Some combinatorial optimization problems arising in telecommunications

Fields Industrial Optimization Seminar
Fields Institute for Research in Mathematical Sciences
Toronto, Canada ~ February 5, 2008

Mauricio G. C. Resende
AT&T Labs Research
Florham Park, New Jersey

mgcr@research.att.com

Summary

- This talk is about finding optimal or near-optimal solutions to combinatorial optimization problems that I have come across working in a large industrial telecommunications research lab.

- We apply two metaheuristics:
  - GRASP and path-relinking
  - Random-keys genetic algorithm

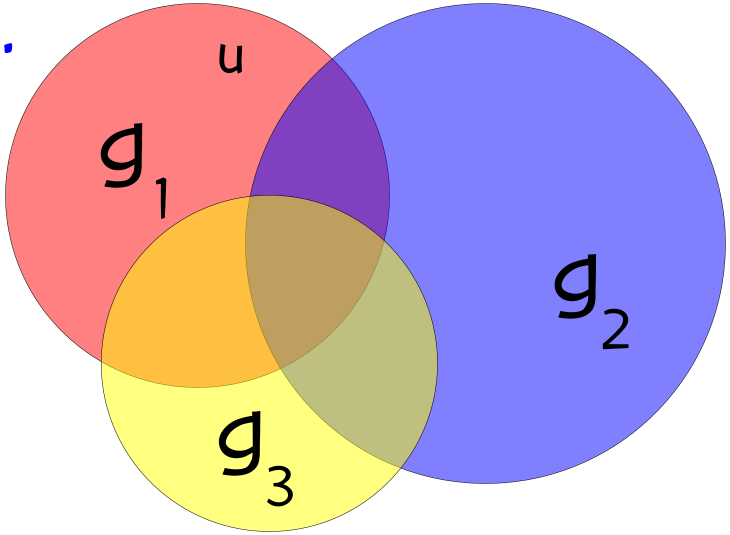
Summary

- GRASP and path-relinking to solve:
  - Scheduling: PBX telephone migration scheduling
  - Network migration: Sequencing the 4ESS deloading process
- Genetic algorithms to solve:
  - Internet routing: Optimal setting of OSPF weights
  - Network design: Survivable IP networks
  - Location: Optimal location of IPTV data centers

Batch scheduling of multi-grouped units

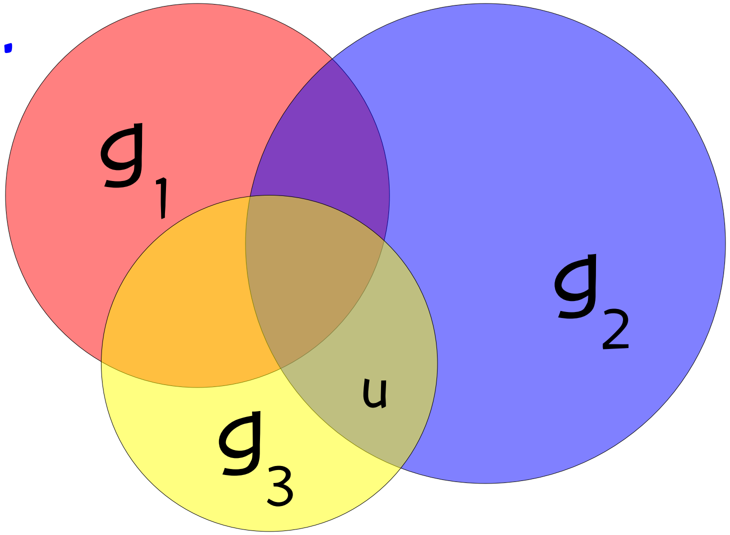
Batch scheduling of multi-grouped units

- Consider a system with
  - a set U of N units
  - a set H of M groups of units
- Each unit $u \in U$ is a member of one or more groups $g_1, g_2, \ldots, g_k \in H$.



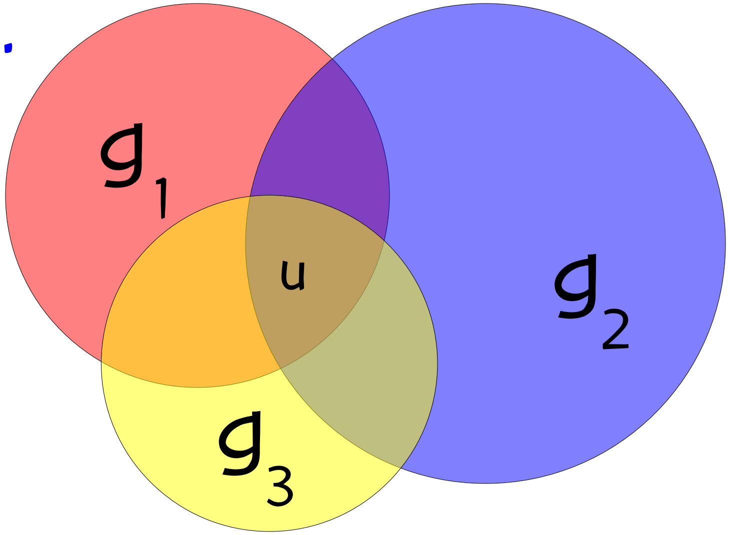
Batch scheduling of multi-grouped units

- Consider a system with
  - a set U of N units
  - a set H of M groups of units
- Each unit $u \in U$ is a member of one or more groups $g_1, g_2, \ldots, g_k \in H$.



Batch scheduling of multi-grouped units

- Consider a system with
  - a set U of N units
  - a set H of M groups of units
- Each unit $u \in U$ is a member of one or more groups $g_1, g_2, \ldots, g_k \in H$.



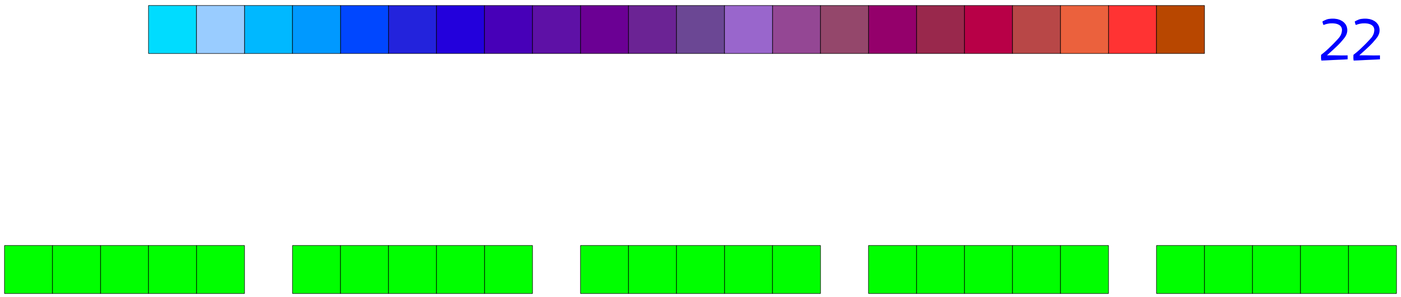
Batch scheduling of multi-grouped units

- Schedule: assign each unit to a time period.
- Given T time periods on which to schedule units.
- No more than C units can be assigned to a single time period.

22 units

Batch scheduling of multi-grouped units

- Schedule: assign each unit to a time period.
- Given T time periods on which to schedule units.
- No more than C units can be assigned to a single time period.



22 units

C=5 units/period

T=5 time periods

Batch scheduling of multi-grouped units

- Given T time periods on which to schedule units.
- No more than C units can be assigned to a single time period.

22 units

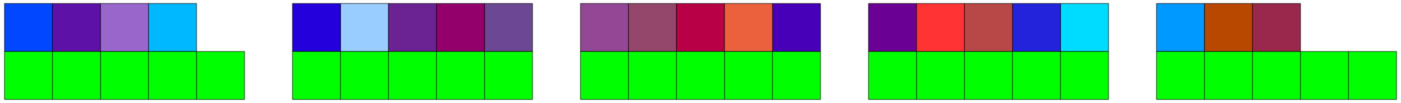
Schedule: assign each unit to a time period

C=5 units/period

T=5 time periods

Batch scheduling of multi-grouped units

- Schedule: assign each unit to a time period.
- Given T time periods on which to schedule units.
- No more than C units can be assigned to a single time period.



C=5 units/period T=5 time periods

Batch scheduling of multi-grouped units

- Objective: Schedule two units sharing same group as close together as possible.
- Let w(u,v,g) be the per-period penalty associated with assigning a group-g pair u and v to different periods.
- Scheduling penalty: Let $G(u,v) \subseteq H$ be the set of groups shared by units u and v. If units u and v are assigned to periods $\pi(u)$ and $\pi(v)$, respectively, then a penalty

$$p(u,v) = |\pi(u) - \pi(v)| \times \sum_{g \in G(u,v)} w(u,v,g)$$

is incurred.

Batch scheduling of multi-grouped units

- Objective: Schedule two units sharing same group as close together as possible.
- Let $w(u,v,g)$ be the per-period penalty associated with assigning a group-g pair u and v to different periods.
- Scheduling penalty: Let $G(u,v) \subseteq H$ be the set of groups shared by units u and v. If units u and v are assigned to periods $\pi(u)$ and $\pi(v)$, respectively, then a penalty

$$p(u,v) = |\pi(u) - \pi(v)| \times \sum_{g \in G(u,v)} w(u,v,g)$$

is incurred.

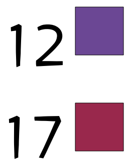
Batch scheduling of multi-grouped units

- Objective: Schedule two units sharing same group as close together as possible.
- Let $w(u,v,g)$ be the per-period penalty associated with assigning a group-g pair u and v to different periods.
- Scheduling penalty: Let $G(u,v) \subseteq H$ be the set of groups shared by units u and v. If units u and v are assigned to periods $\pi(u)$ and $\pi(v)$, respectively, then a penalty

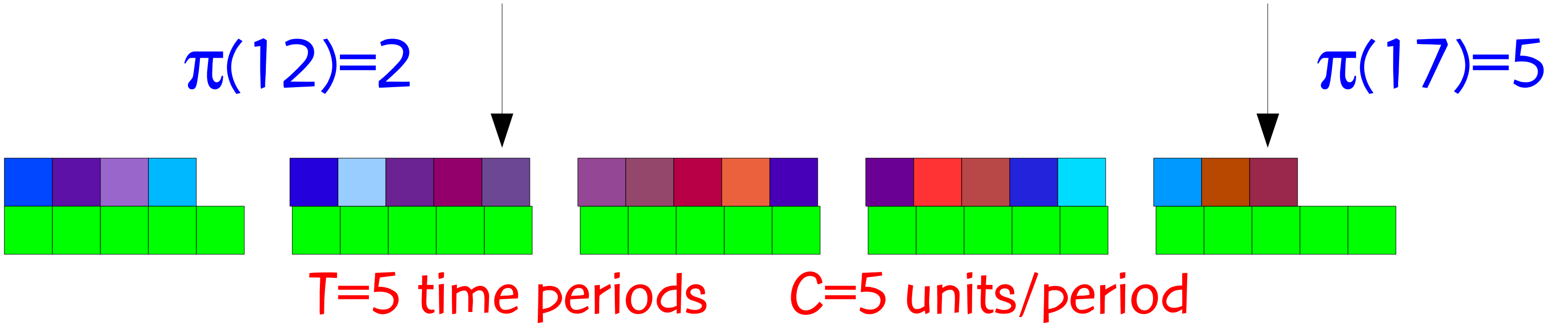
$$p(u,v) = |\pi(u) - \pi(v)| \times \sum_{g \in G(u,v)} w(u,v,g)$$

is incurred.

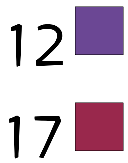
Batch scheduling of multi-grouped units



Units 12 and 17 share groups 2, 4, and 8.

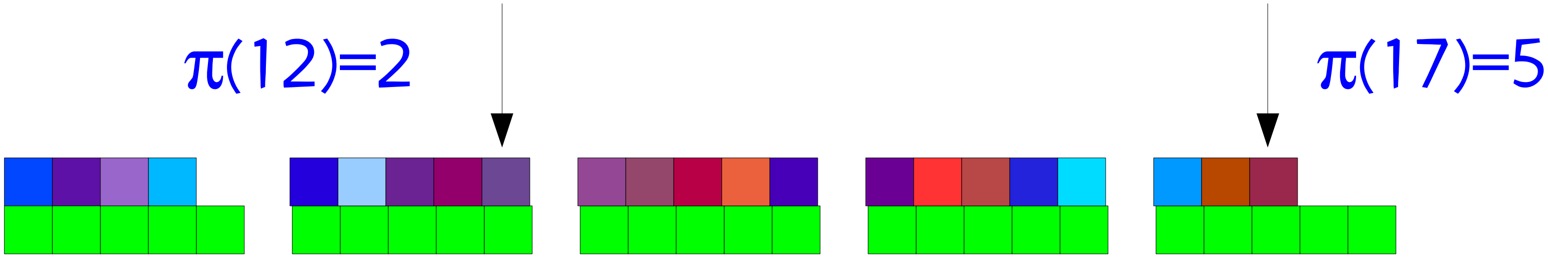


Batch scheduling of multi-grouped units



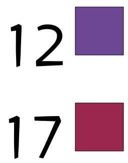
Units 12 and 17 share groups 2, 4, and 8.

Let w(12,17,2) = 10, w(12,17,4) = 20, w(12,17,8) = 5.



T=5 time periods C=5 units/period

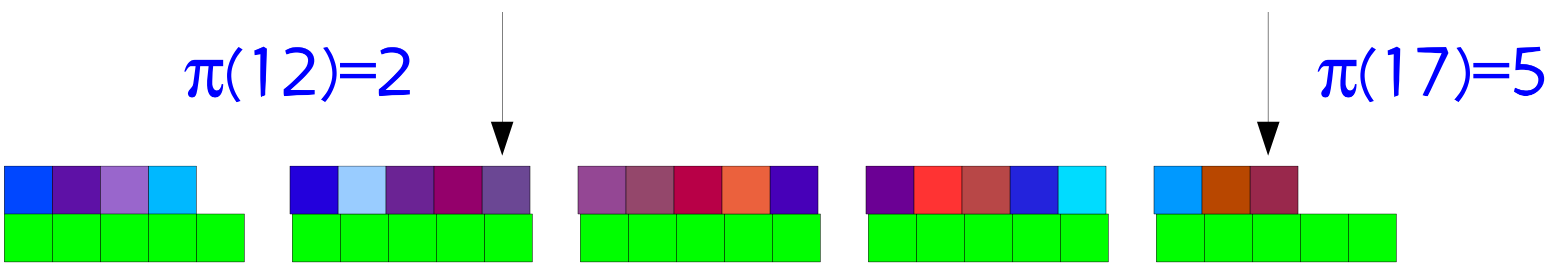
Batch scheduling of multi-grouped units



Units 12 and 17 share groups 2, 4, and 8.

Let $w(12,17,2) = 10$, $w(12,17,4) = 20$, $w(12,17,8) = 5$.

Then $w(12,17,2) + w(12,17,4) + w(12,17,8) = 35$.



T=5 time periods   C=5 units/period

Batch scheduling of multi-grouped units

12

17

Units 12 and 17 share groups 2, 4, and 8.

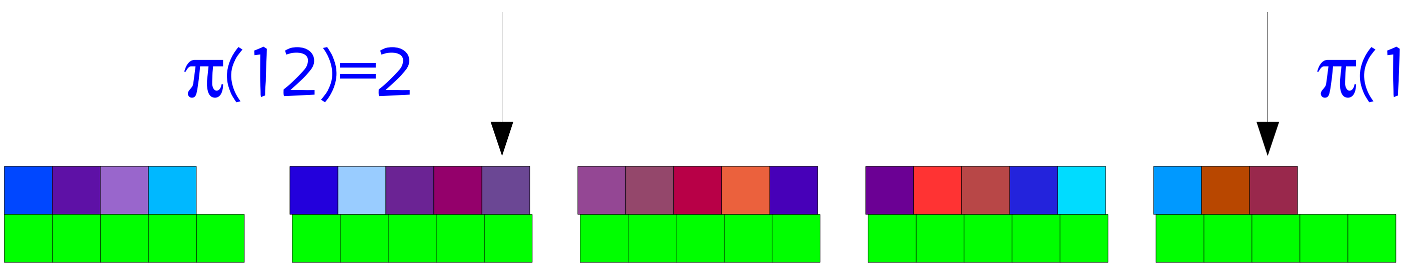
Let $w(12,17,2) = 10$, $w(12,17,4) = 20$, $w(12,17,8) = 5$.

Then $w(12,17,2) + w(12,17,4) + w(12,17,8) = 35$.

Since $\pi(12)=2$ and $\pi(17)=5$, then

$$p(12,17) = |5-2| \times 35 = 105.$$

$\pi(12)=2$ $\pi(17)=5$



T=5 time periods C=5 units/period

Batch scheduling of multi-grouped units

- Problem: Find assignment π of units to periods that

  minimizes $\sum_{\substack{u,v \in U \times U \\ (u>v)}} |\pi(u) - \pi(v)| \times \sum_{g \in G(u,v)} w(u,v,g)$

  such that

  no more than C units are assigned to any time period.

- Problem is NP-hard. It generalizes the minimum linear arrangement problem: Given a graph G(V,E), find $\pi: V \rightarrow \{1,\ldots,|V|\}$ that

  minimizes $\sum_{(u,v) \in E} |\pi(u) - \pi(v)|$

Batch scheduling of multi-grouped units

- Problem: Find assignment π of units to periods that

  minimizes $\sum_{\substack{u,v \in U \times U \\ (u>v)}} |\pi(u) - \pi(v)| \times \sum_{g \in G(u,v)} w(u,v,g)$

  such that

  no more than C units are assigned to any time period.

- Problem is NP-hard. It generalizes the minimum linear arrangement problem: Given a graph G(V,E), find $\pi: V \rightarrow \{1,\ldots,|V|\}$ that

  minimizes $\sum_{(u,v) \in E} |\pi(u) - \pi(v)|$

Batch scheduling of multi-grouped units

- Problem: Find assignment π of units to periods that

  minimizes $\sum_{\substack{u,v \in U \times U \\ (u>v)}} |\pi(u) - \pi(v)| \times \sum_{g \in G(u,v)} w(u,v,g)$

  such that

  no more than C units are assigned to any time period.

