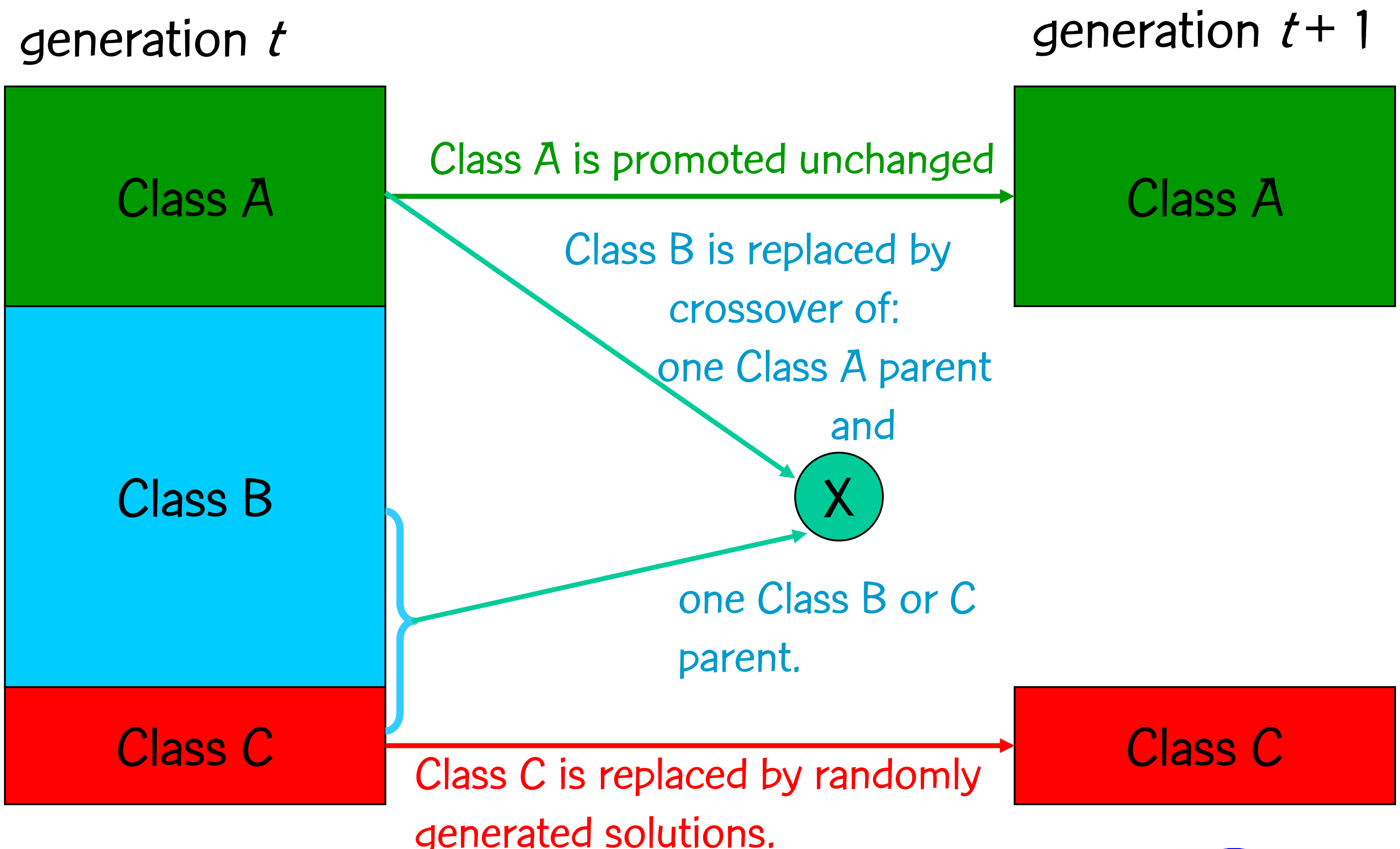
Population dynamics



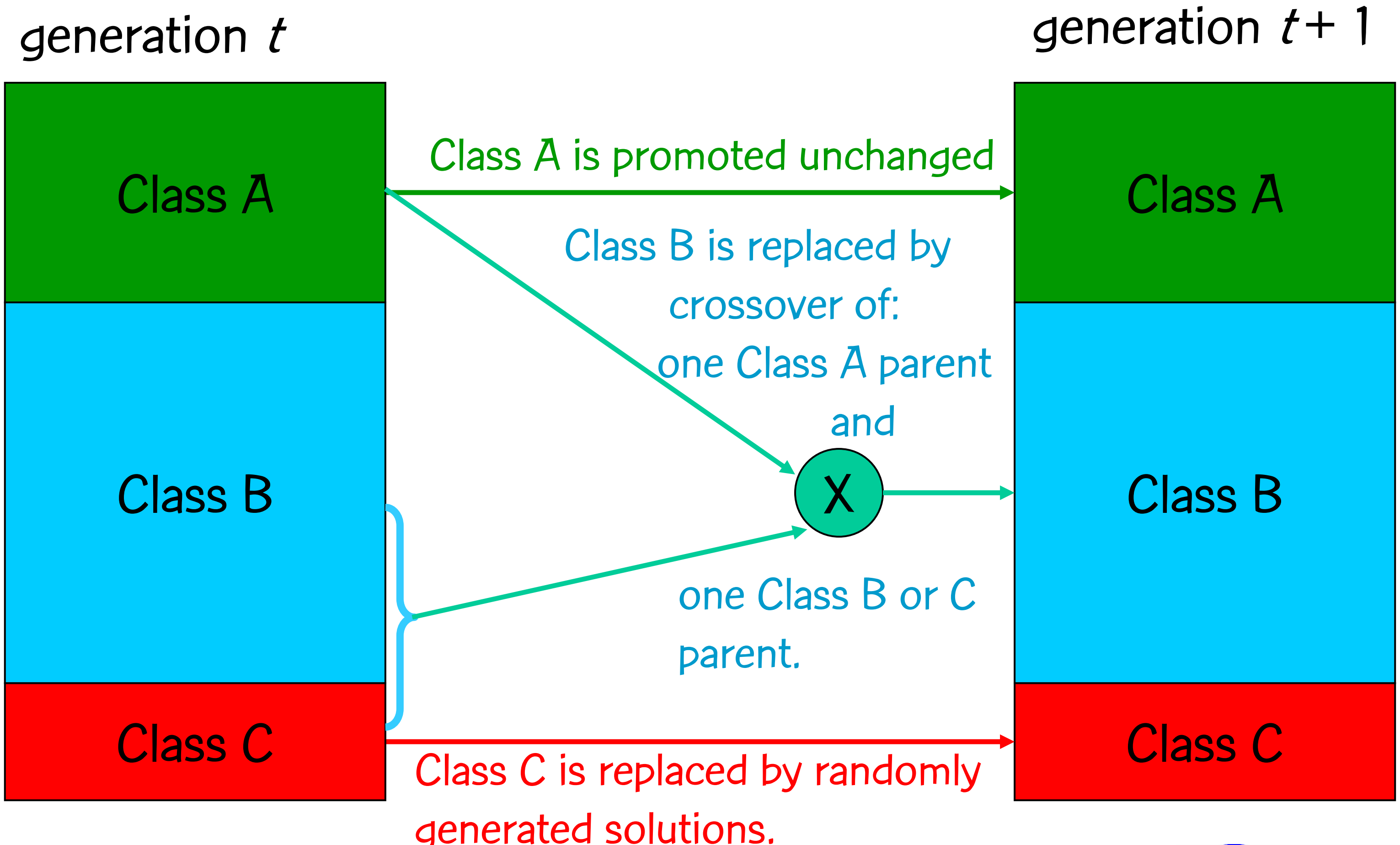
Population dynamics



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)