- Problem is NP-hard. It generalizes the minimum linear arrangement problem: Given a graph G(V,E), find $\pi: V \rightarrow \{1,\ldots,|V|\}$ that

  minimizes $\sum_{(u,v) \in E} |\pi(u) - \pi(v)|$

Our solution

- Metaheuristics are high level procedures that coordinate simple heuristics, such as local search, to find solutions that are of better quality than those found by the simple heuristics alone.
- One such metaheuristic is GRASP (Feo & R., 1995)
  - repeat ...
    - construct solution using randomized greedy algorithm
    - do local search starting from constructed solution
  - return best local minimum found

Our solution

- Hybridization of metaheuristics
- GRASP with evolutionary path-relinking (PR)
  - repeat ... maintaining pool of elite solutions
    - construct solution using randomized greedy algorithm
    - do local search starting from constructed solution
    - PR: explore path connecting local min and some elite solution
    - once in while do evolutionary PR, i.e. improve pool by exploring paths connecting elite solutions
  - return best elite solution found

GRASP and path-relinking

- M.G.C.R. and C. C. Ribeiro, "Greedy randomized adaptive search procedures," in "Handbook of Metaheuristics," F. Glover and G. Kochenberger, eds., Kluwer Academic Publishers, pp. 219-249, 2003.
- M.G.C.R. and C.C. Ribeiro, "GRASP with path-relinking: Recent advances and applications," in "Metaheuristics: Progress as Real Problem Solvers," T. Ibaraki, K. Nonobe and M. Yagiura, (Eds.), Springer, pp. 29-63, 2005

PBX telephone migration scheduling

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.
- 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

- 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.
- 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

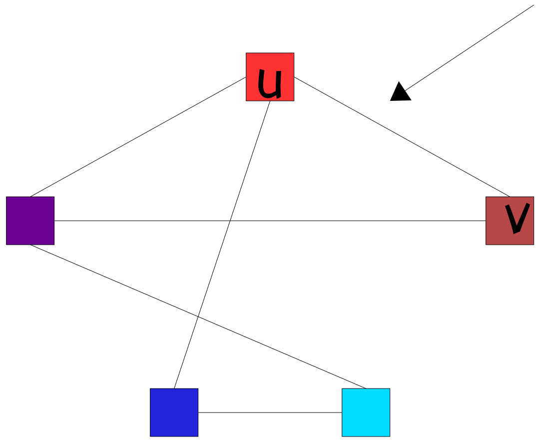
- Phone migration occurs when an organization upgrades to a newer phone switch (PBX).
- All phones using the old PBX must be moved to the new PBX.
- Each phone belongs to one or more groups of phones that interact and should to be moved together in same time period.
- Given penalties for not moving a 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

- 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.
- 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

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

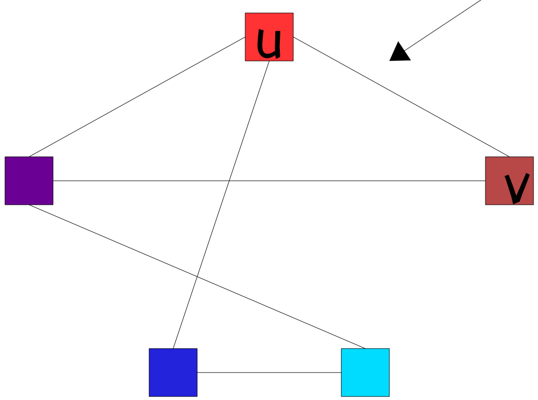


Each phone is a unit.

Edge weight between phones is penalty associated with phone pair.

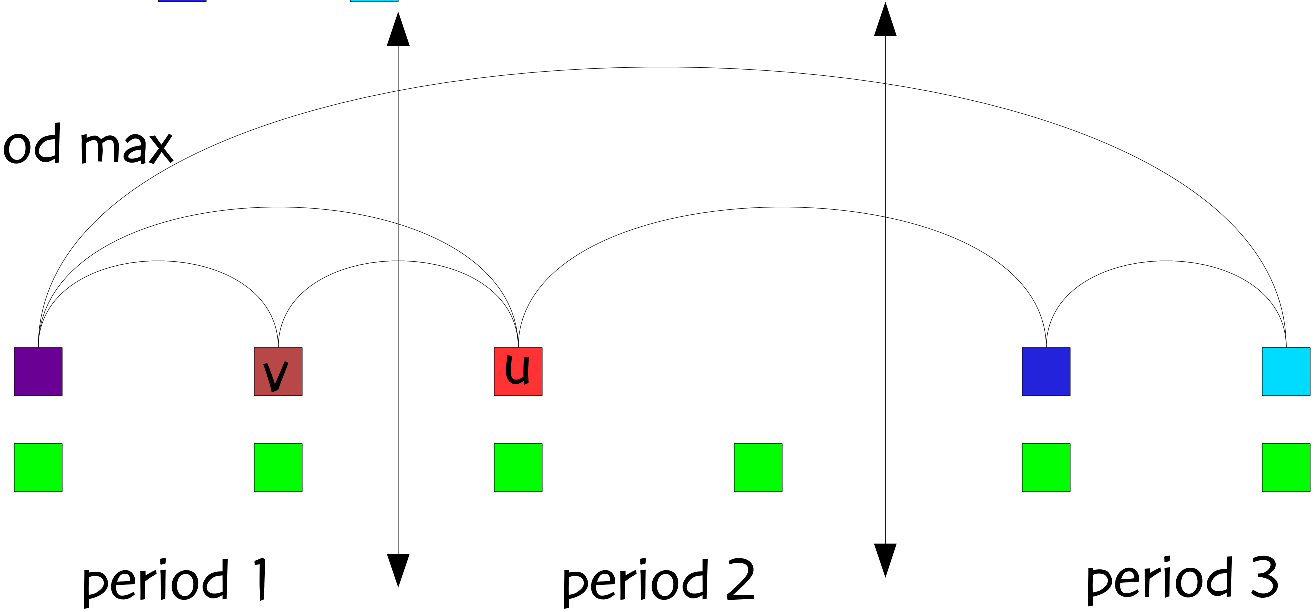
PBX telephone migration scheduling

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



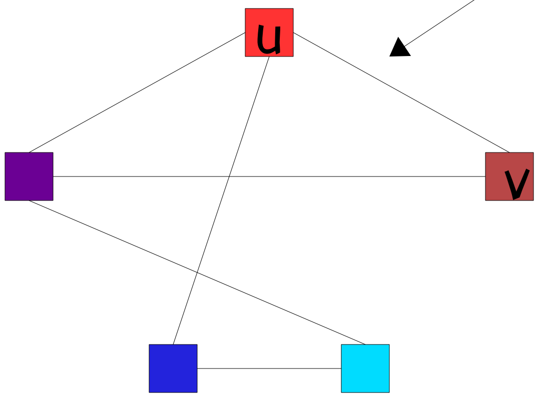
Each phone is a unit.
Edge weight between phones is penalty associated with phone pair.

3 periods
2 units/period max



PBX telephone migration scheduling

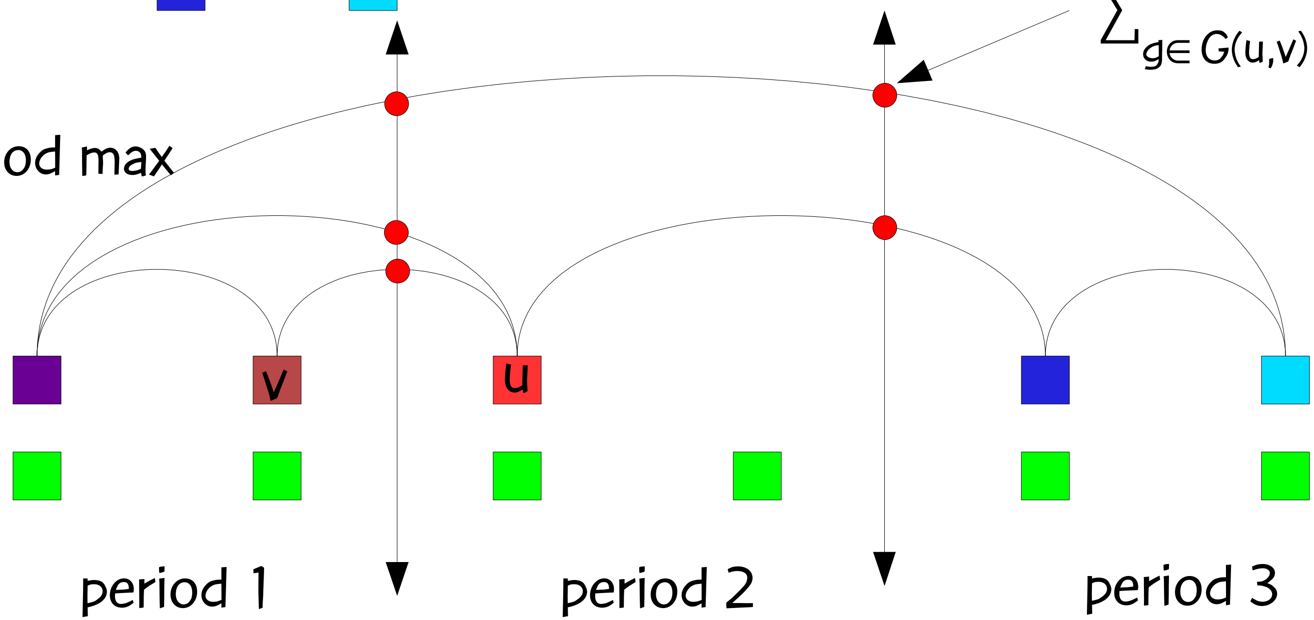
$\sum_{g \in G(u,v)} w(u,v,g)$



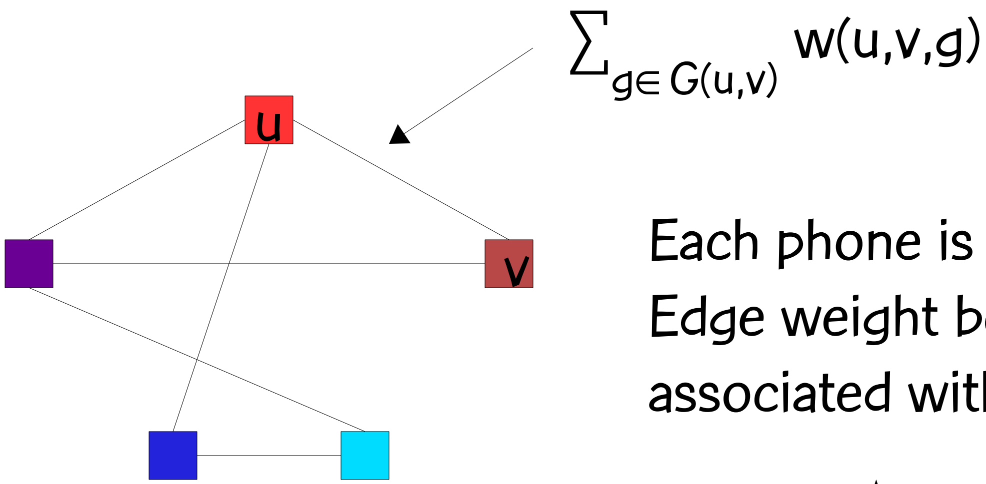
Each phone is a unit.
Edge weight between phones is penalty associated with phone pair.

3 periods
2 units/period max

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

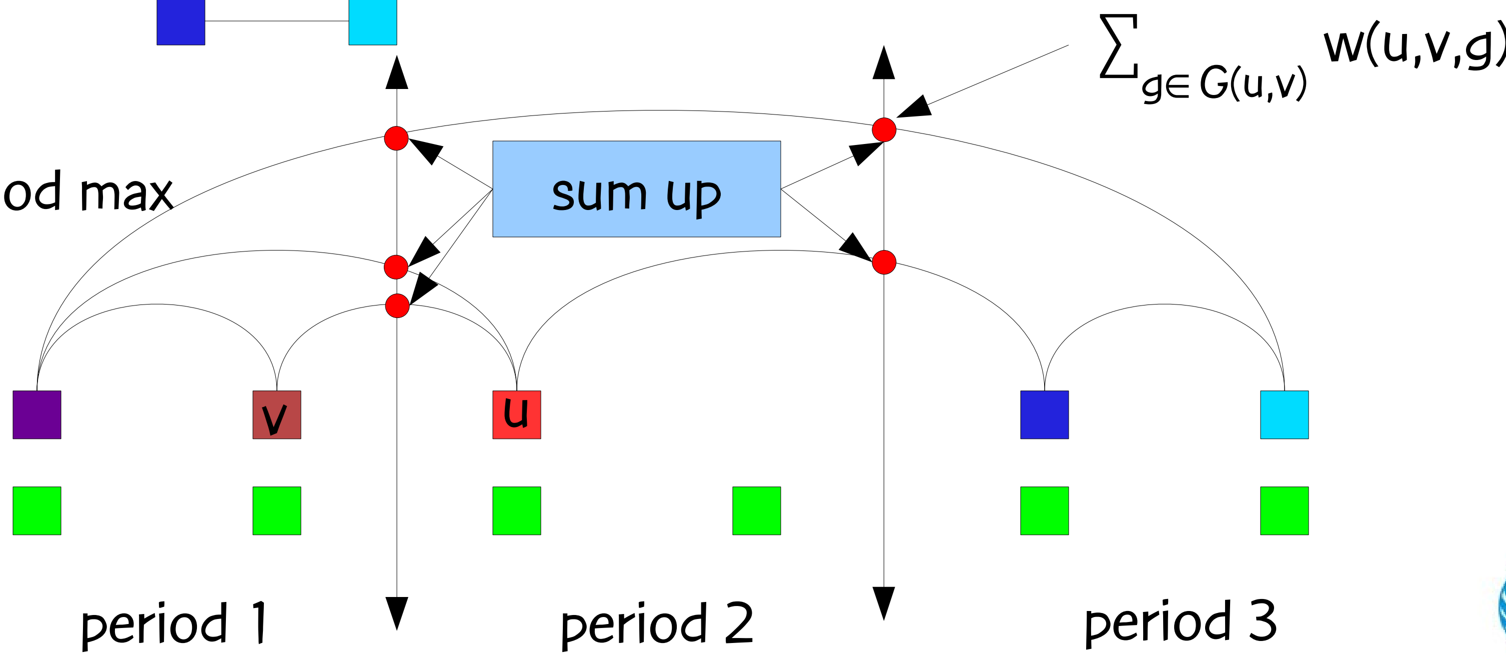


PBX telephone migration scheduling

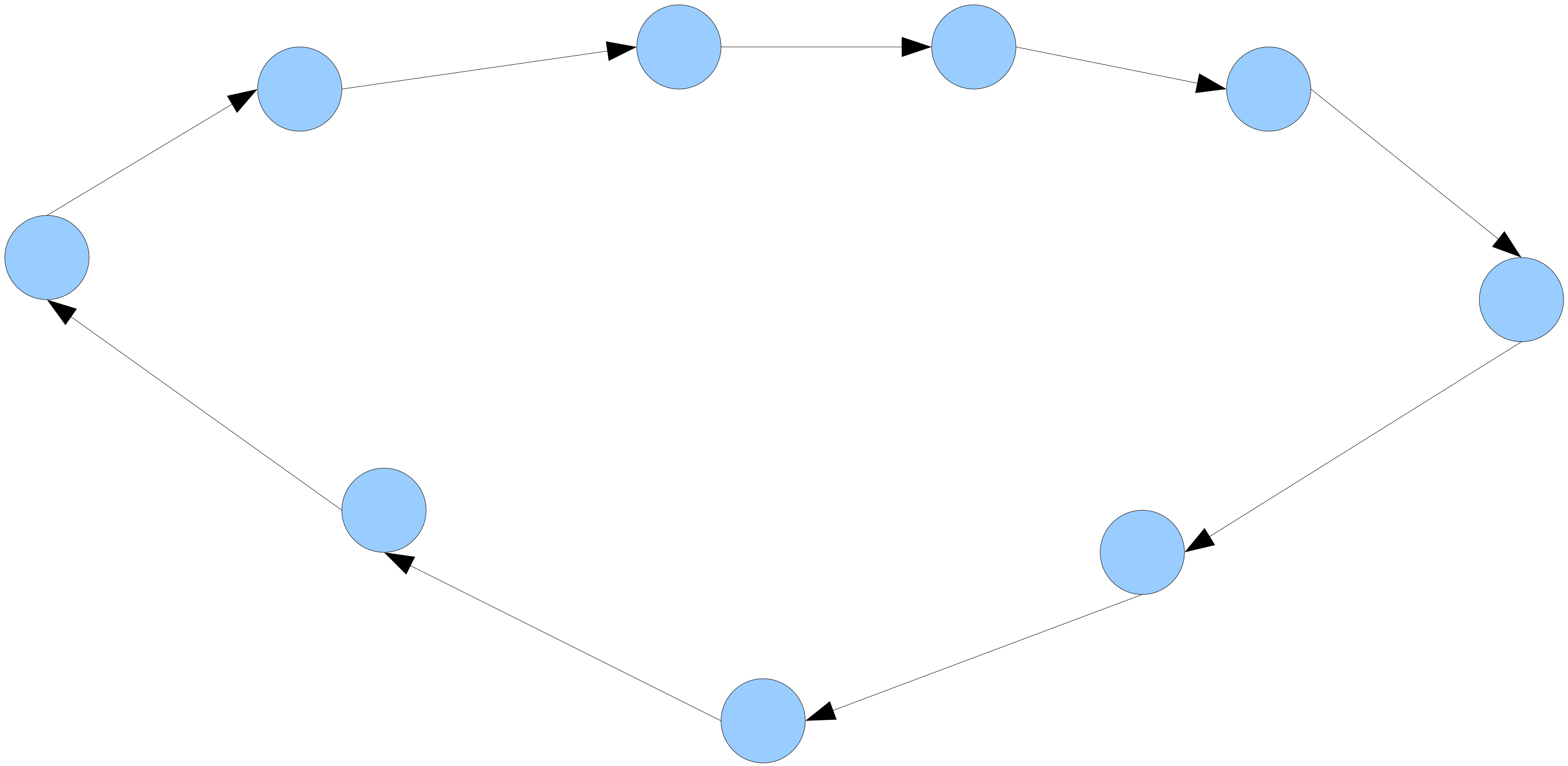


Each phone is a unit.
Edge weight between phones is penalty associated with phone pair.

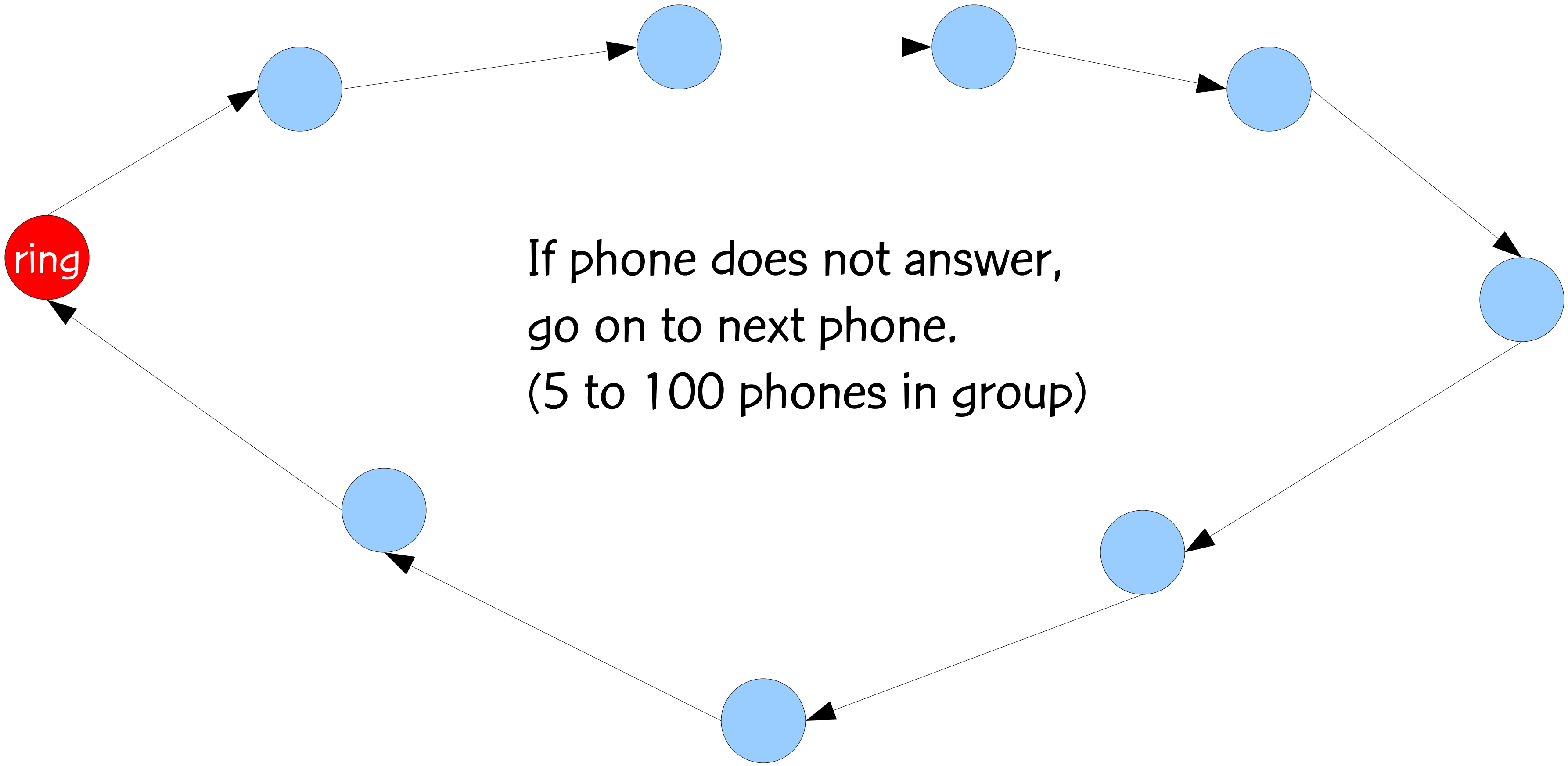
3 periods
2 units/period max



Multi-line hunt group

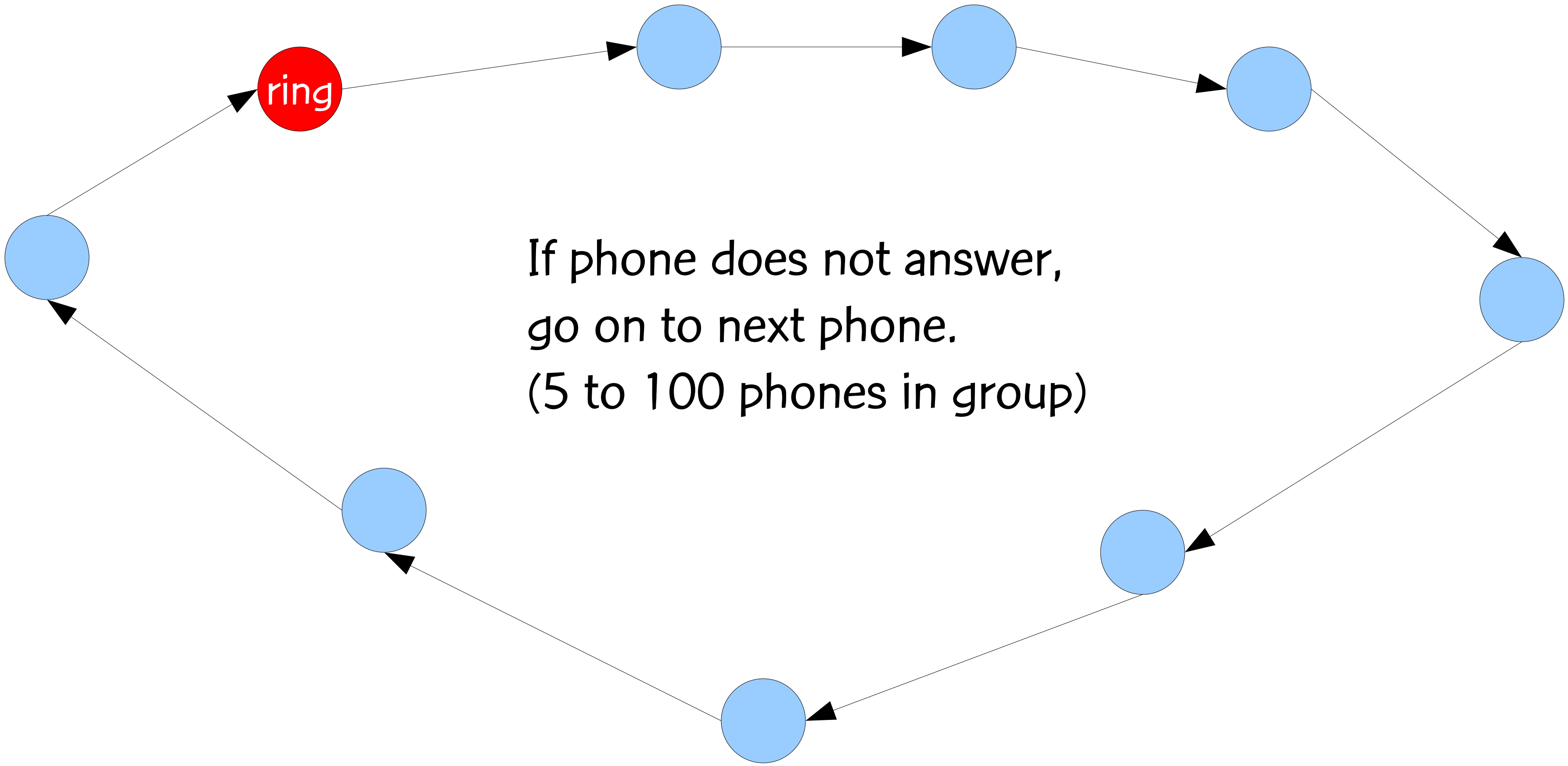


Multi-line hunt group



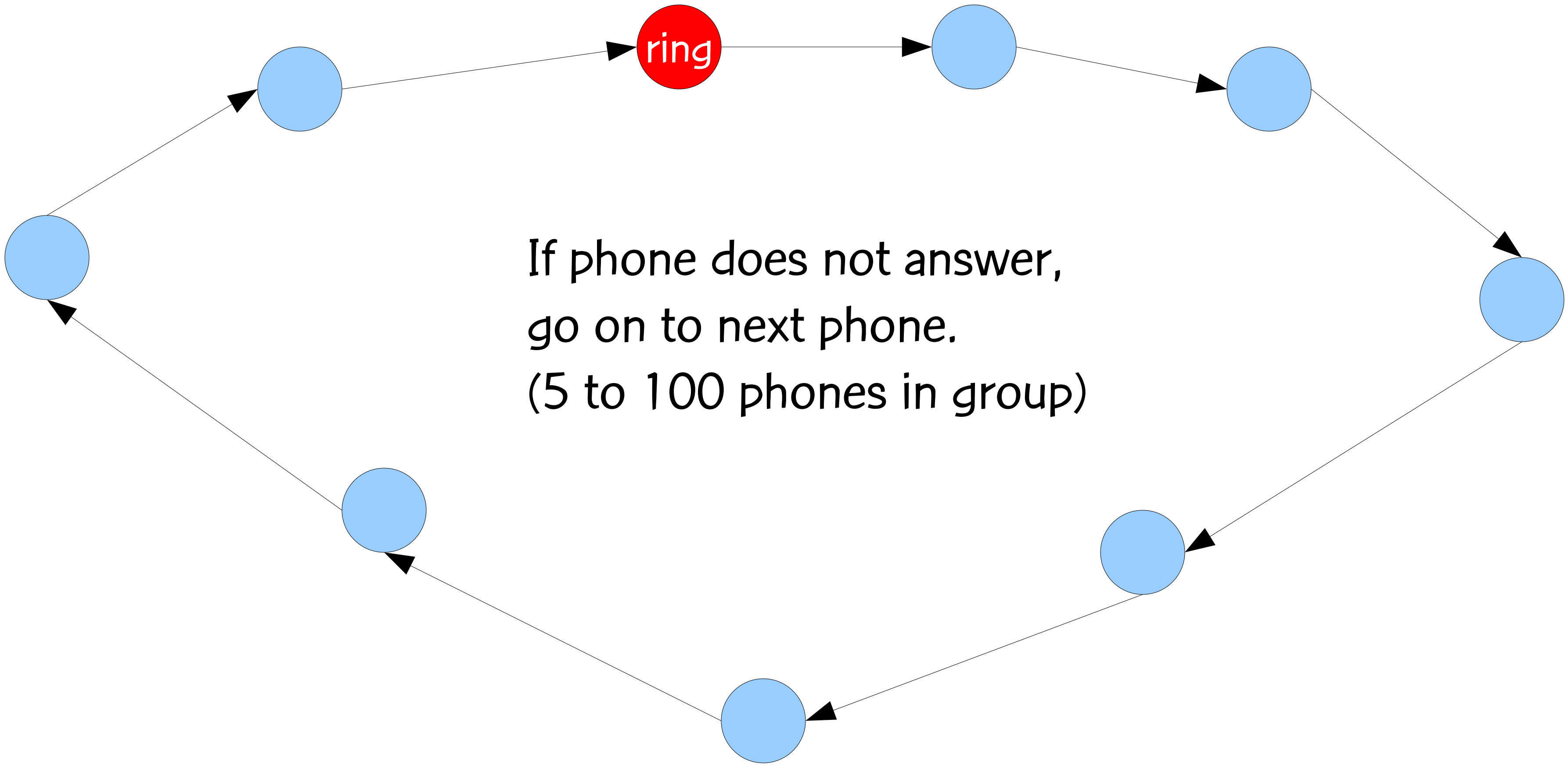
If phone does not answer,
go on to next phone.
(5 to 100 phones in group)

Multi-line hunt group



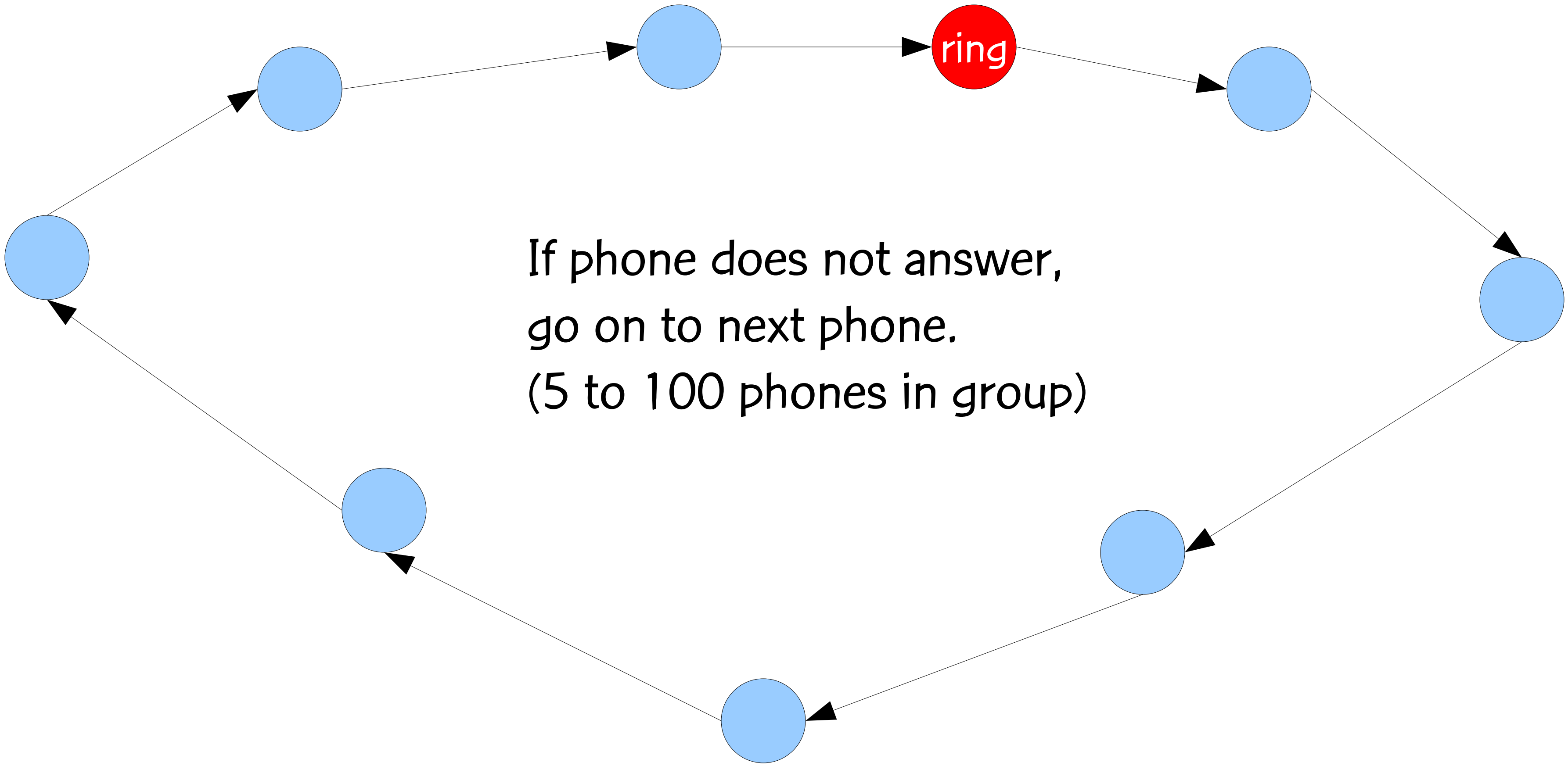
If phone does not answer,
go on to next phone.
(5 to 100 phones in group)

Multi-line hunt group



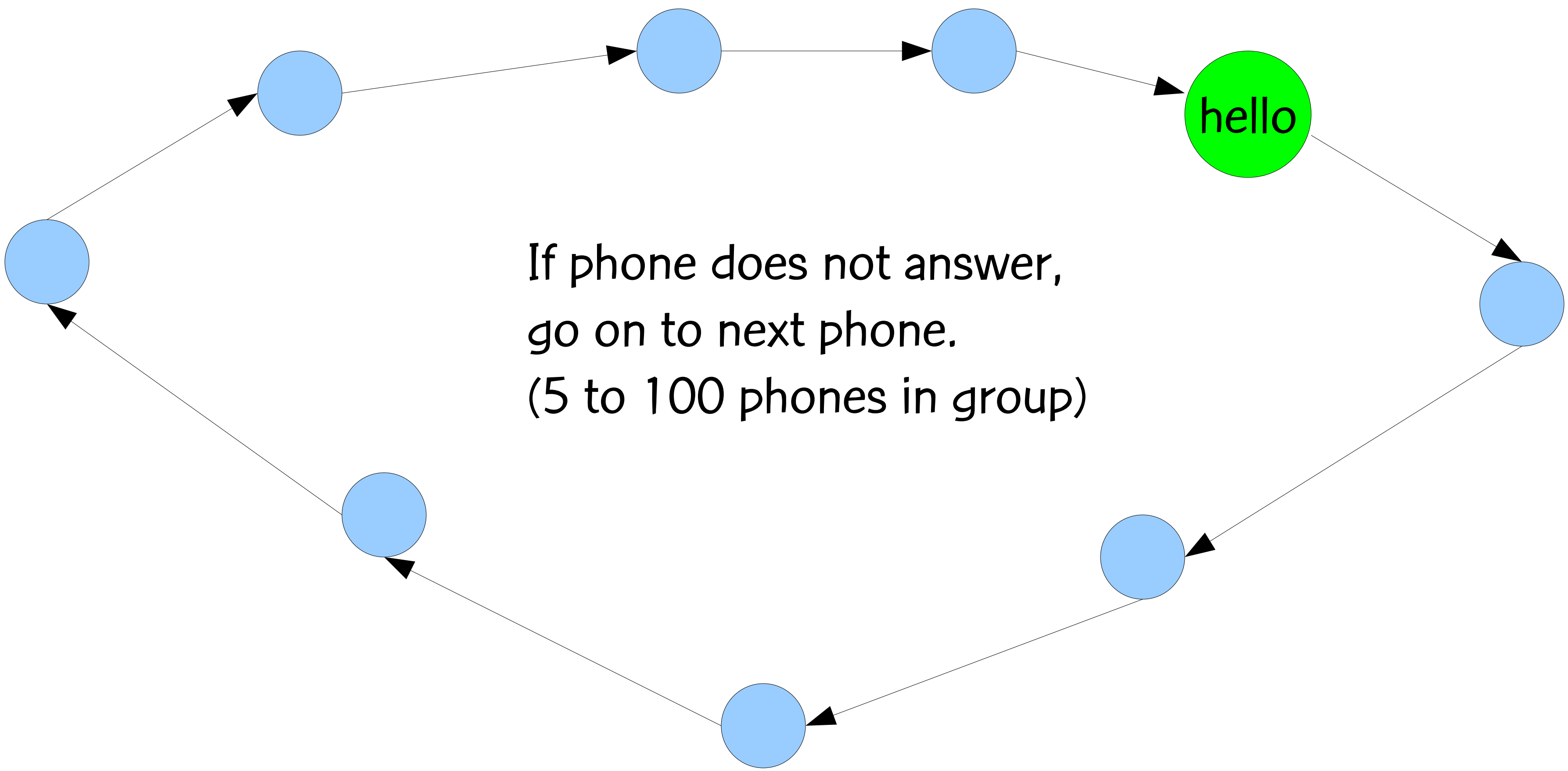
If phone does not answer,
go on to next phone.
(5 to 100 phones in group)

Multi-line hunt group



If phone does not answer,
go on to next phone.
(5 to 100 phones in group)