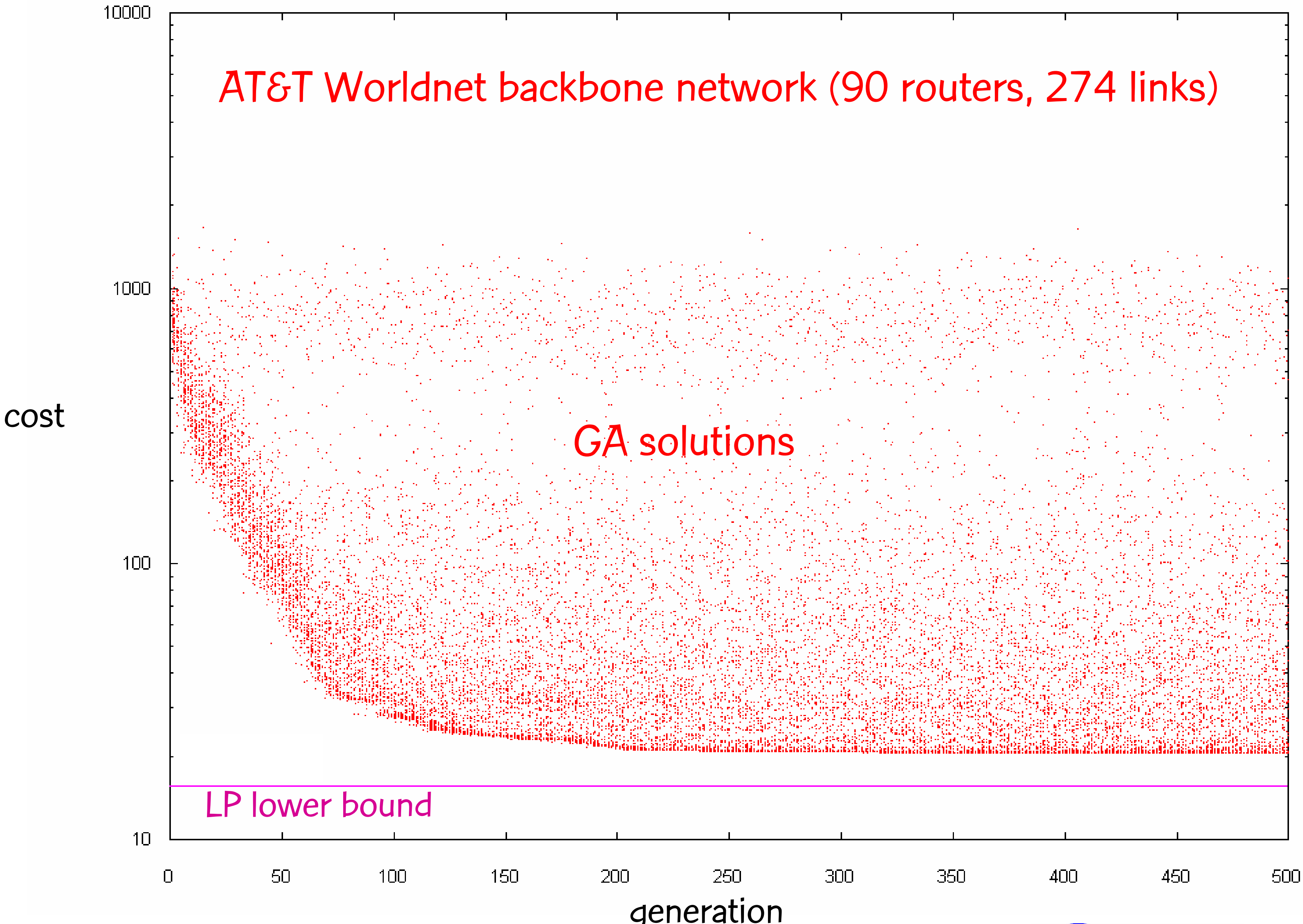
Crossover combines elite parent p_1 with non-elite parent p_2 to produce child c:

With small probability child has single gene mutation.

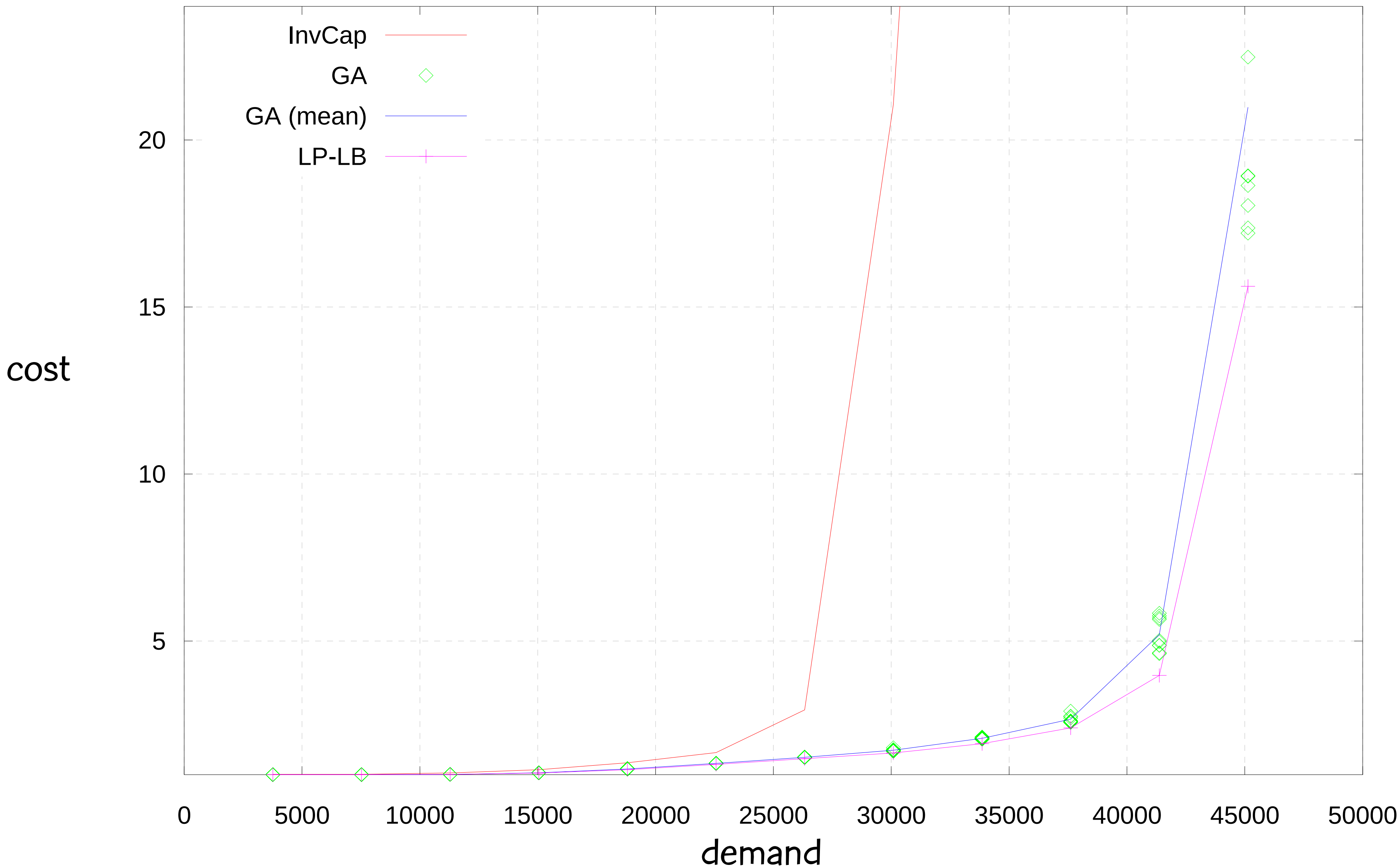
Child is more likely to inherit gene of elite parent.

```
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] = p_1[i]
    else c[i] = p_2[i]
end
```

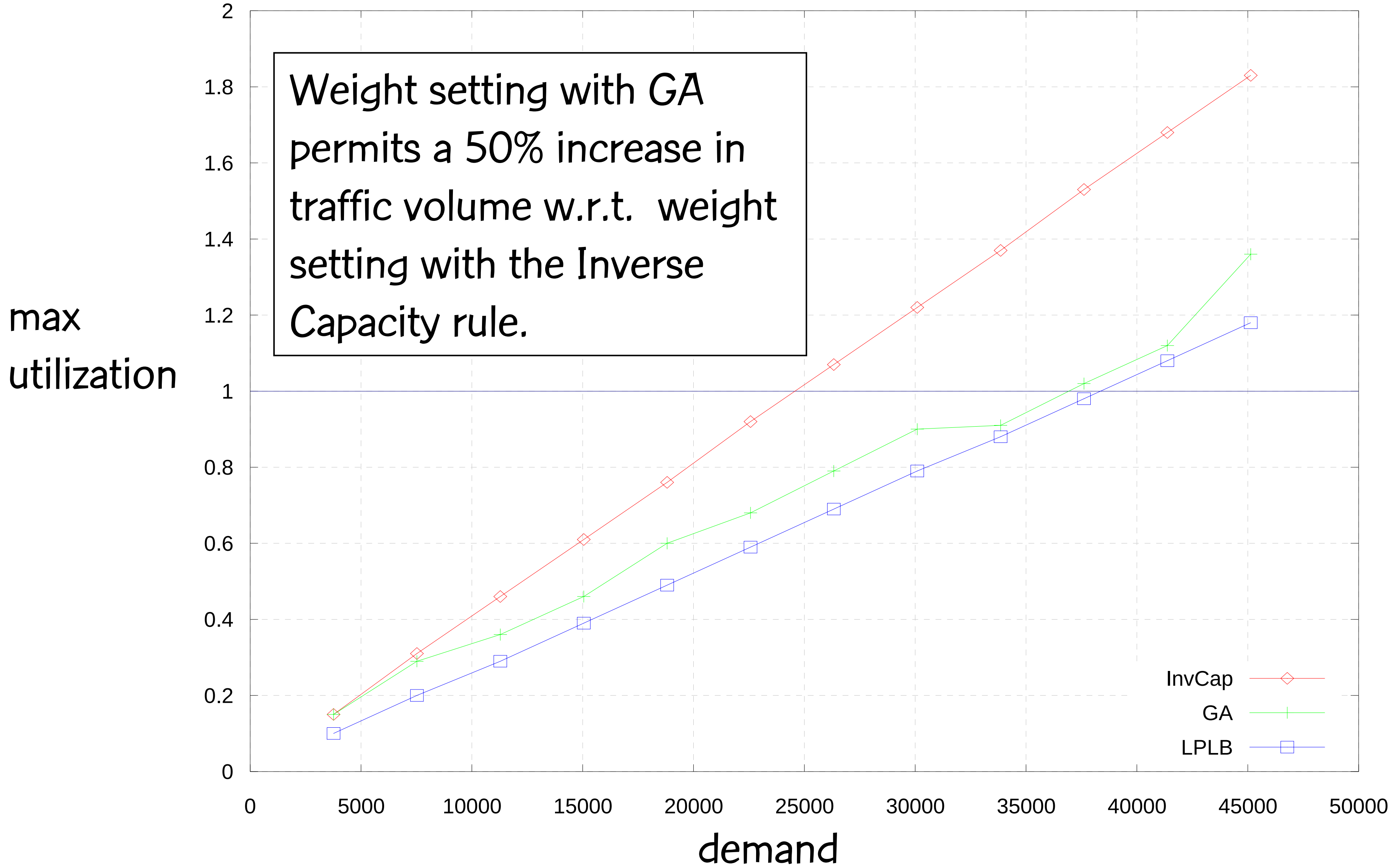

AT&T


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)

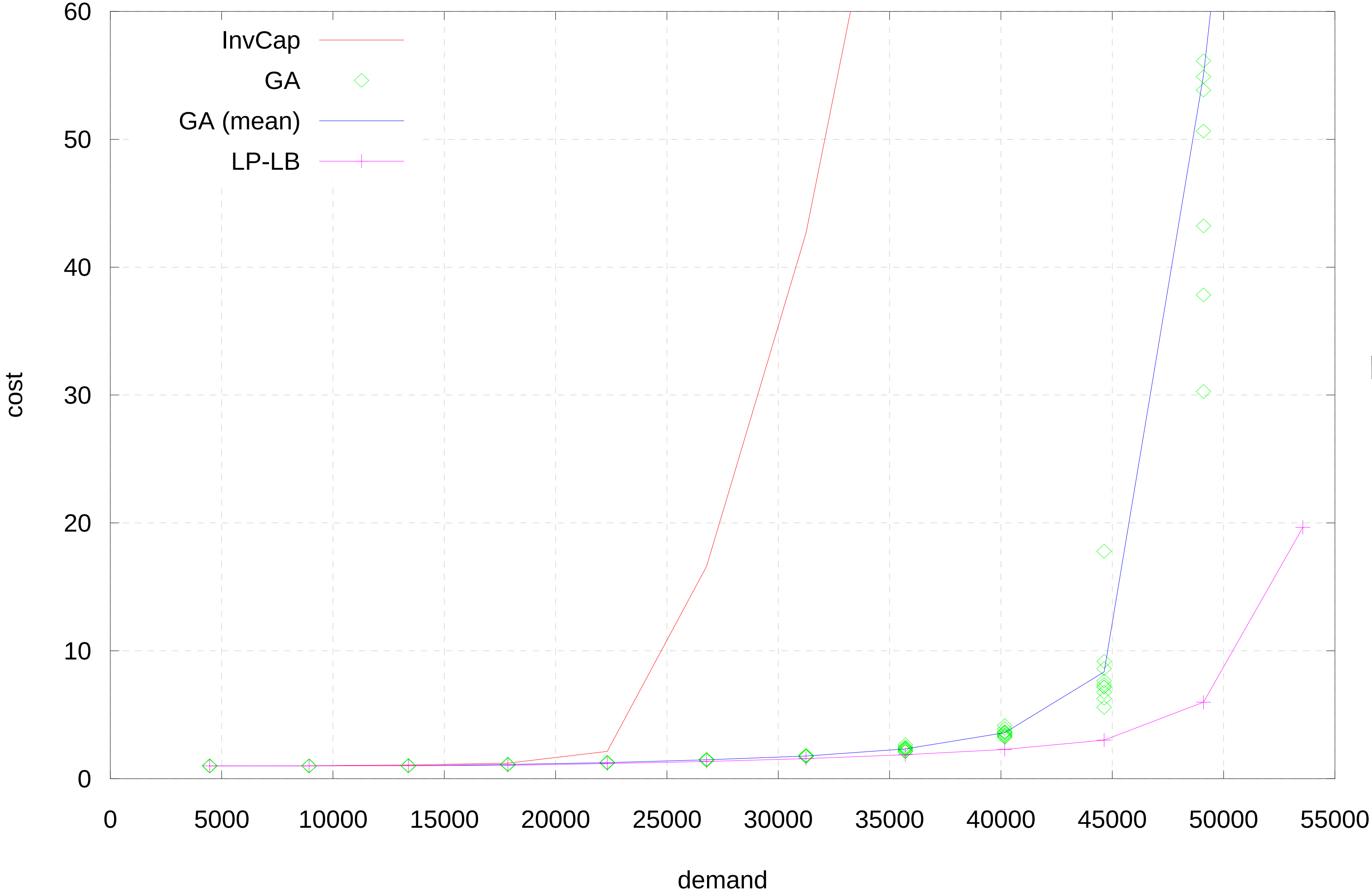


AT&T Worldnet backbone network (90 routers, 274 links)



AT&T

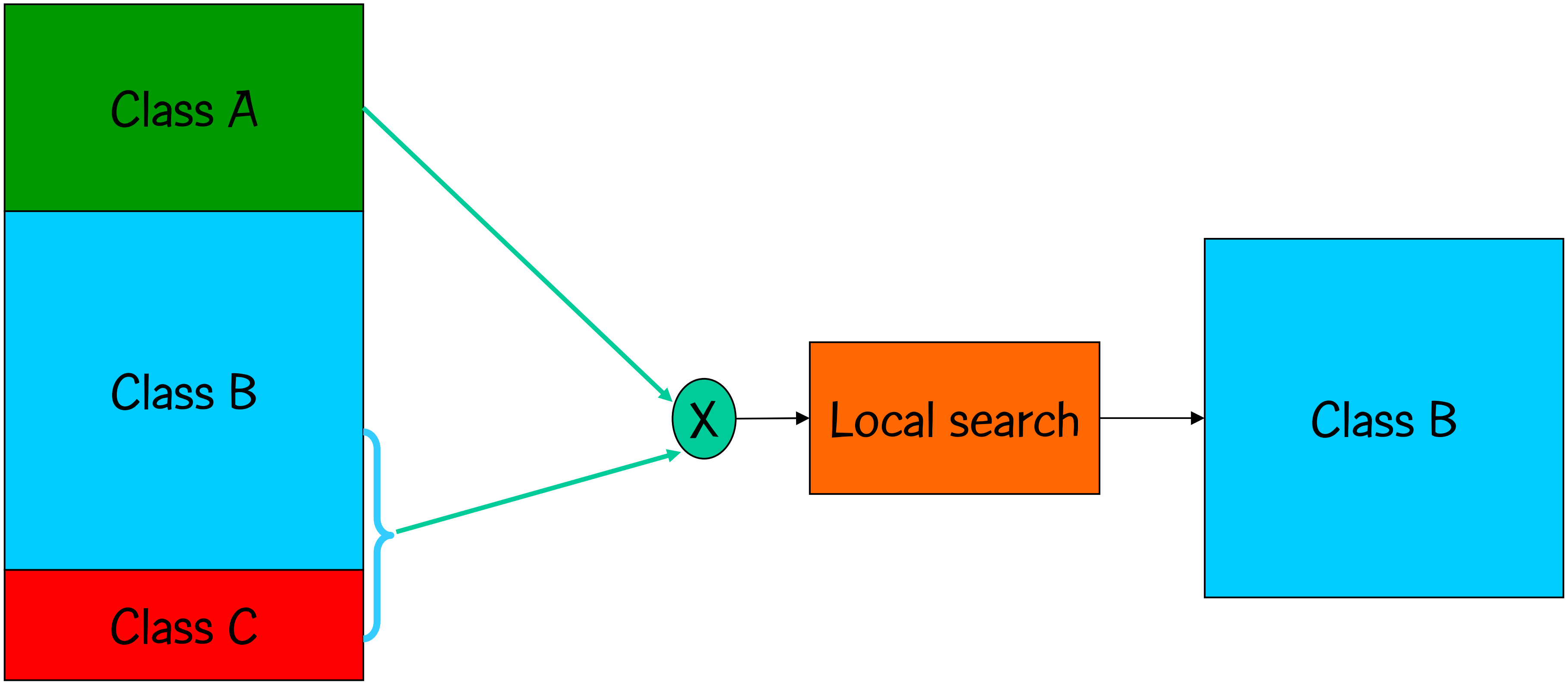
Rand50a: random graph with 50 nodes and 245 arcs.



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.