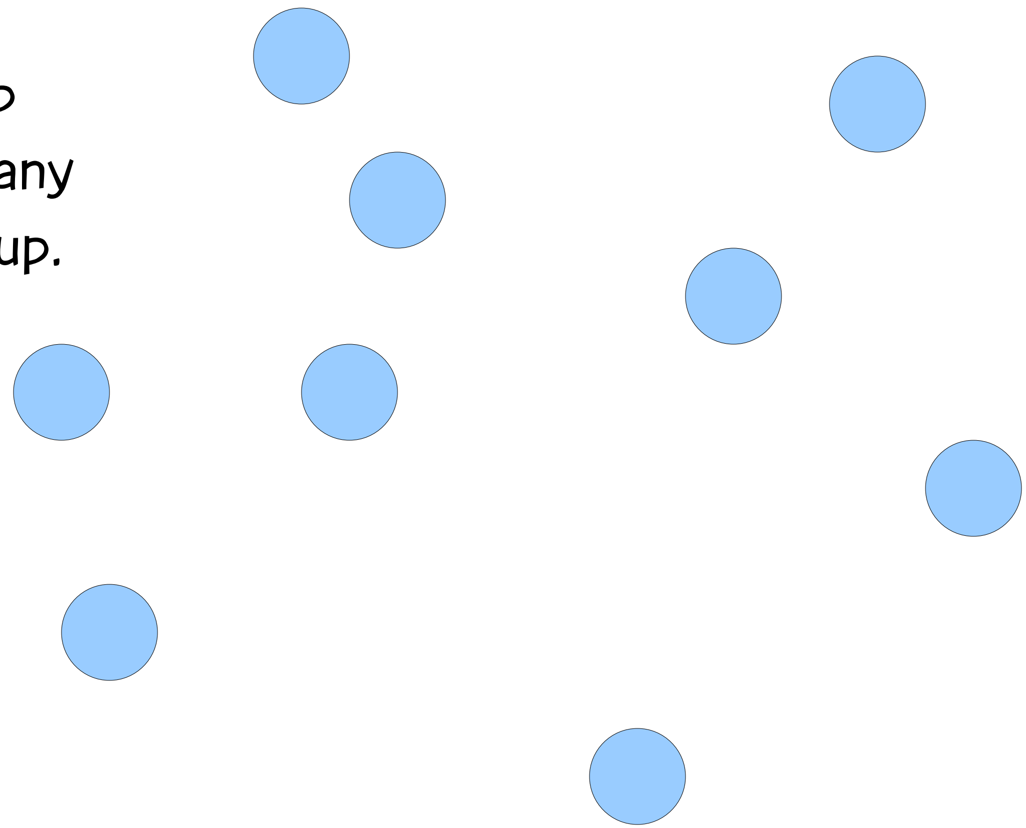
Multi-line hunt group



If phone does not answer,
go on to next phone.
(5 to 100 phones in group)

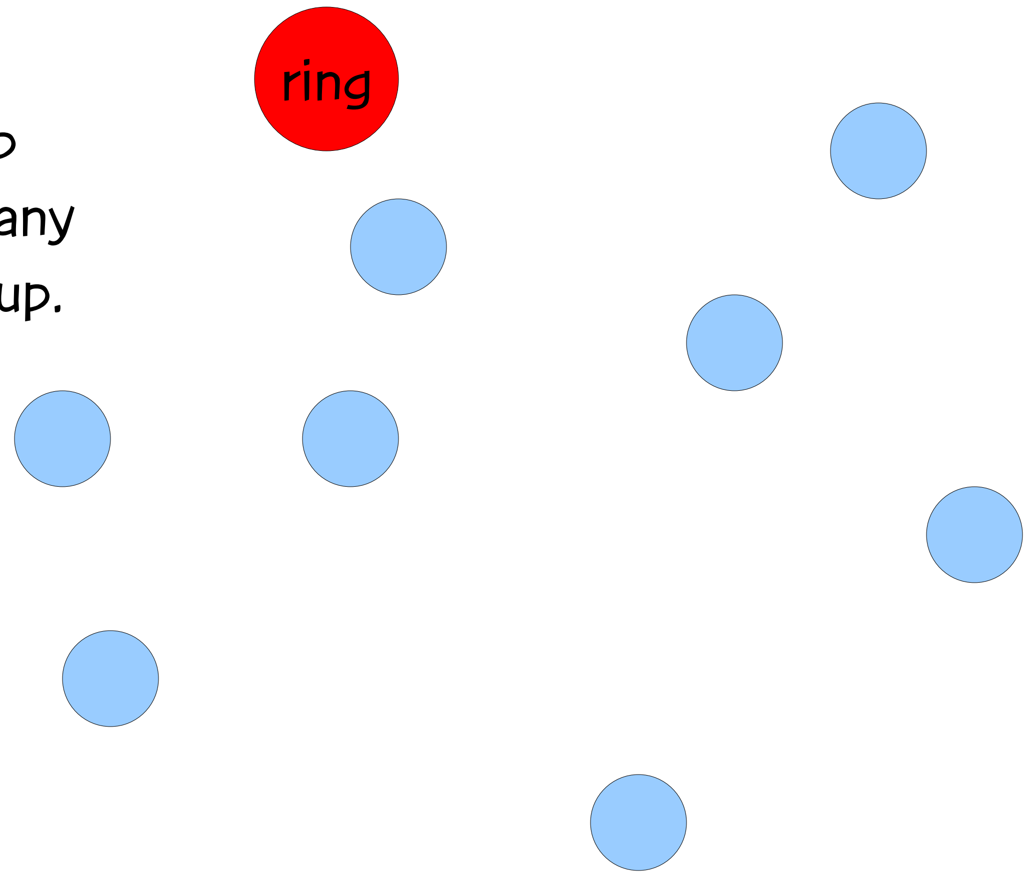
Call pickup (CPU)

Any phone in group can pickup call for any other phone in group.



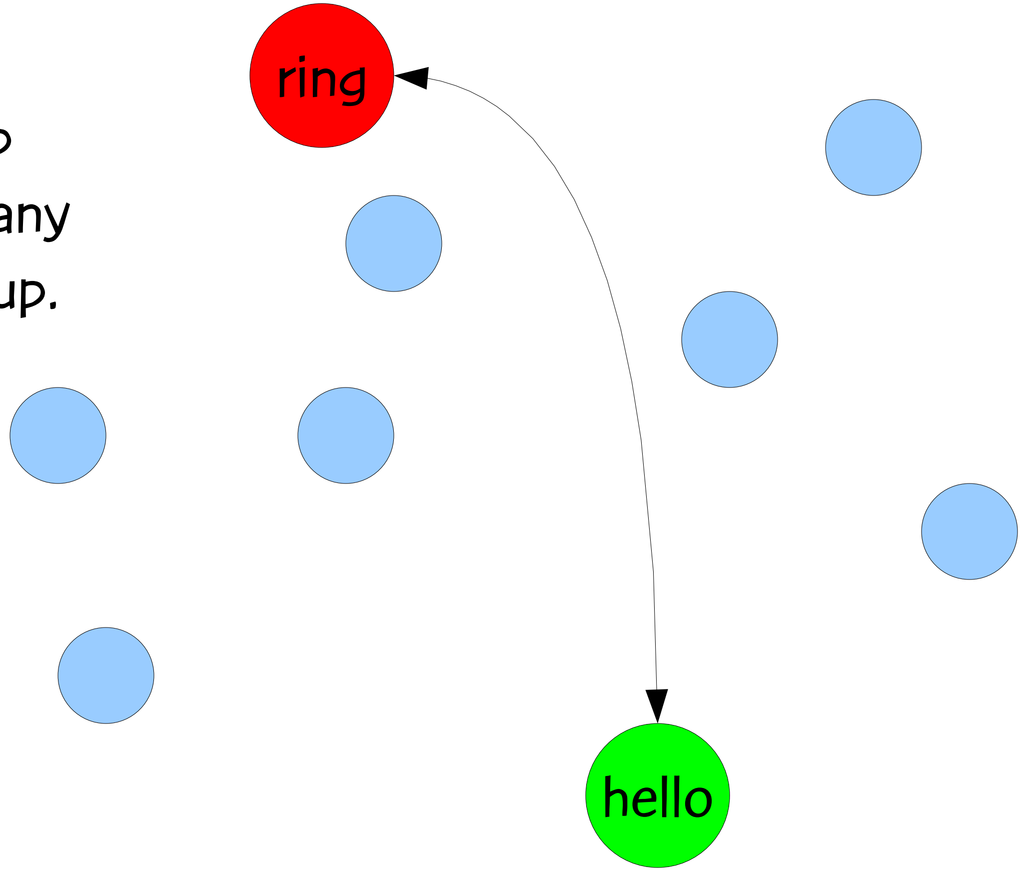
Call pickup (CPU)

Any phone in group can pickup call for any other phone in group.



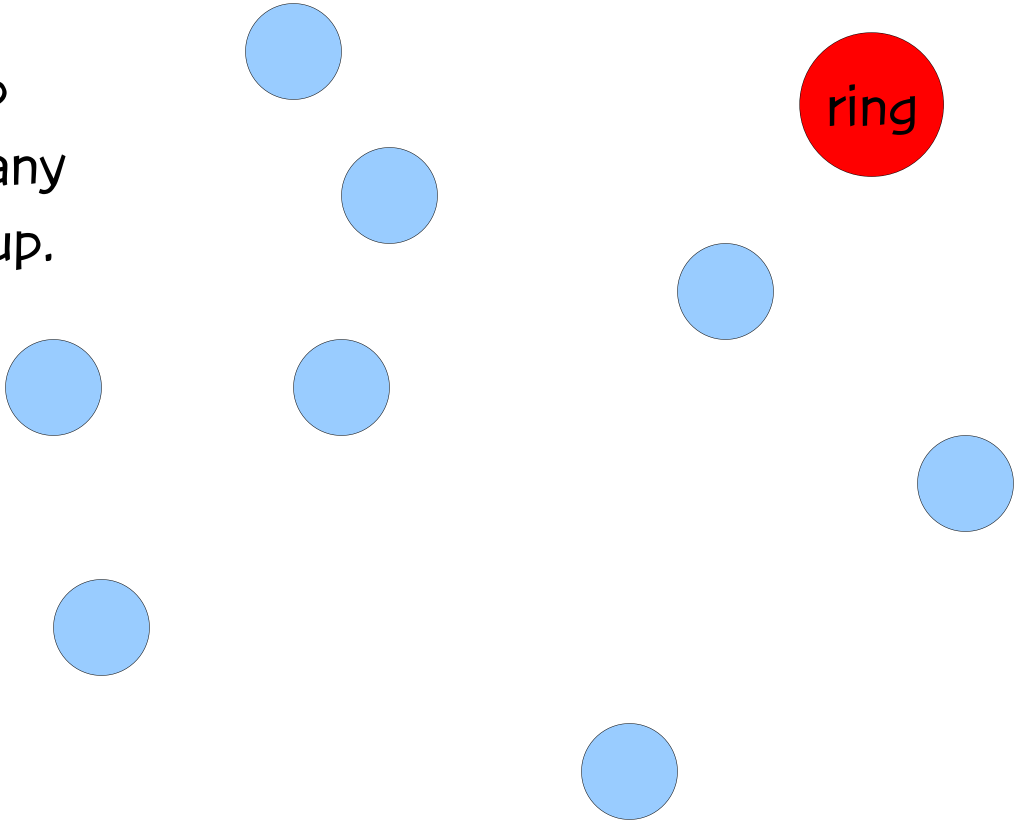
Call pickup (CPU)

Any phone in group can pickup call for any other phone in group.



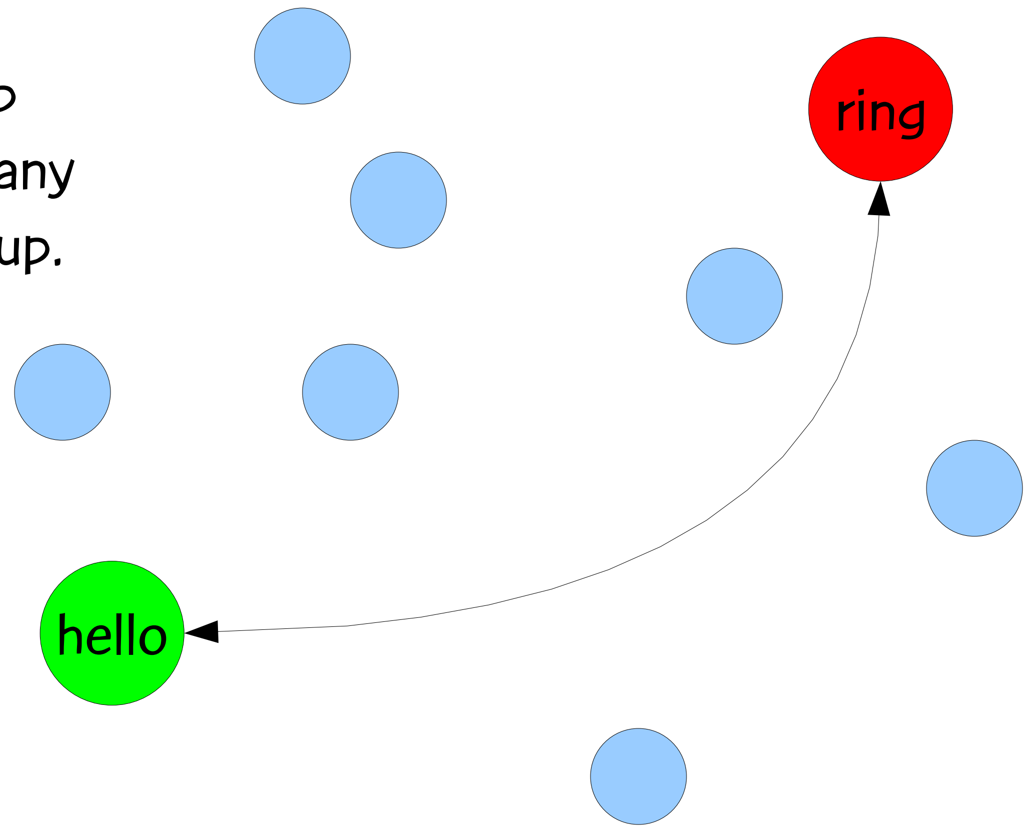
Call pickup (CPU)

Any phone in group can pickup call for any other phone in group.



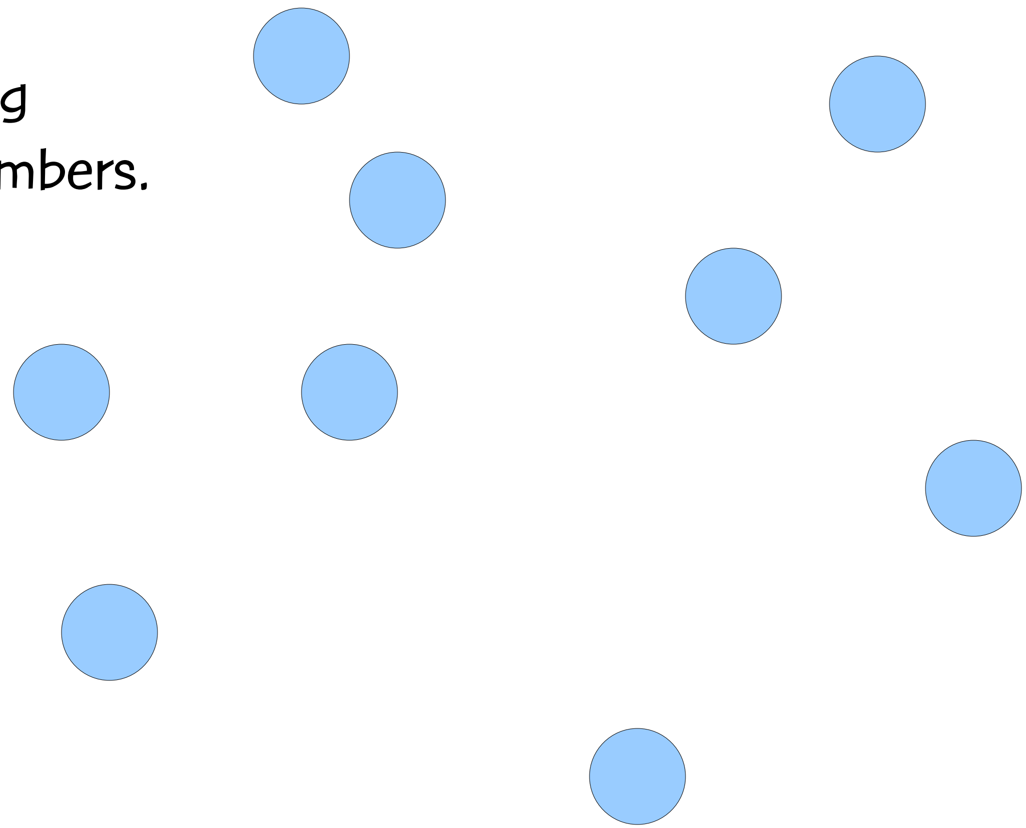
Call pickup (CPU)

Any phone in group
can pickup call for any
other phone in group.



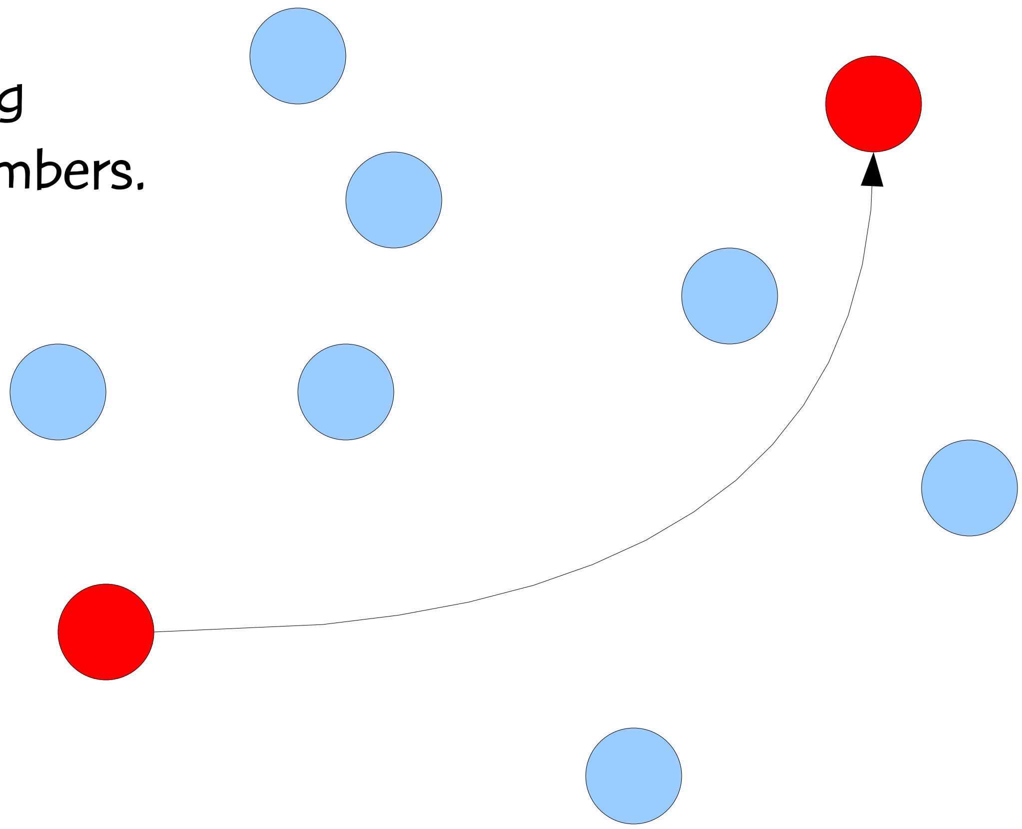
Intercomm (ICOM)

Allows speed dialing
between group members.