AT&T

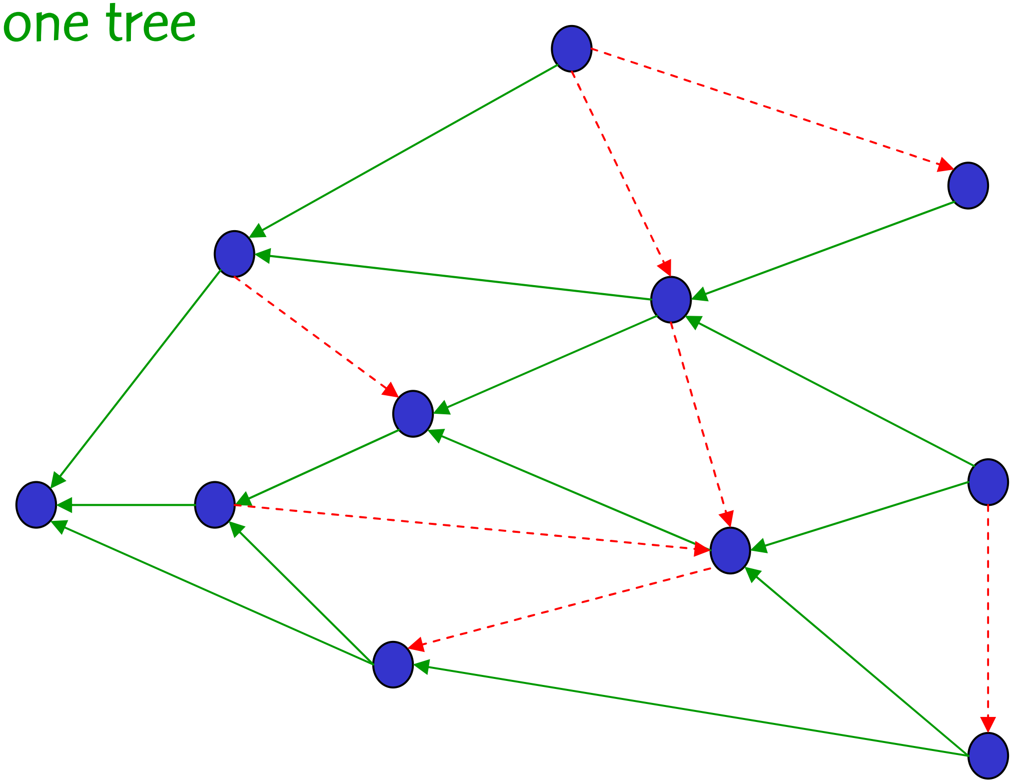
Dynamic shortest path

- 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.

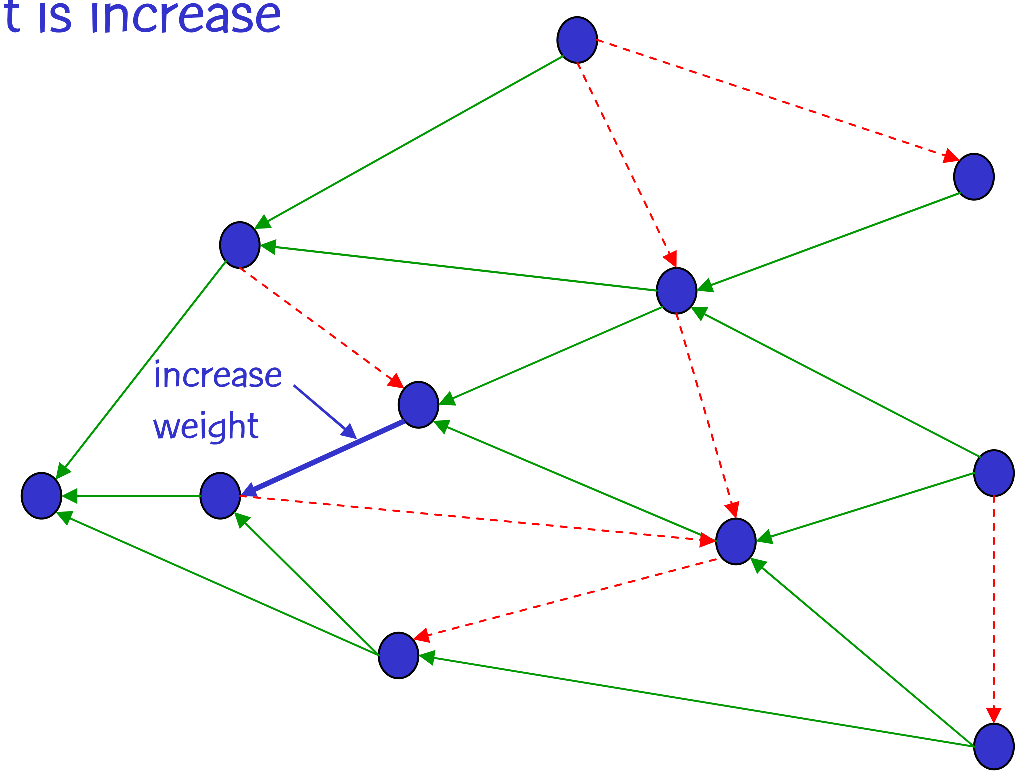
Dynamic shortest path

Consider one tree at a time.



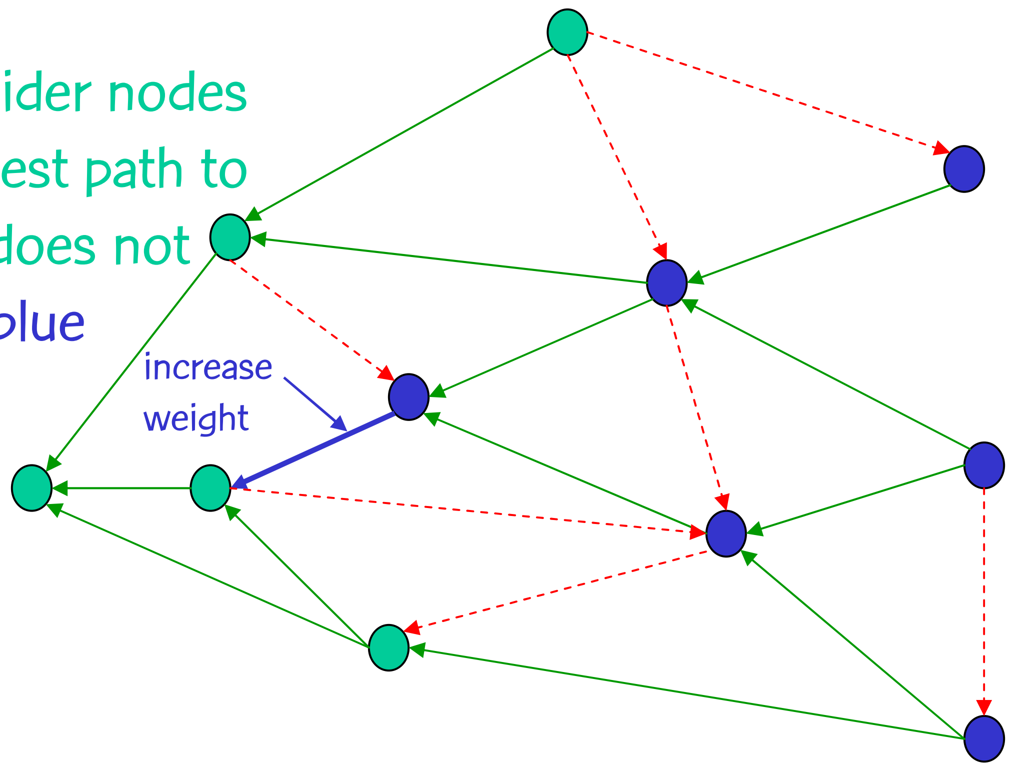
Dynamic shortest path

Arc weight is increase by 1.

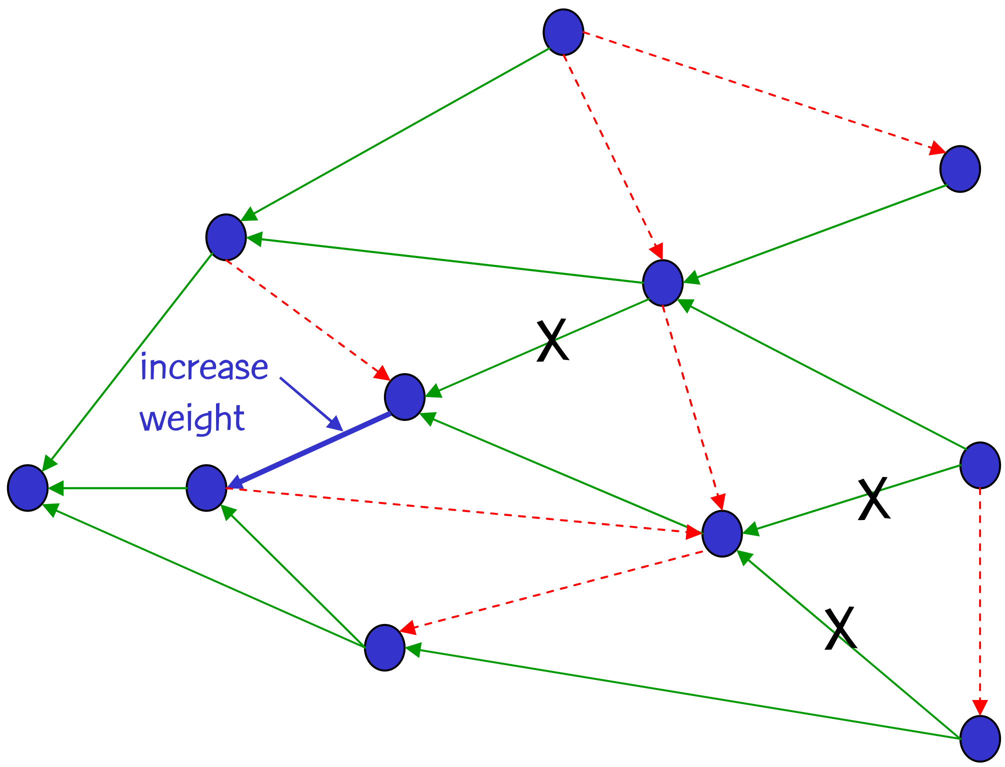


Dynamic shortest path

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

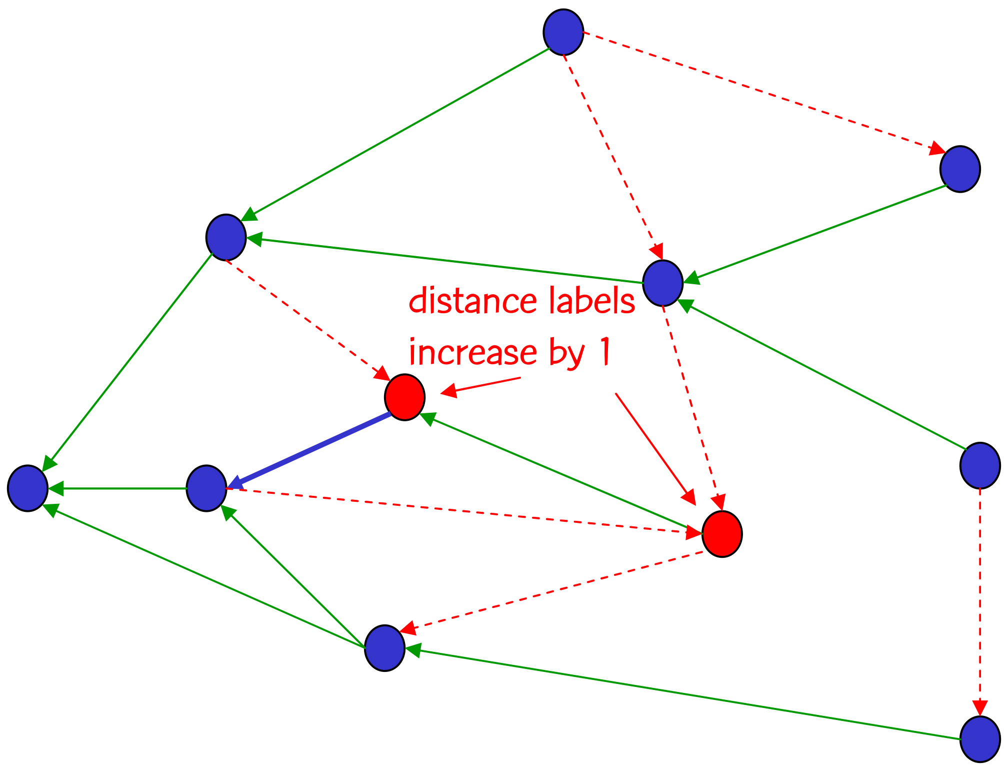


Dynamic shortest path



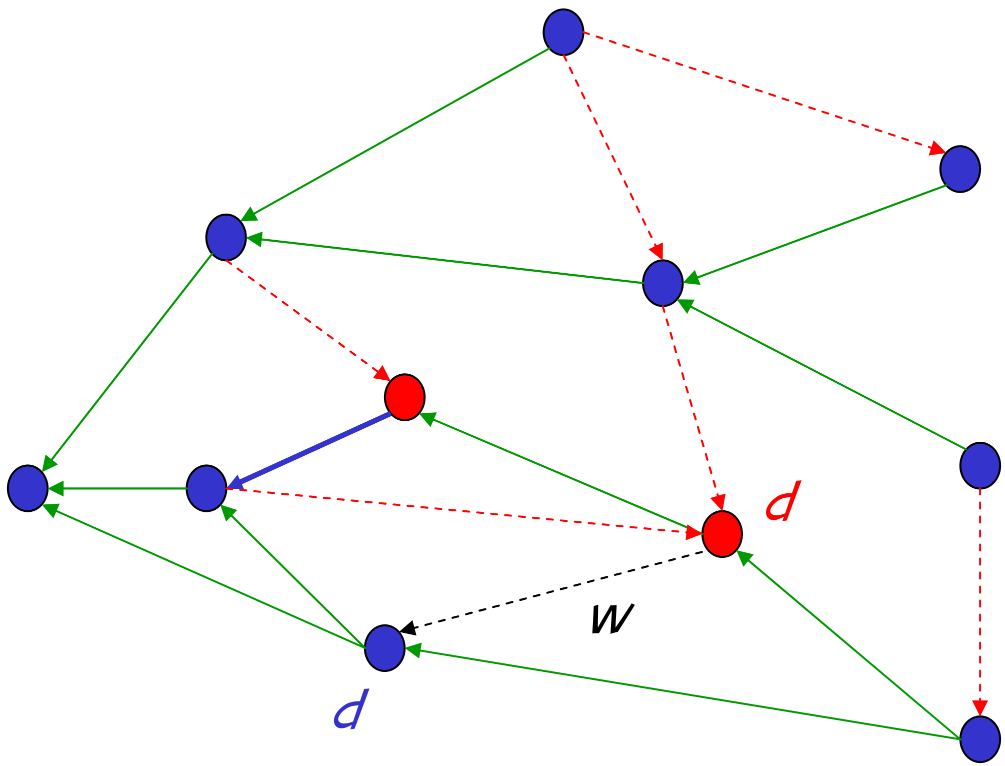
Arc (u,v) is removed from tree since alternative paths from node u to the destination node exist.

Dynamic shortest path

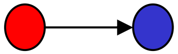


Shortest paths from red nodes must traverse blue arc.

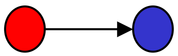
Dynamic shortest path



Test all arcs of type

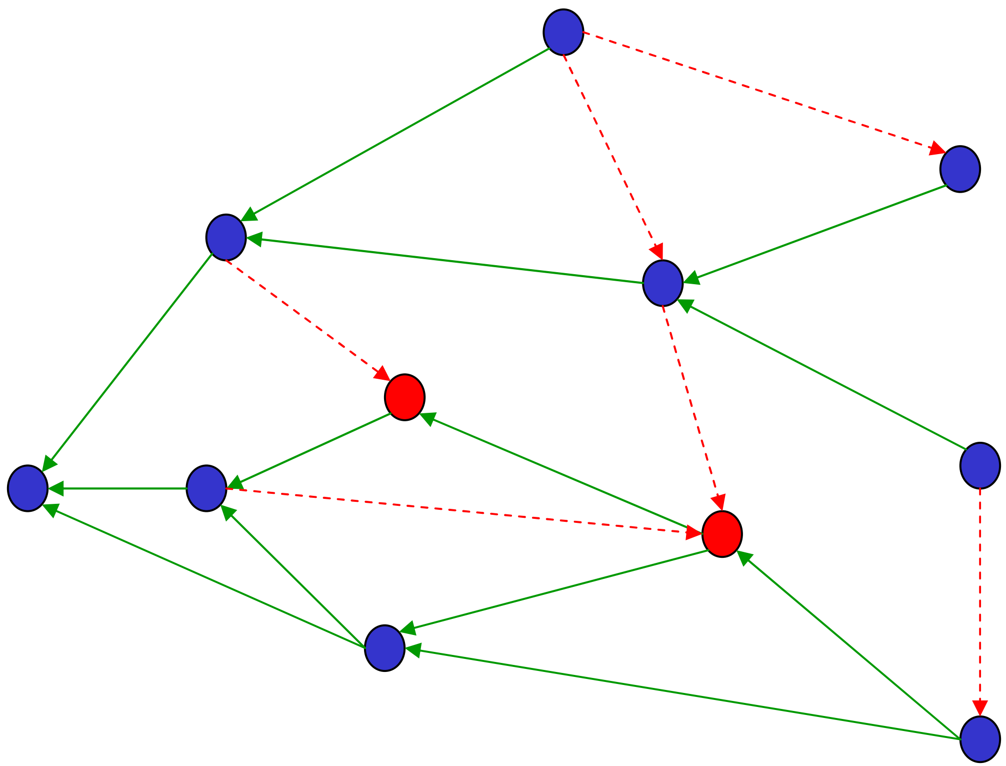


If $d - d = w$, then

 enters

tree.