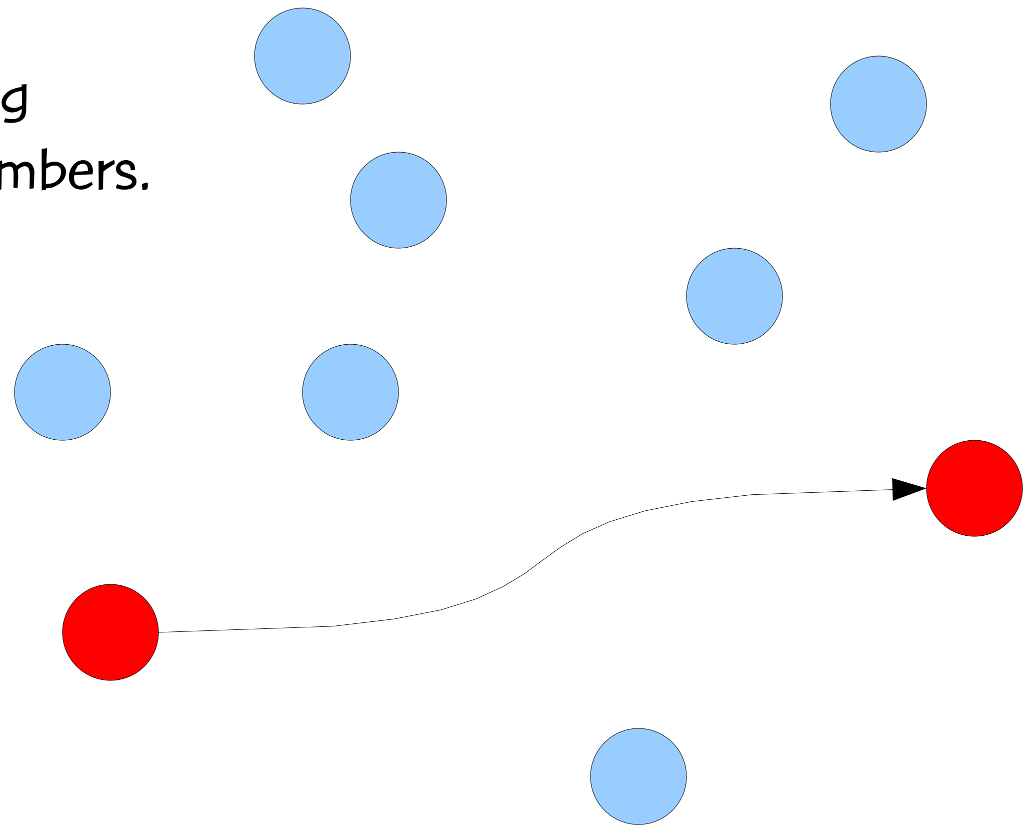
Intercomm (ICOM)

Allows speed dialing
between group members.



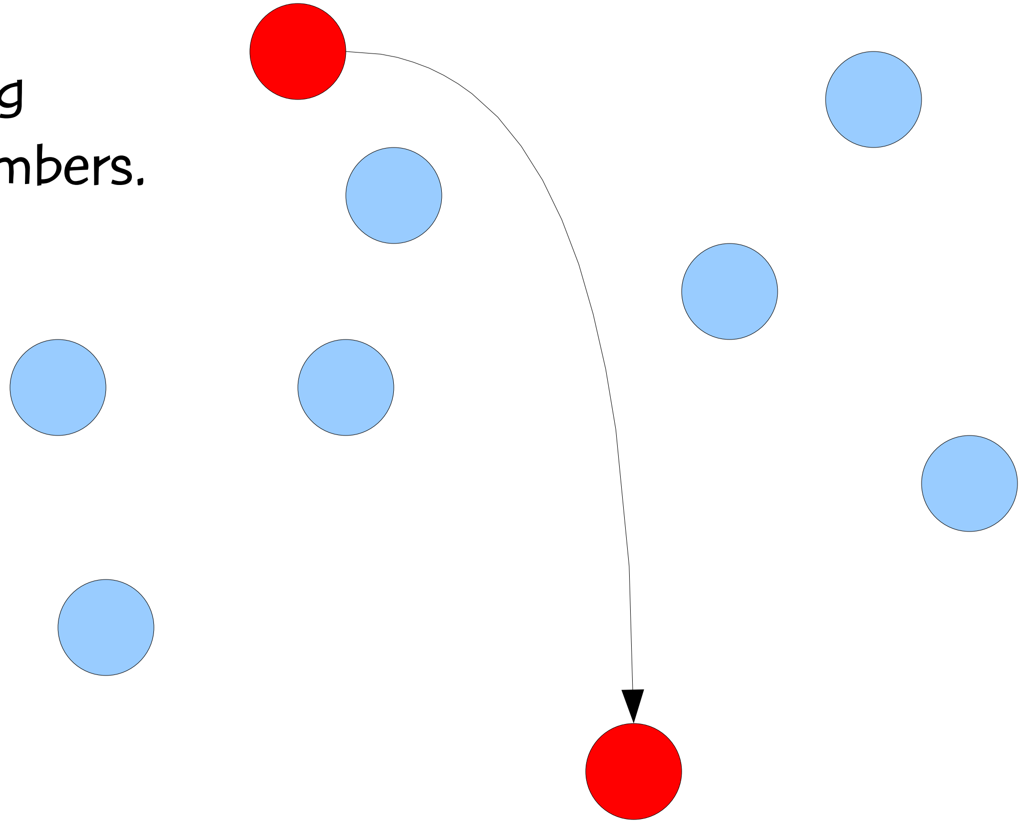
Intercomm (ICOM)

Allows speed dialing between group members.

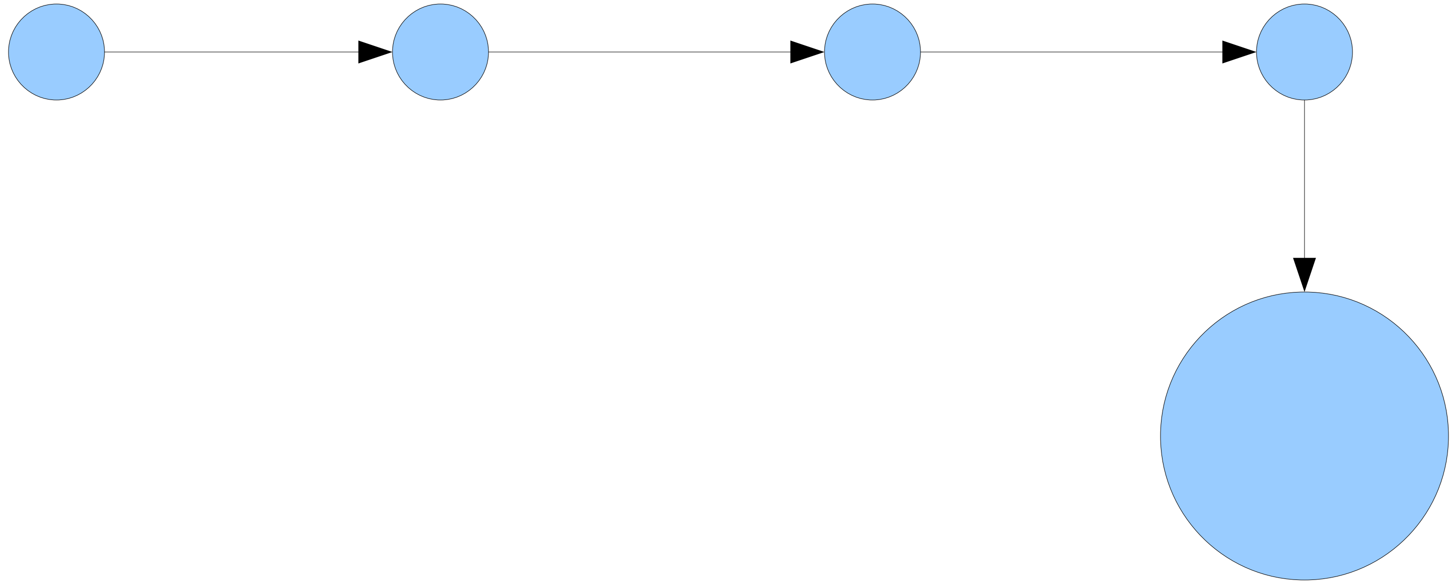


Intercomm (ICOM)

Allows speed dialing between group members.

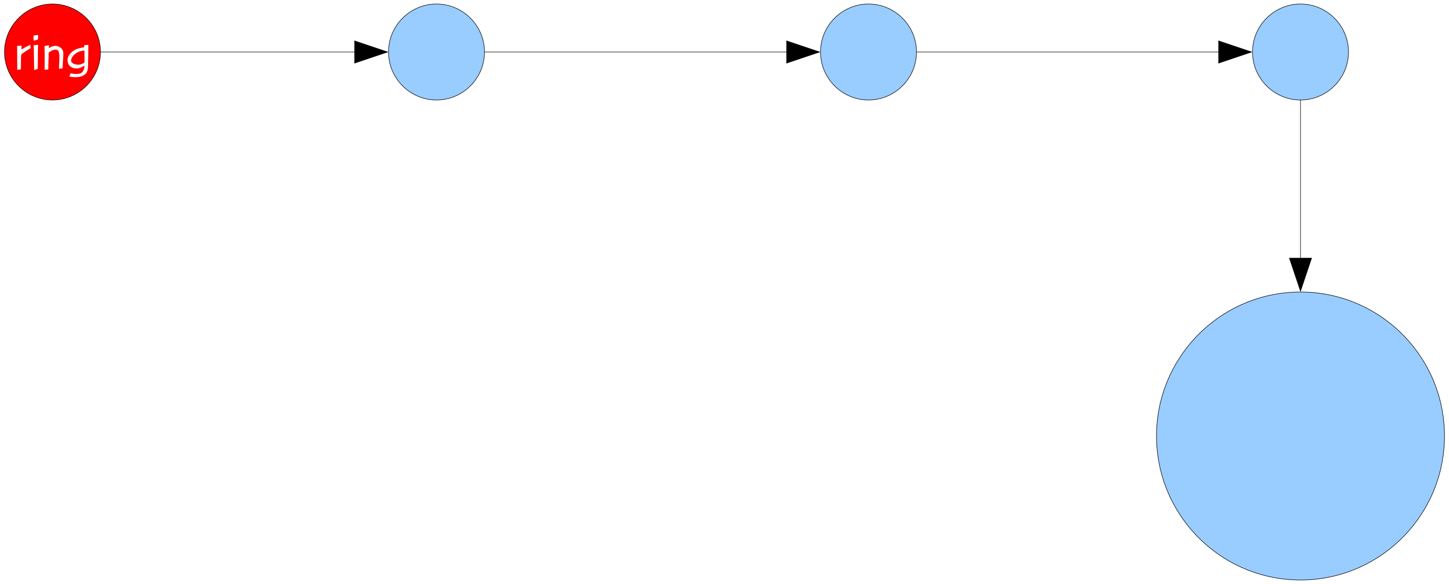


Series completion



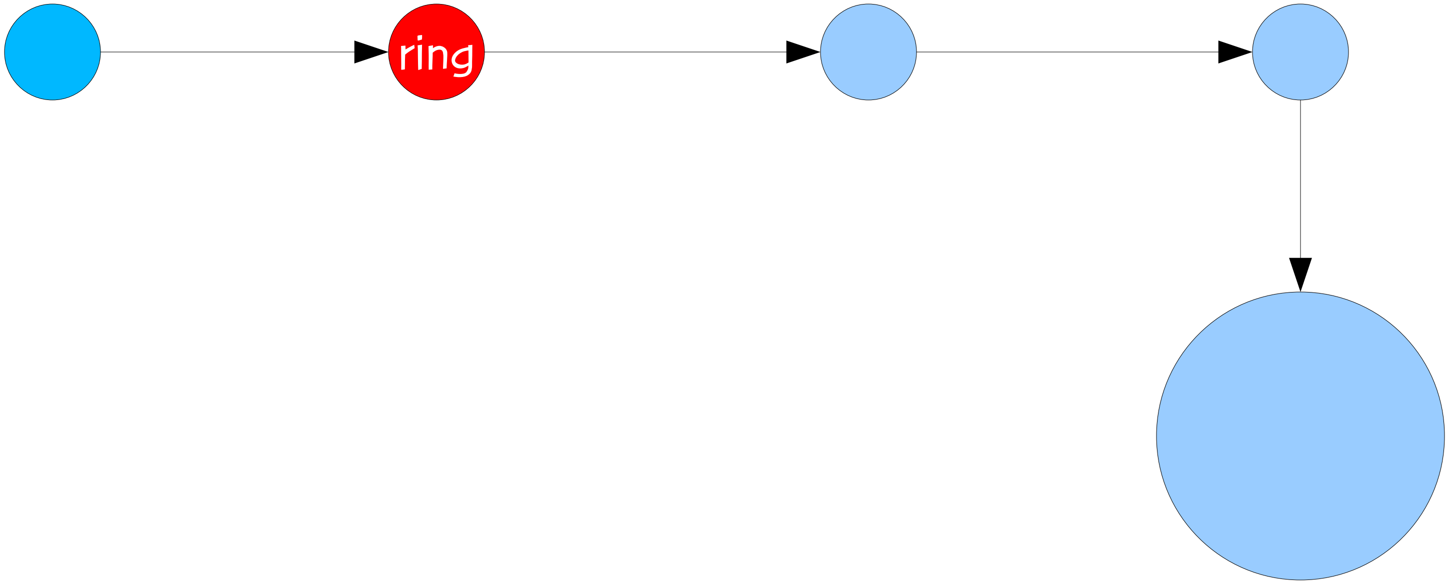
Series completion

If call not answered ...



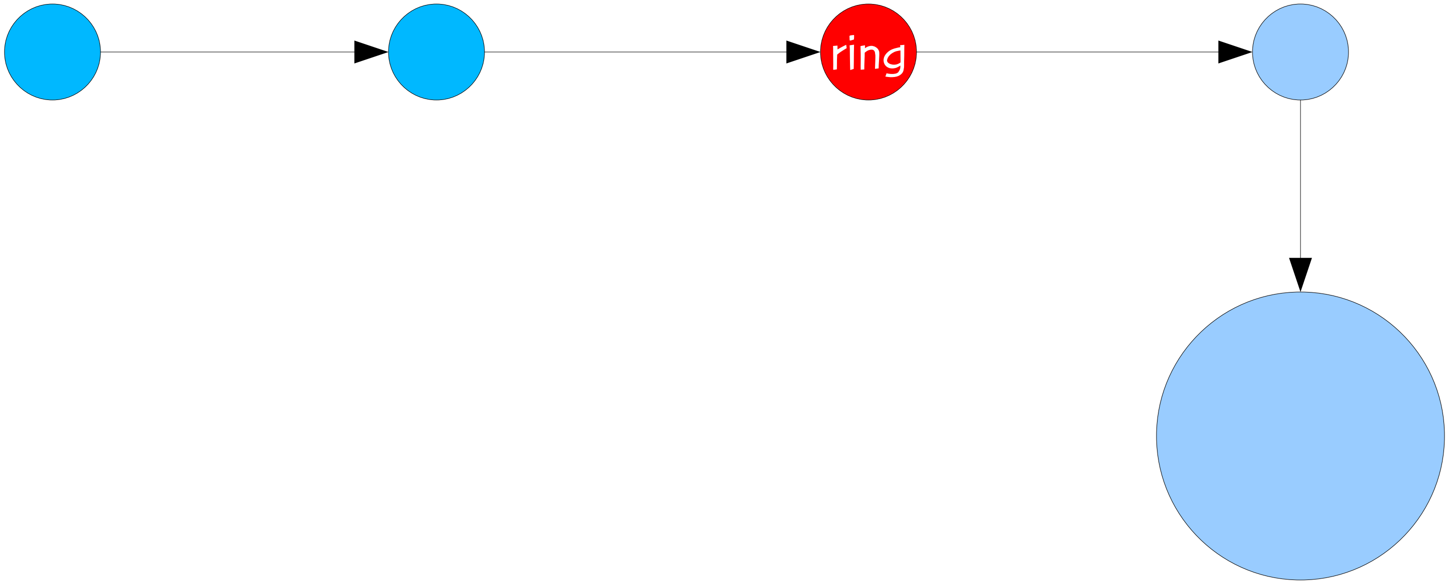
Series completion

If call not answered, it moves
to ....next in series.



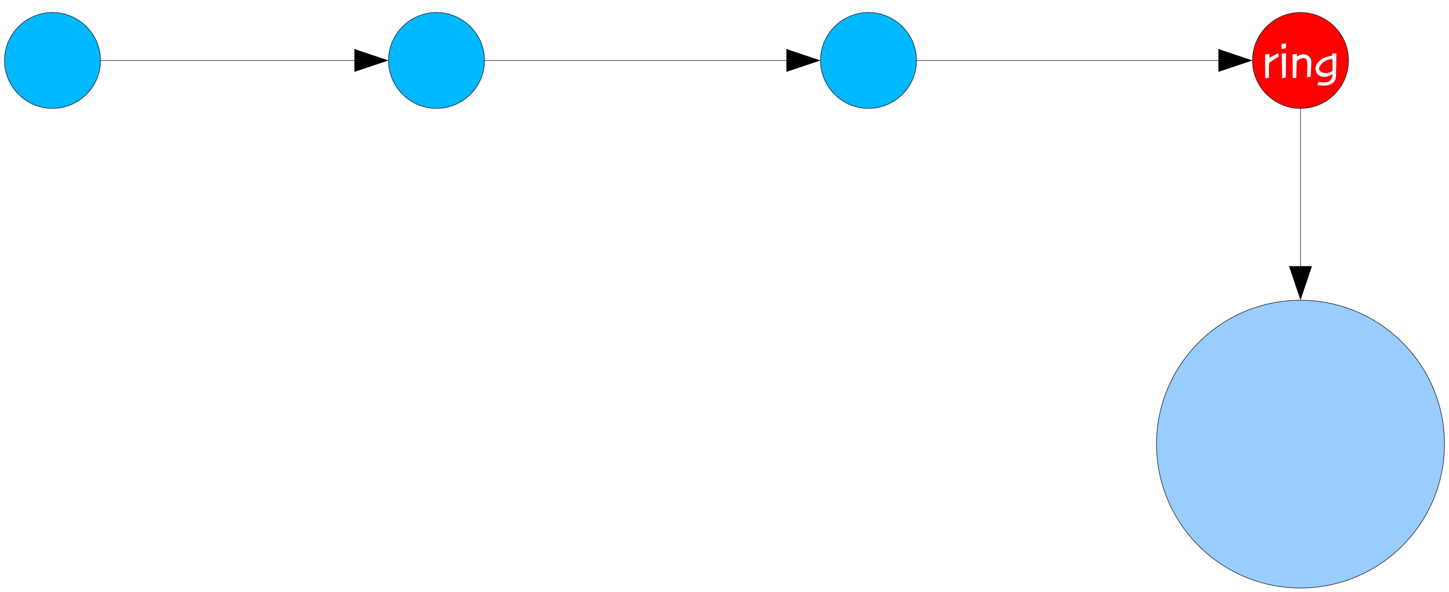
Series completion

If call not answered, it moves
to .... next in series.