Dynamic shortest path

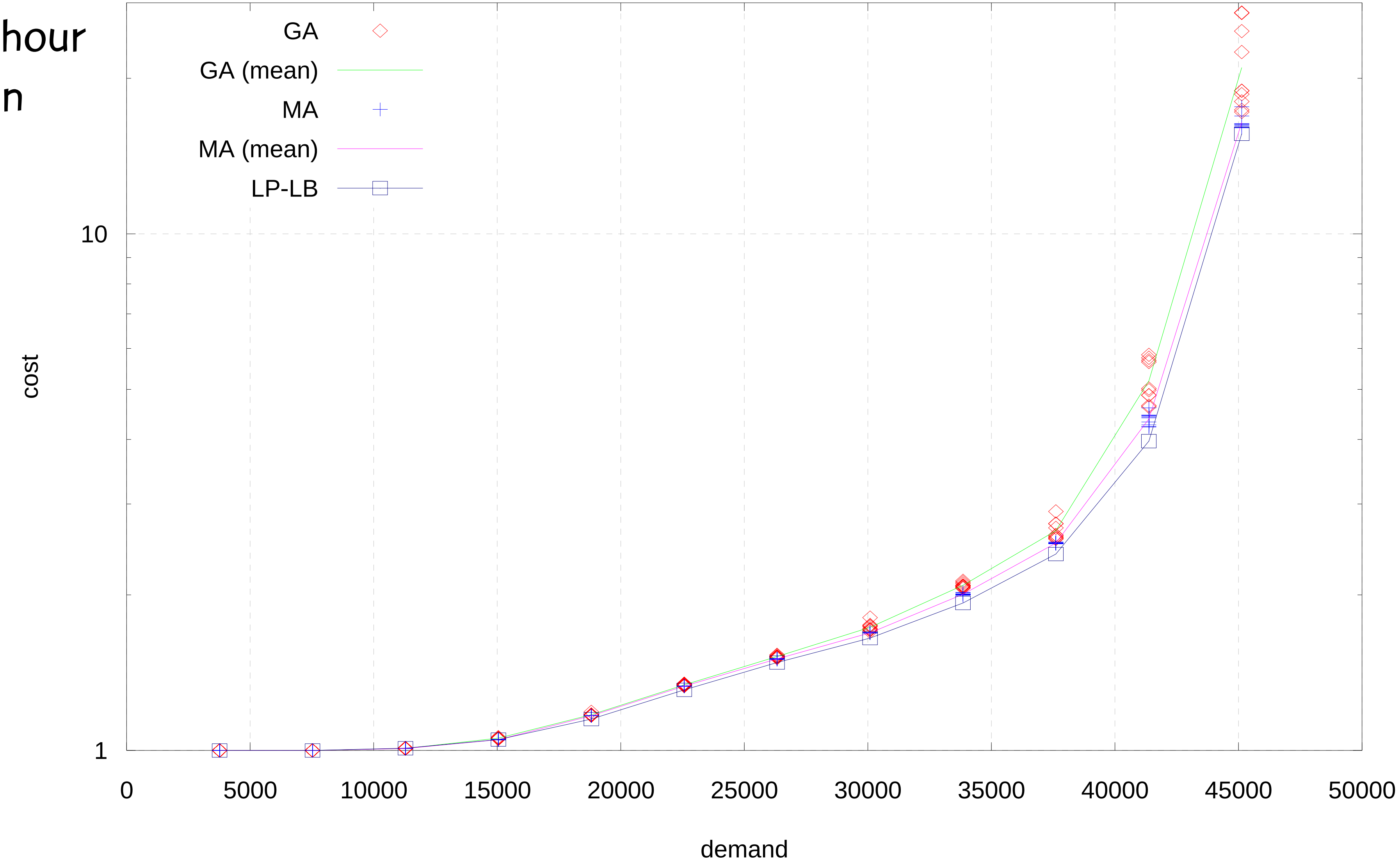


Dynamic shortest path

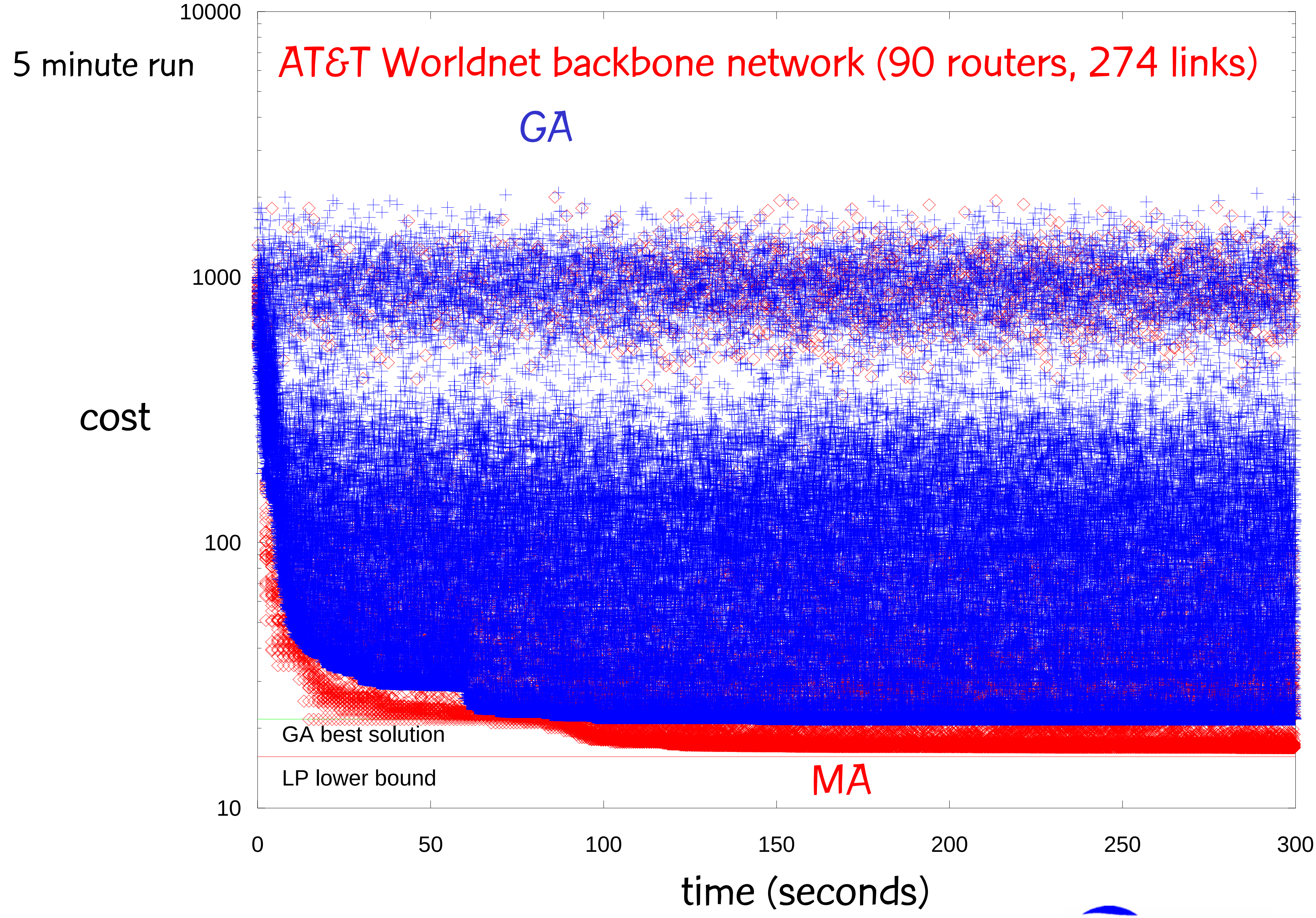
- Ramalingam & Reps (1996) allow arbitrary arc weight change.
- We specialized the Ramalingam & Reps algorithm for unit arc weight change.
  - Avoid use of heaps
  - Achieve a factor of 2~3 speedup w.r.t. Ramalingam & Reps on these test problems

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

1 hour run

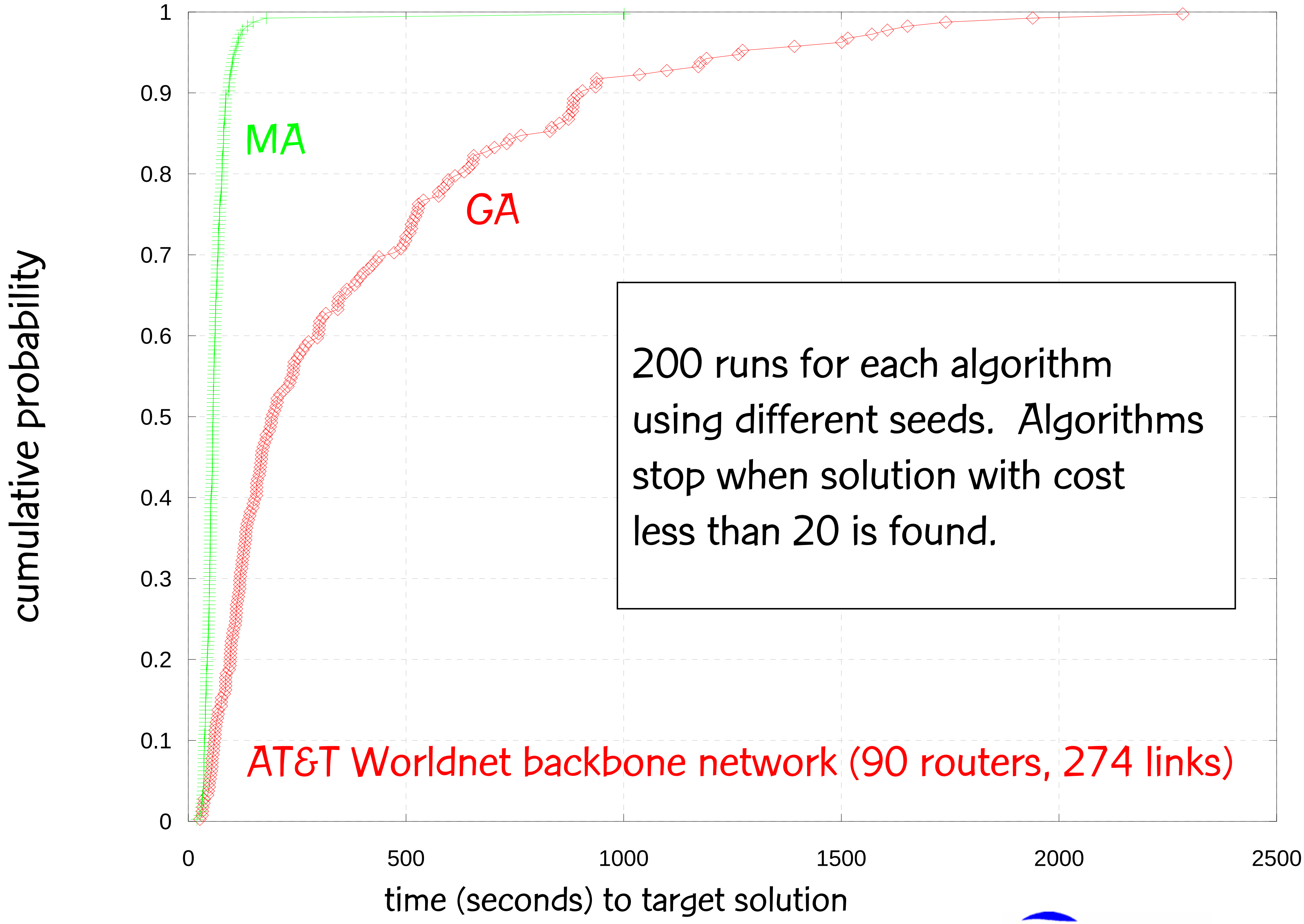


AT&T



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

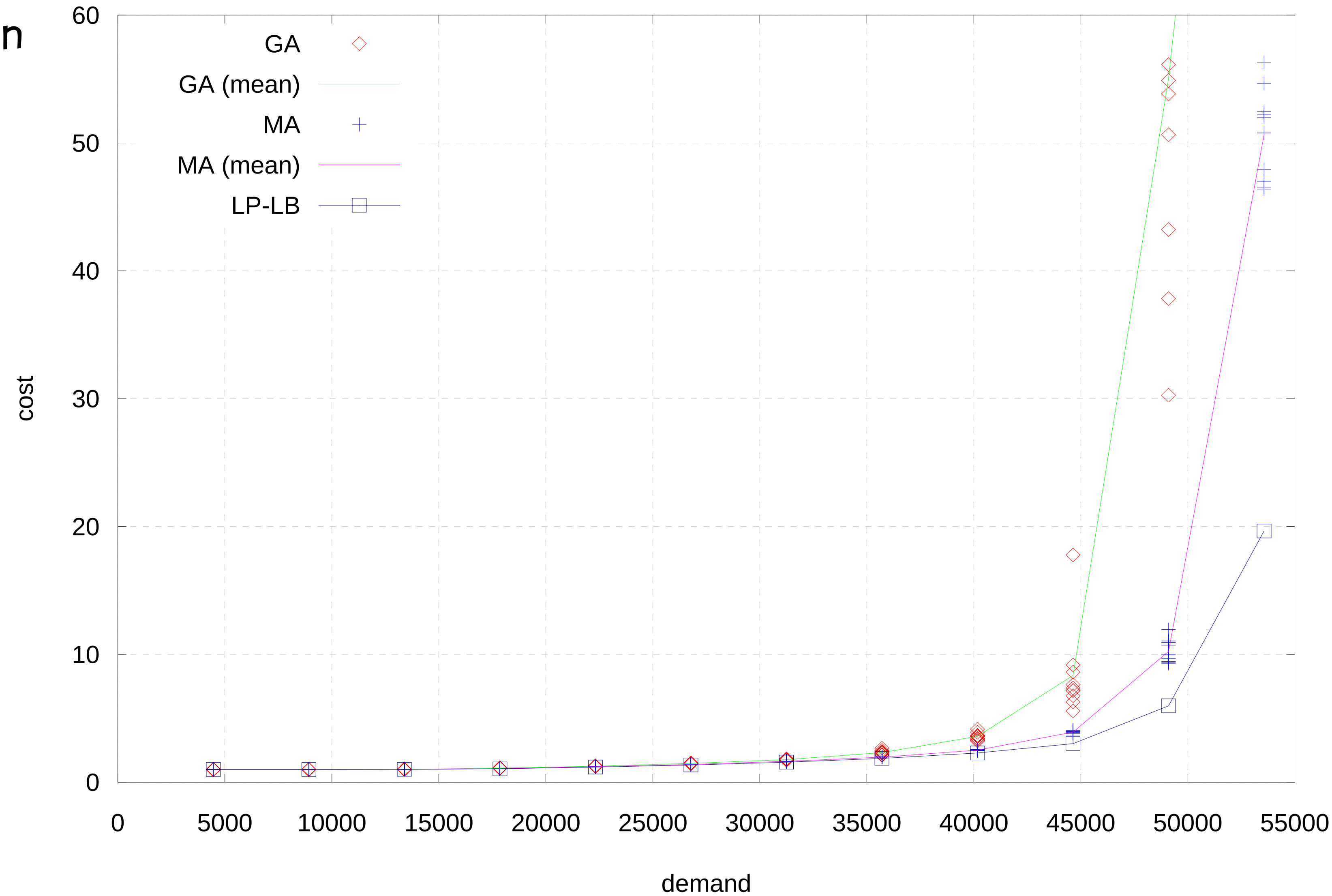
AT&T



1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
cumulative probability
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
500
1000
1500
2000
2500
time (seconds) to target solution

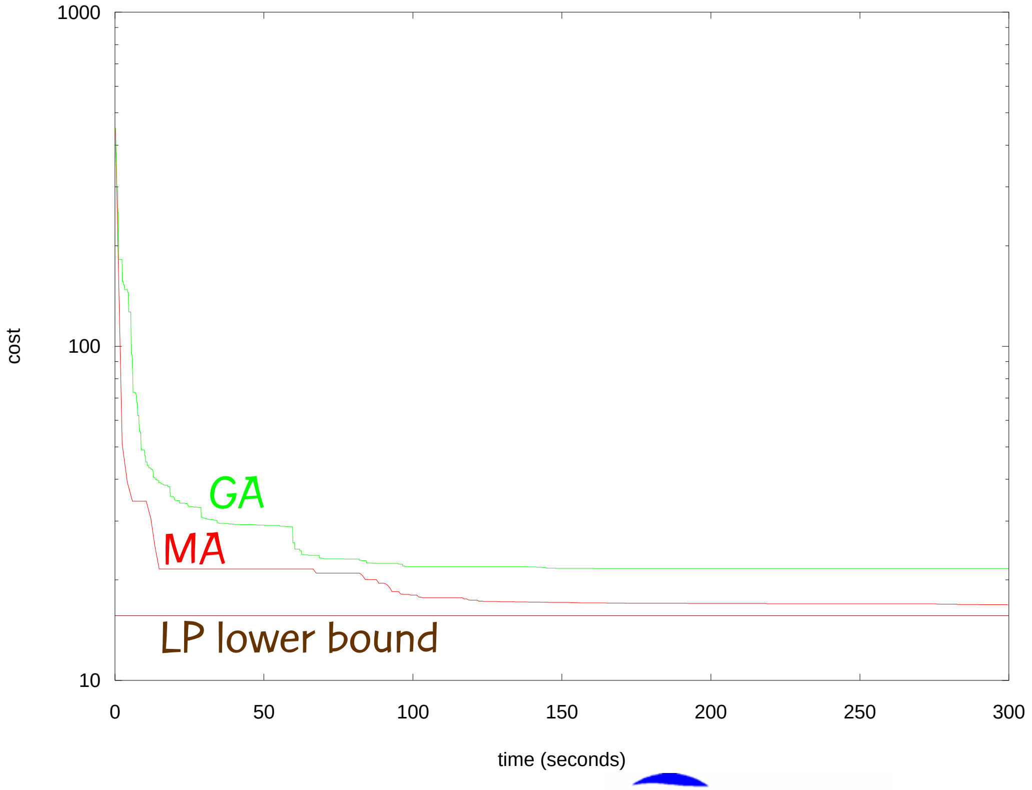
Rand50a: random graph with 50 nodes and 245 arcs.