Series completion

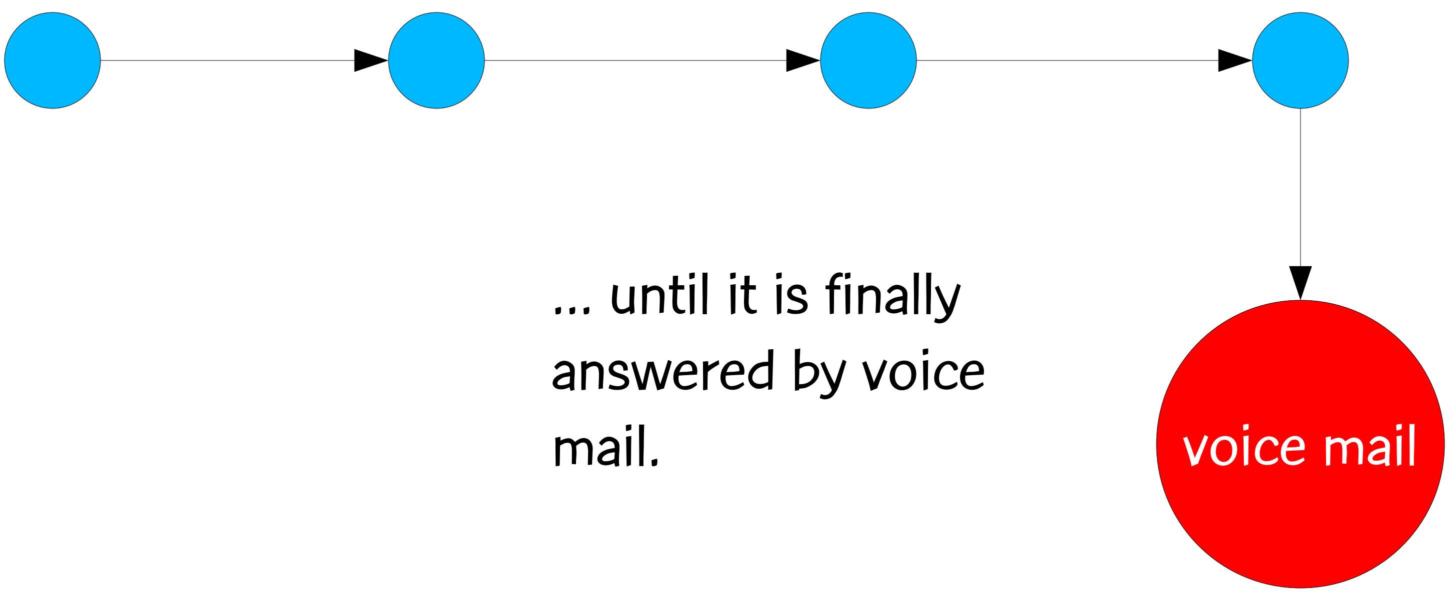
If call not answered, it moves
to .... next in series.



Series completion

If call not answered, it moves

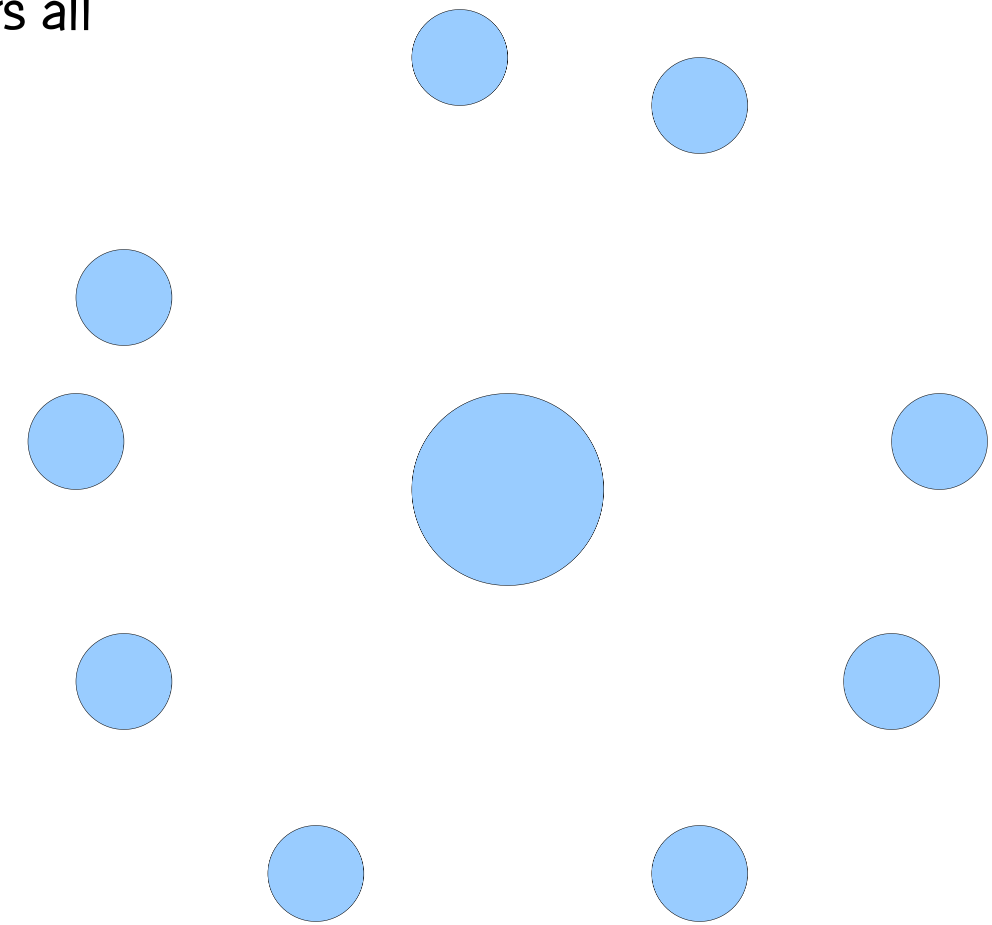
to .... next in series ...



... until it is finally answered by voice mail.

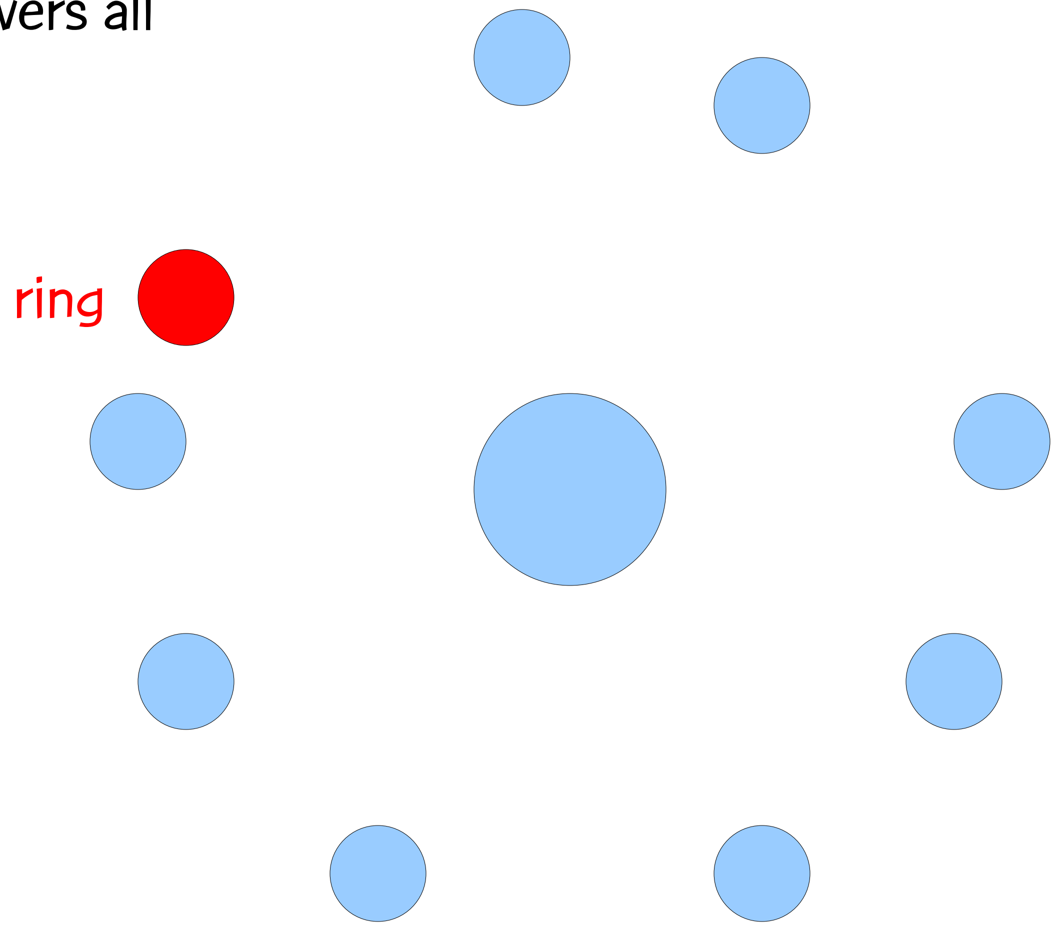
Shared TN

Assistant answers all calls to group.



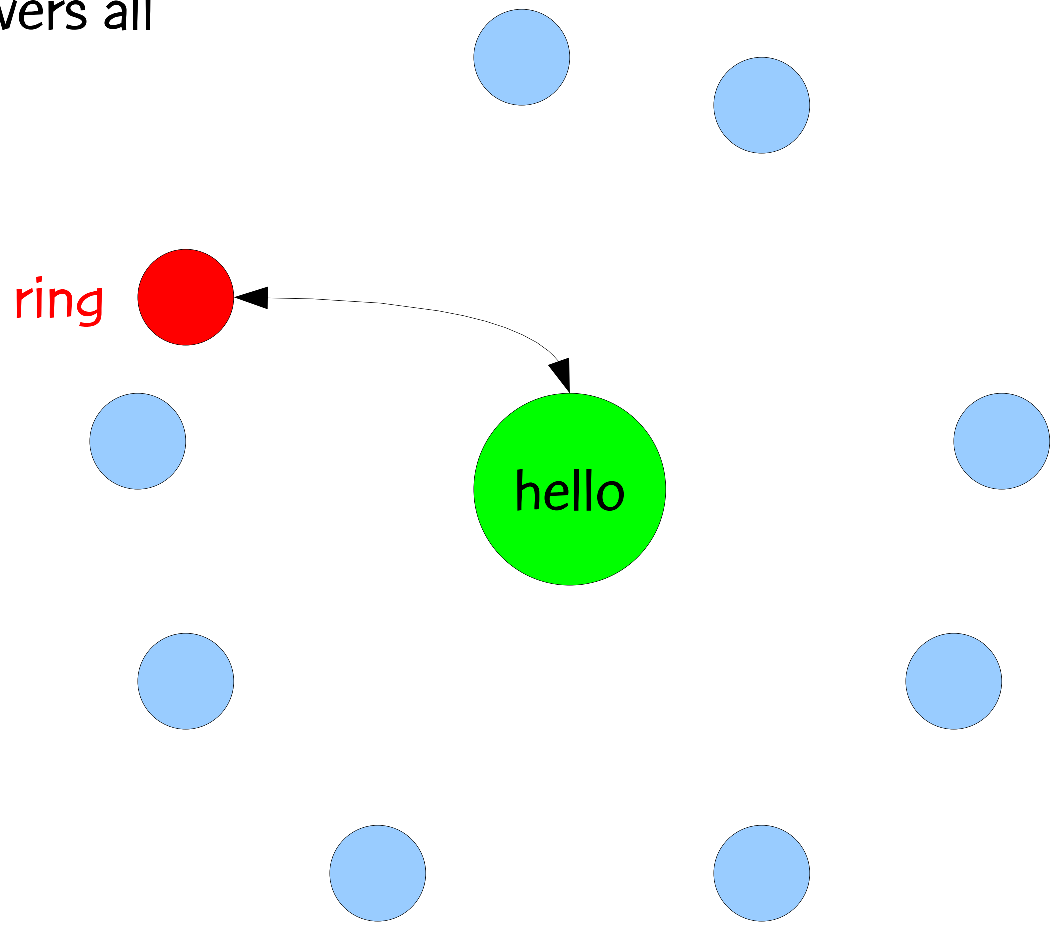
Shared TN

Assistant answers all calls to group.



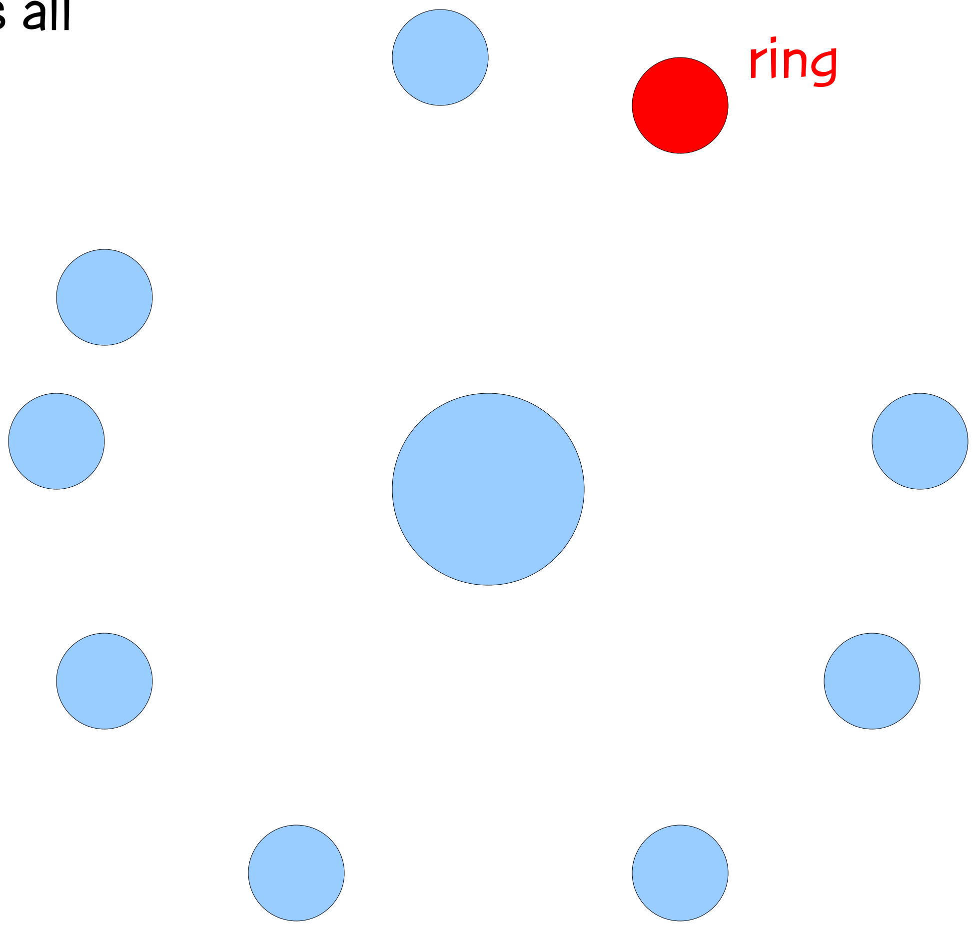
Shared TN

Assistant answers all calls to group.



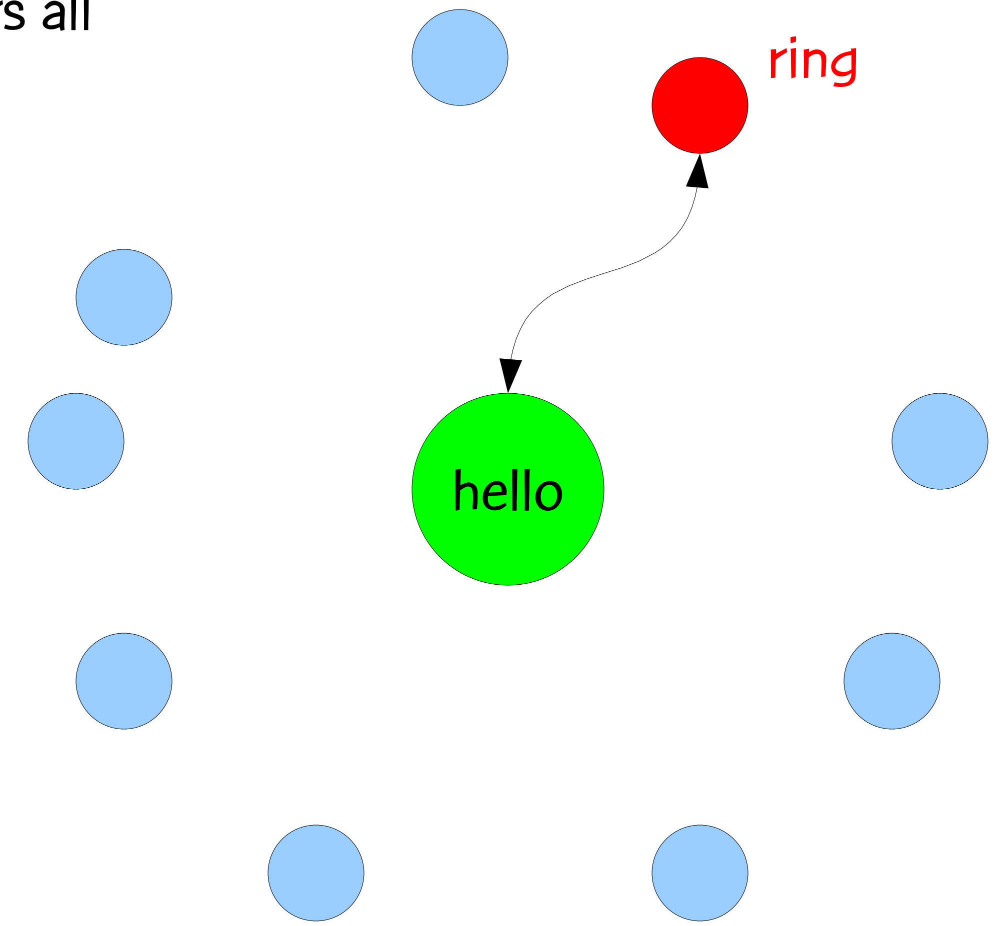
Shared TN

Assistant answers all calls to group.



Shared TN

Assistant answers all calls to group.