1 hour run



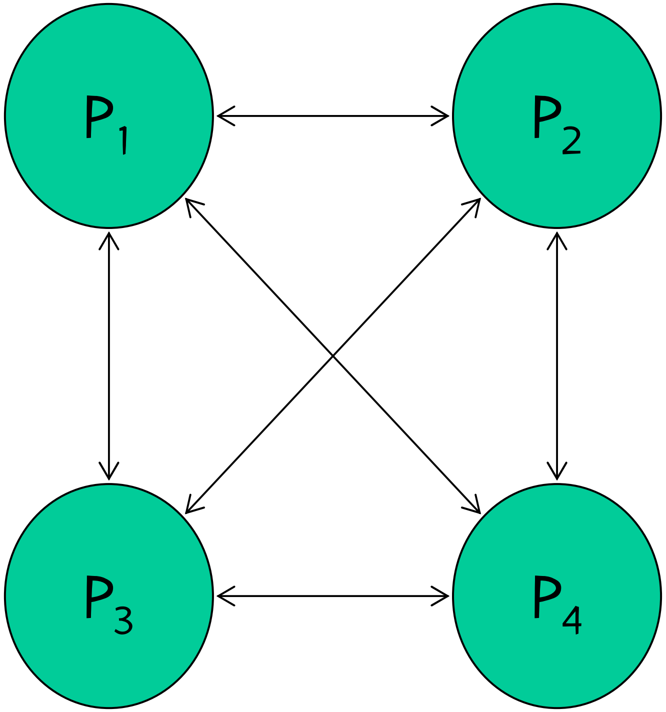
Remark

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



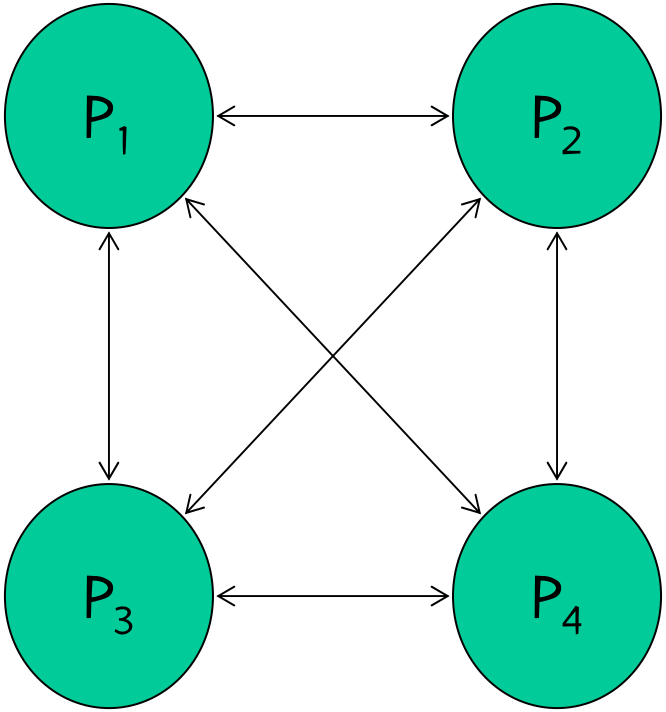
AT&T

Collaborative parallel implementation



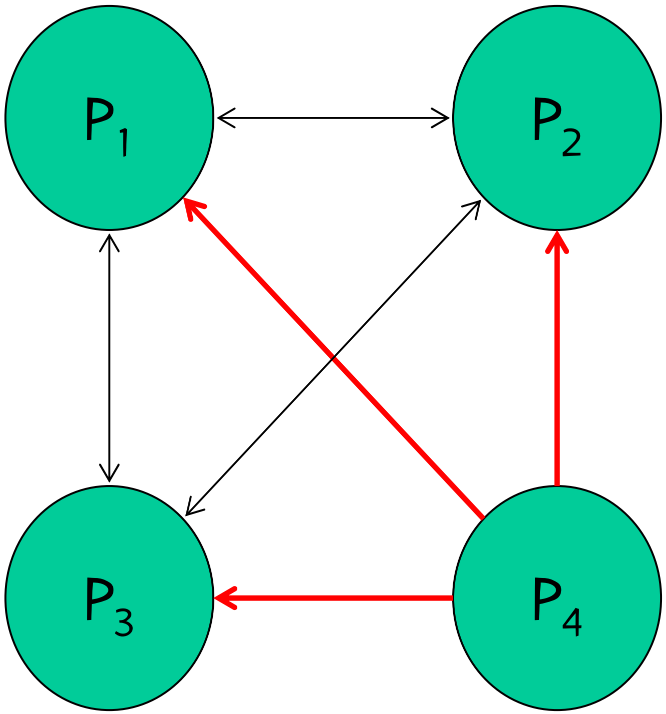
MPI: Message Passing Interface

Collaborative parallel implementation



If P_4 finds a new incumbent solution.

Collaborative parallel implementation

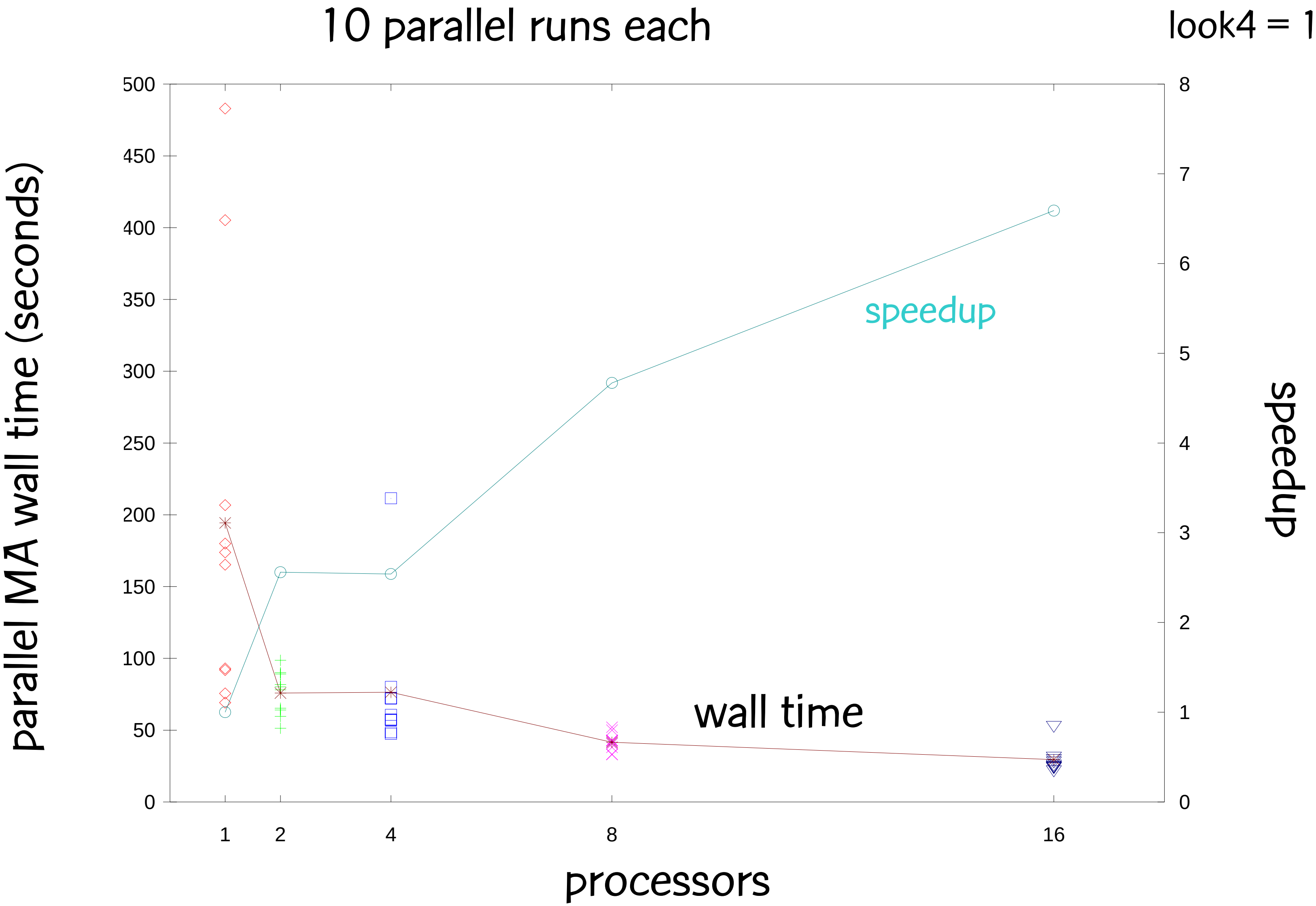


If P_4 finds a new incumbent solution.

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 capacity $\times$ distance of links to guarantee traffic flow subject to failures.
- Routing: Minimize maximum utilization subject to single link and router failures.
- Server placement: Locate minimum number of cache servers on network for multicast of streaming video.

Other applications of optimization in telecommunications

- location of traffic concentrators
  - It is sometimes beneficial to concentrate traffic into a high capacity circuit and backhaul the traffic
  - Traffic is concentrated at specific nodes
  - Problem is to decided how many nodes and which
- global routing of Frame Relay service
  - To maximize the utilization of transport infrastructure one can take advantage of varying point-to-point demands due to time zone differences

Other applications of optimization in telecommunications

- disjoint paths
  - for survivability, route several circuits between pairs of nodes on resource (node, edge) disjoint paths
  - if impossible, minimize sharing of resources
- frequency assignment
  - assign different frequencies to cellular telephone antennas to avoid interference
- communities of interest
  - in call detail graph, find communities of interest
  - find cliques or large dense subsets of phone numbers that call each other
  - found cliques of size 33 in 5 day graph with 250 million nodes and one billion edges

Concluding Remarks

- we have seen a small sample of applications of optimization in telecommunications
- opportunities for optimization arise in practice all the time
- our profession call have a major impact in telecommunications
- these slides are in my homepage
  - http://www.research.att.com/~mgcr