Real-world example

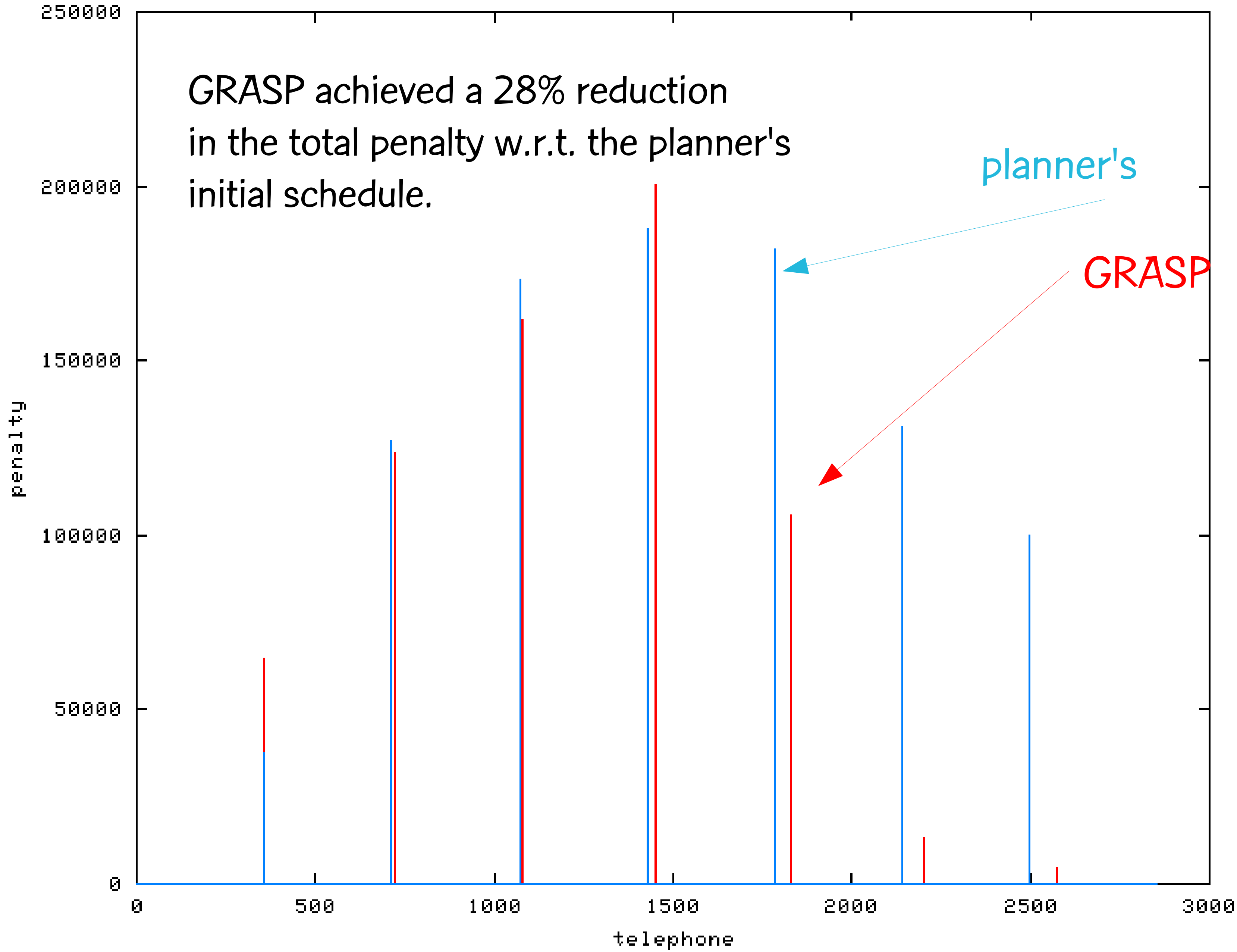
- AT&T sold new switch to large NYC-based investment bank.
- Post sales contacted us about improving a tentative schedule they had produced.
- Objective was to minimize business disruption during migration to new switch.
- Problem size:
  - 2855 phone numbers, 397 groups.
  - At most 375 phones could be moved in a period: 8 periods.

Real-world example

- Objective: Minimize disruption by minimizing total migration penalty
- Groups and Penalties:
  - Multi-Line Hunt : 10
  - Call Pickup : 4
  - Intercomm : 3
  - Series Completion: 2
  - Shared TN : 1

Real-world example

- Objective: Minimize disruption by minimizing total migration penalty
- Groups and Penalties:
  - Multi-Line Hunt : 10
  - Call Pickup : 4
  - Intercomm : 3
  - Series Completion: 2
  - Shared TN : 1


GRASP achieved a 28% reduction
in the total penalty w.r.t. the planner's
initial schedule.
planner's
GRASP
250000
200000
150000
100000
50000
0
penalty
0
500
1000
1500
2000
2500
3000
telephone

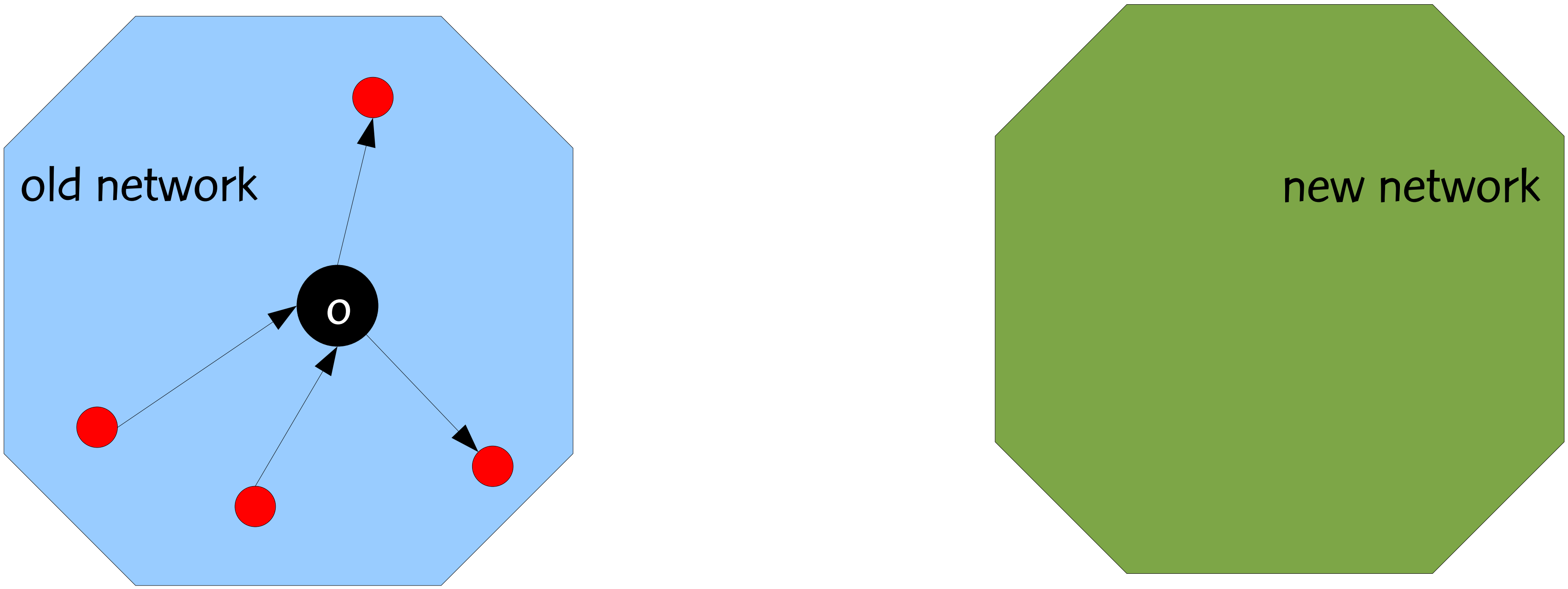
- D.V. Andrade and M.G.C.R., "A GRASP for PBX telephone migration scheduling," Proceedings of The Eighth INFORMS Telecommunications Conference, Dallas, Texas, April 2006

Network traffic migration scheduling

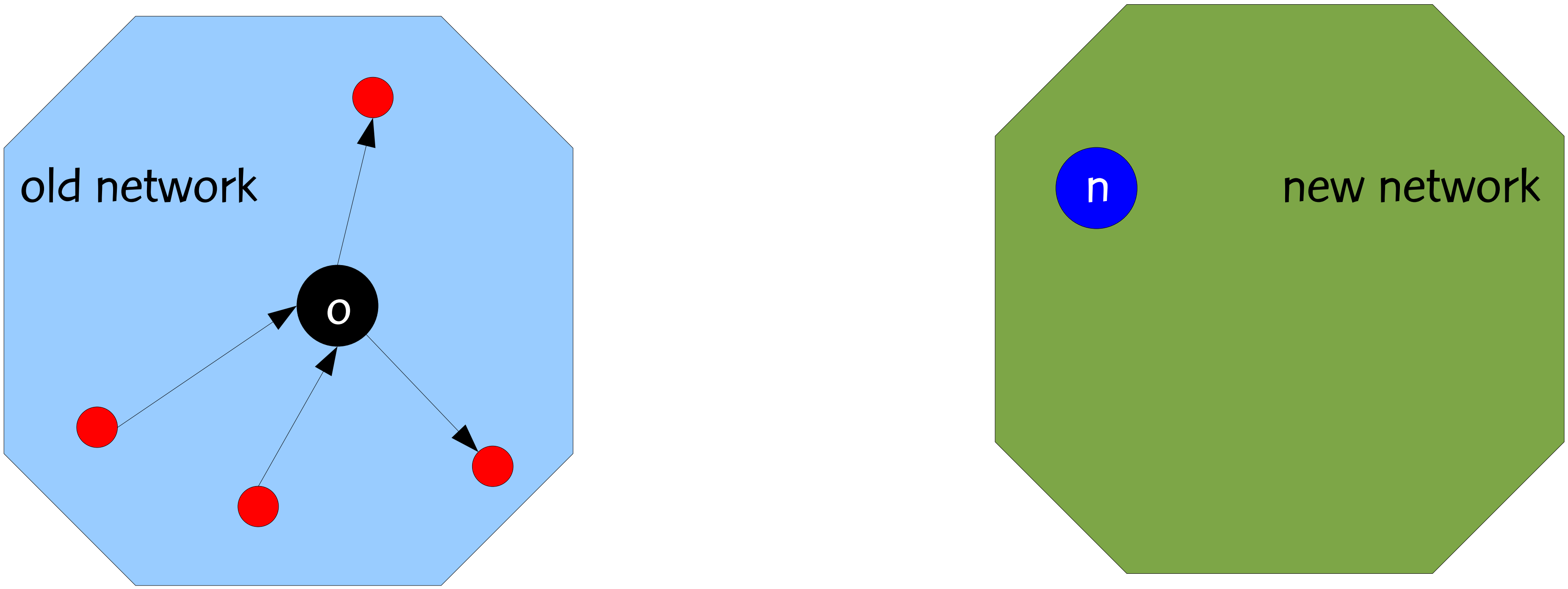
Network traffic migration scheduling

- 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.

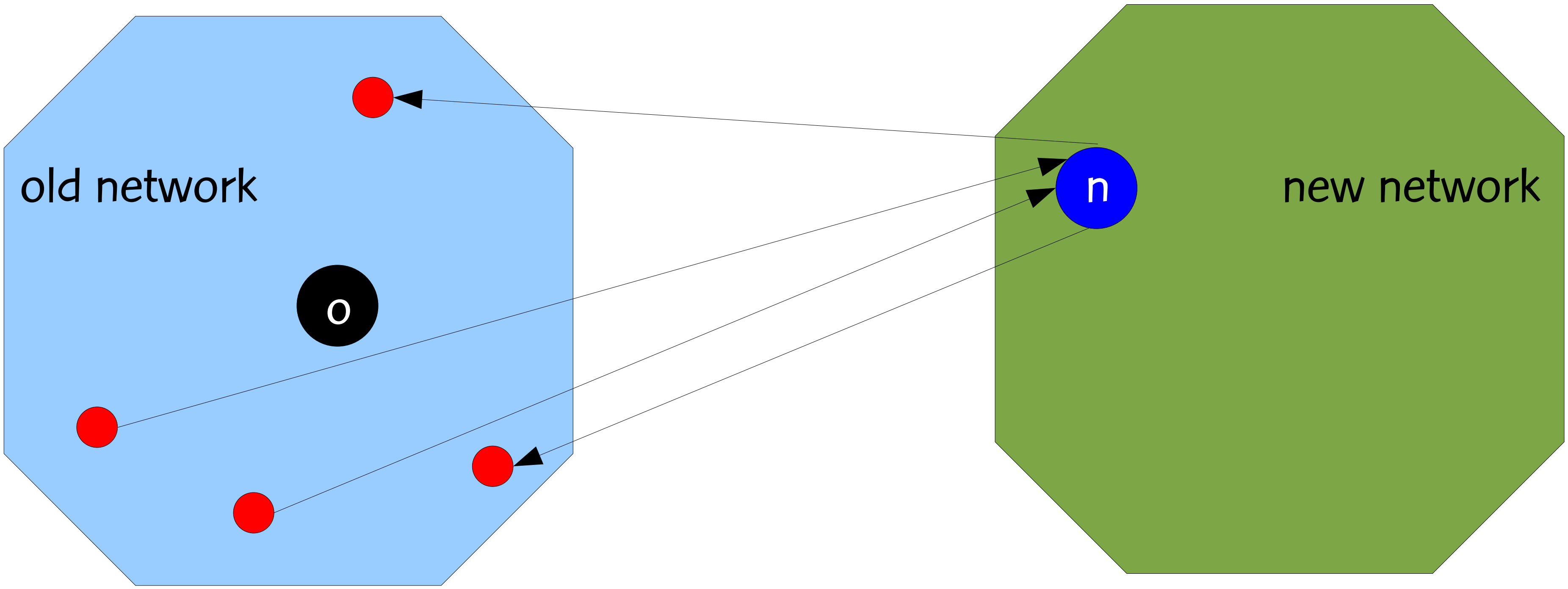
Node decommissioning



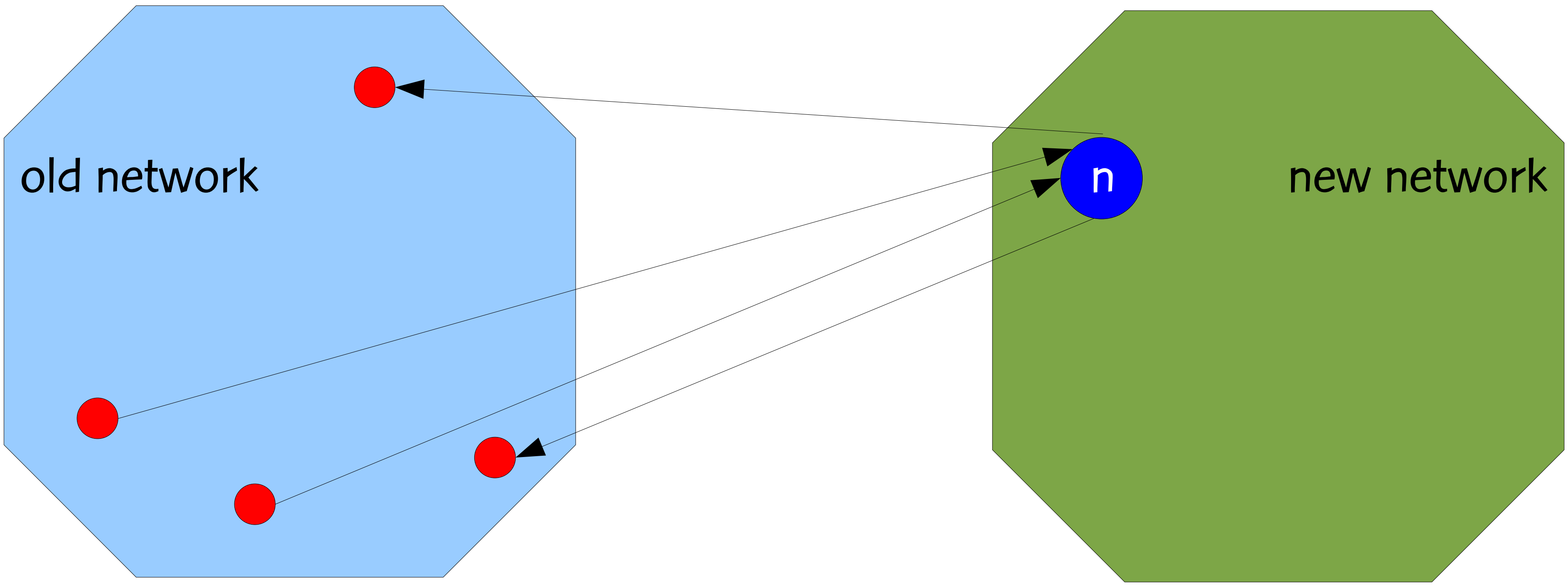
Node decommissioning



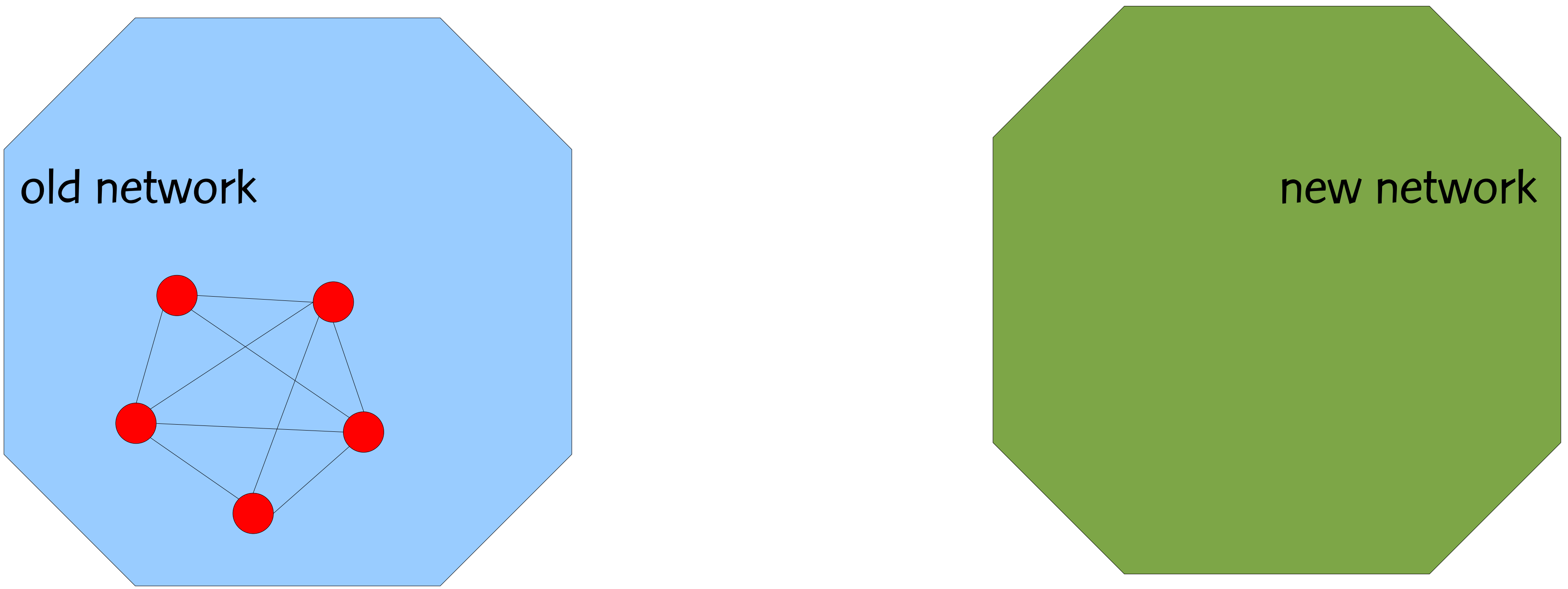
Node decommissioning



Node decommissioning

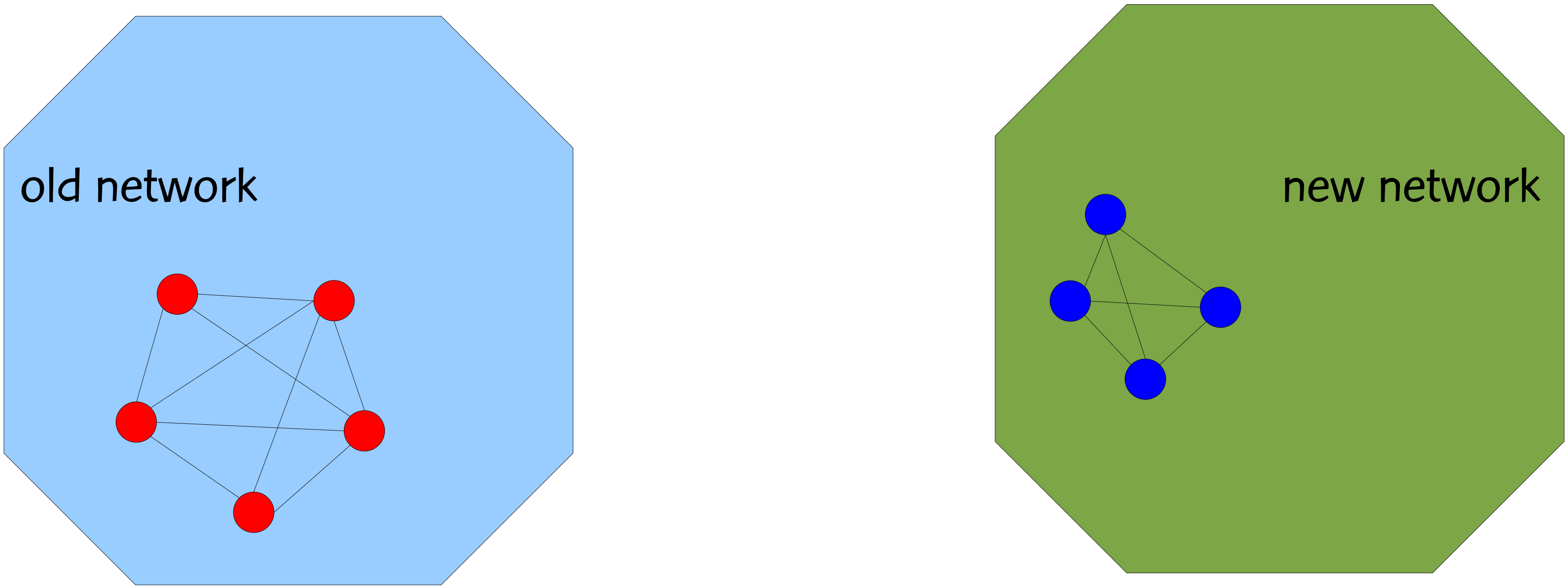


After partial decommissioning of nodes



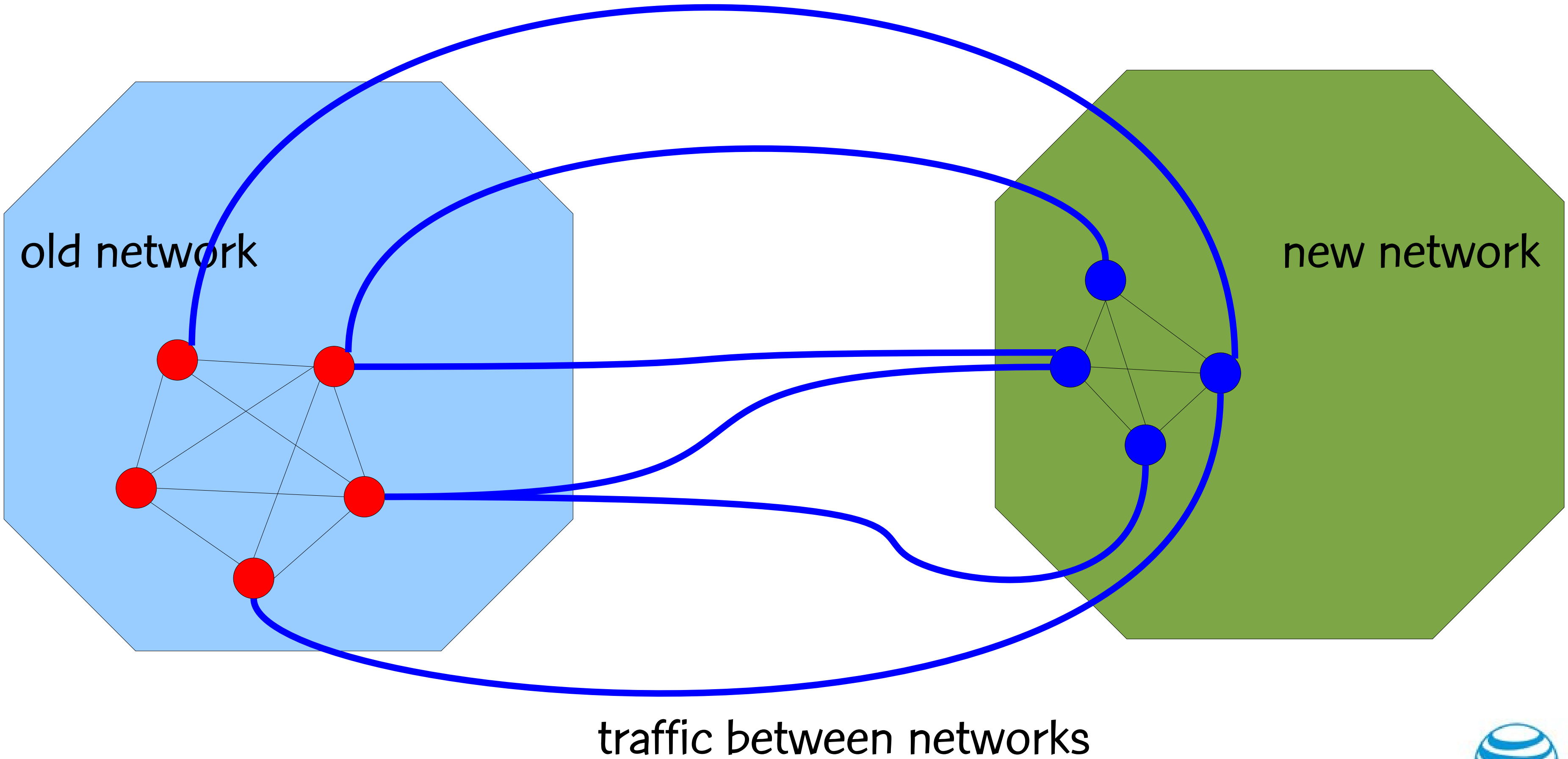
traffic in old network

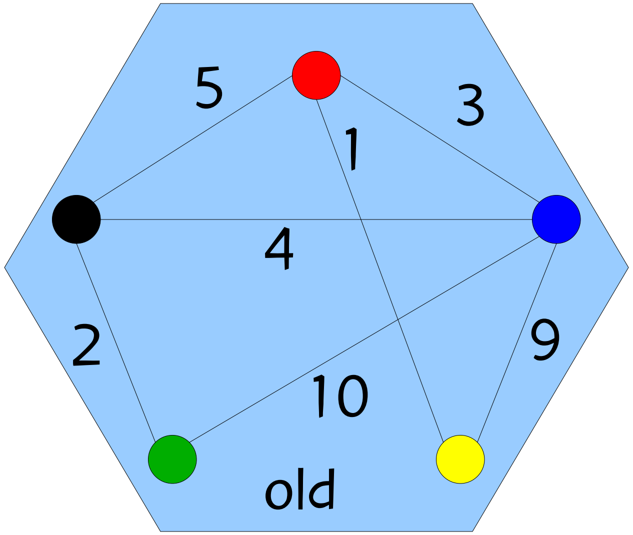
After partial decommissioning of nodes

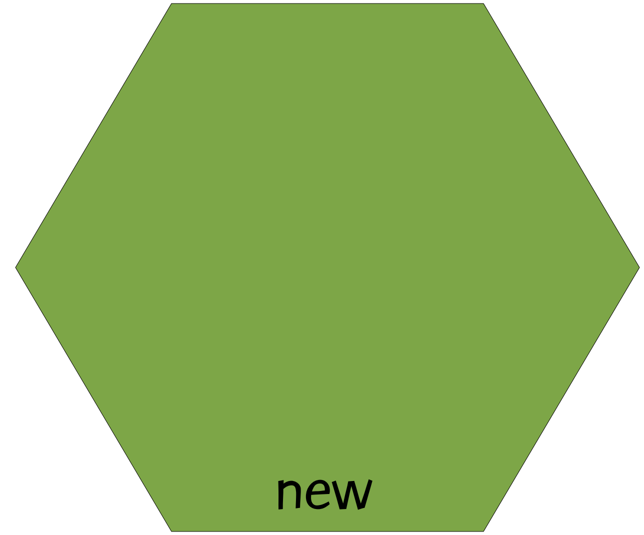


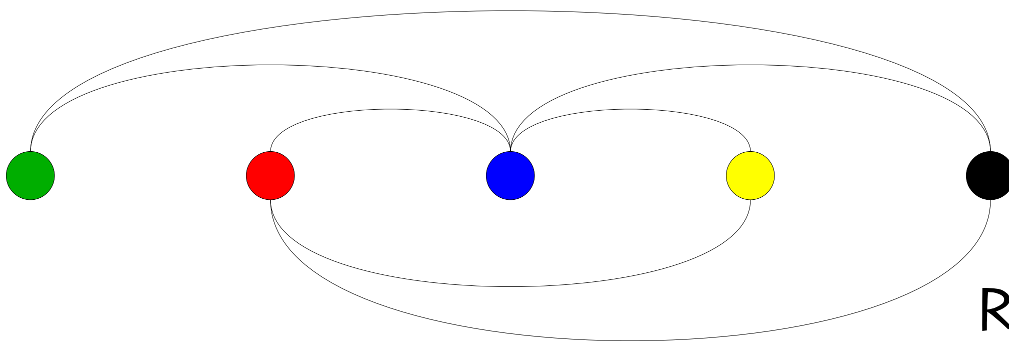
traffic in new network

After partial decommissioning of nodes

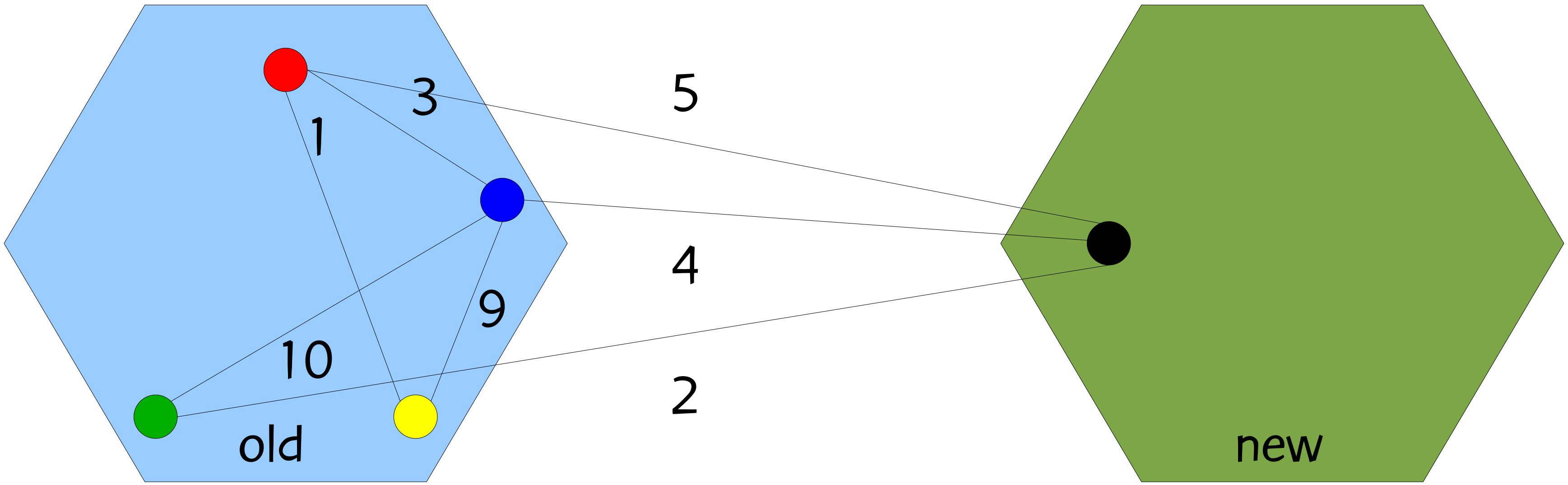


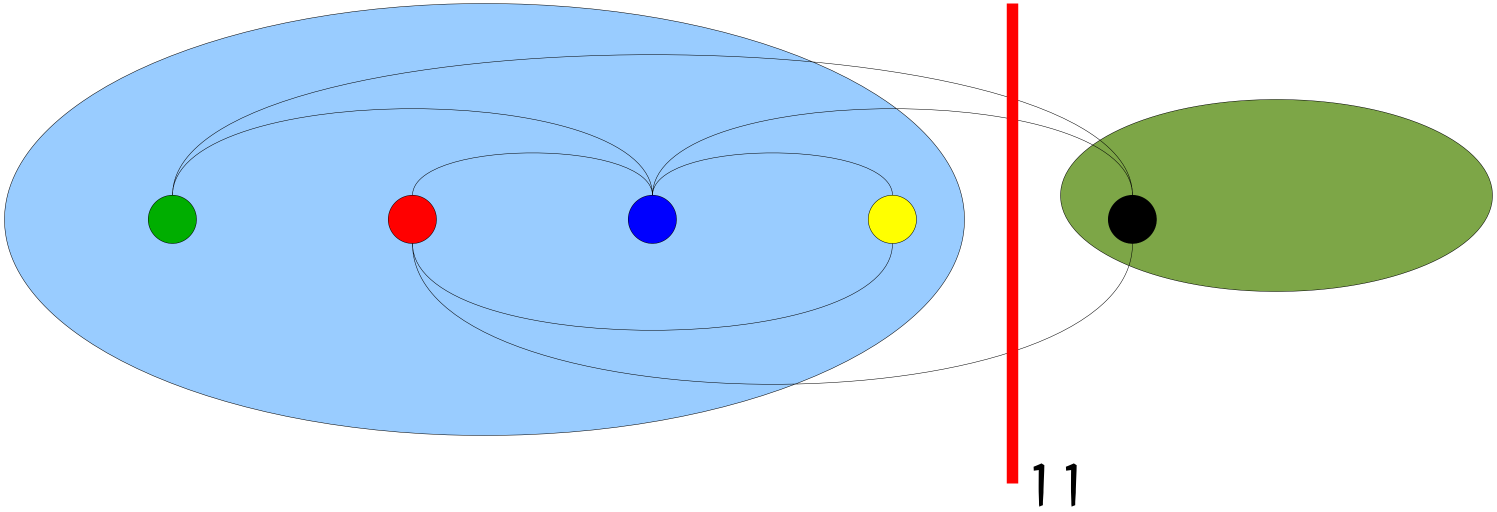


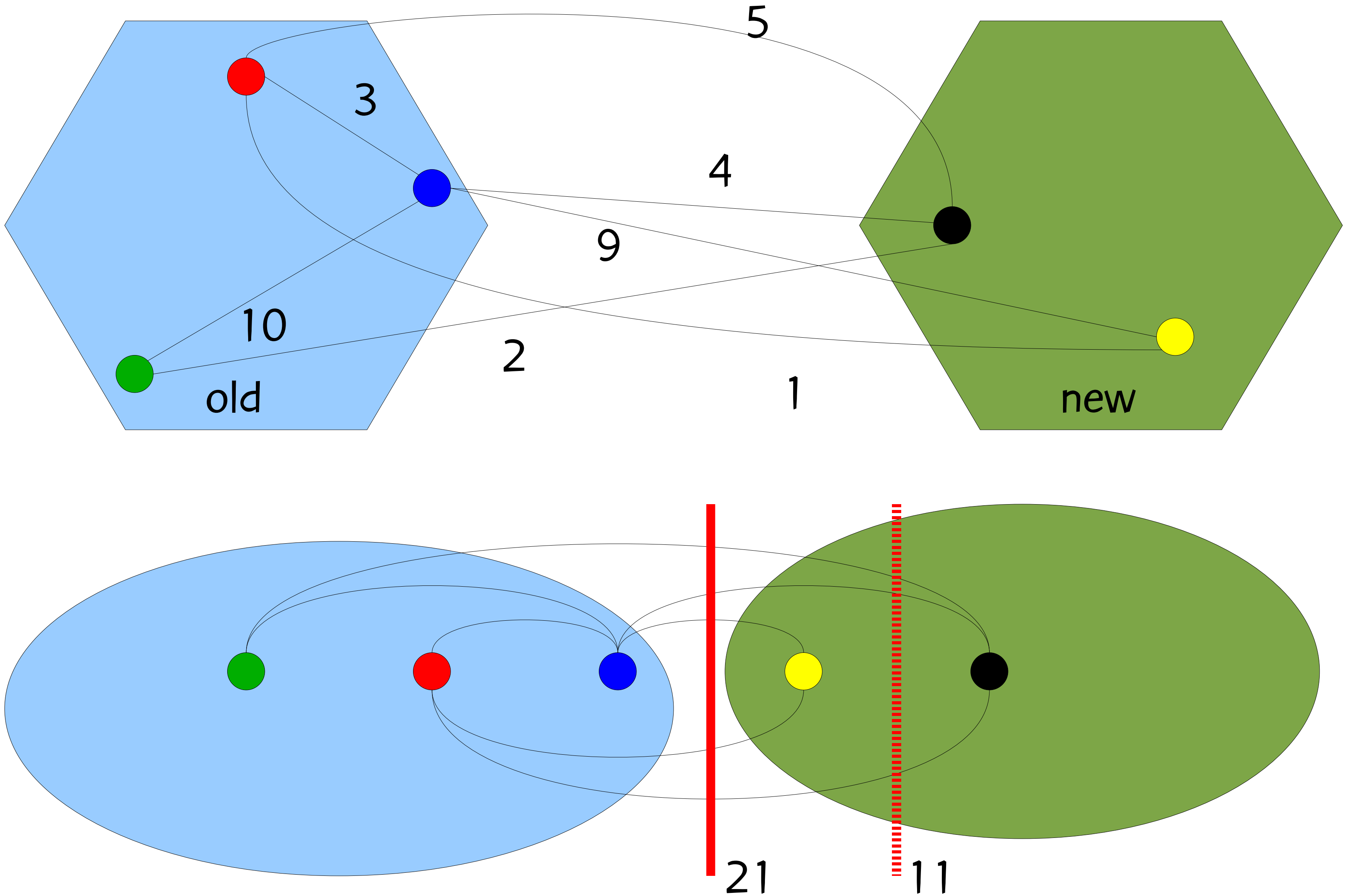


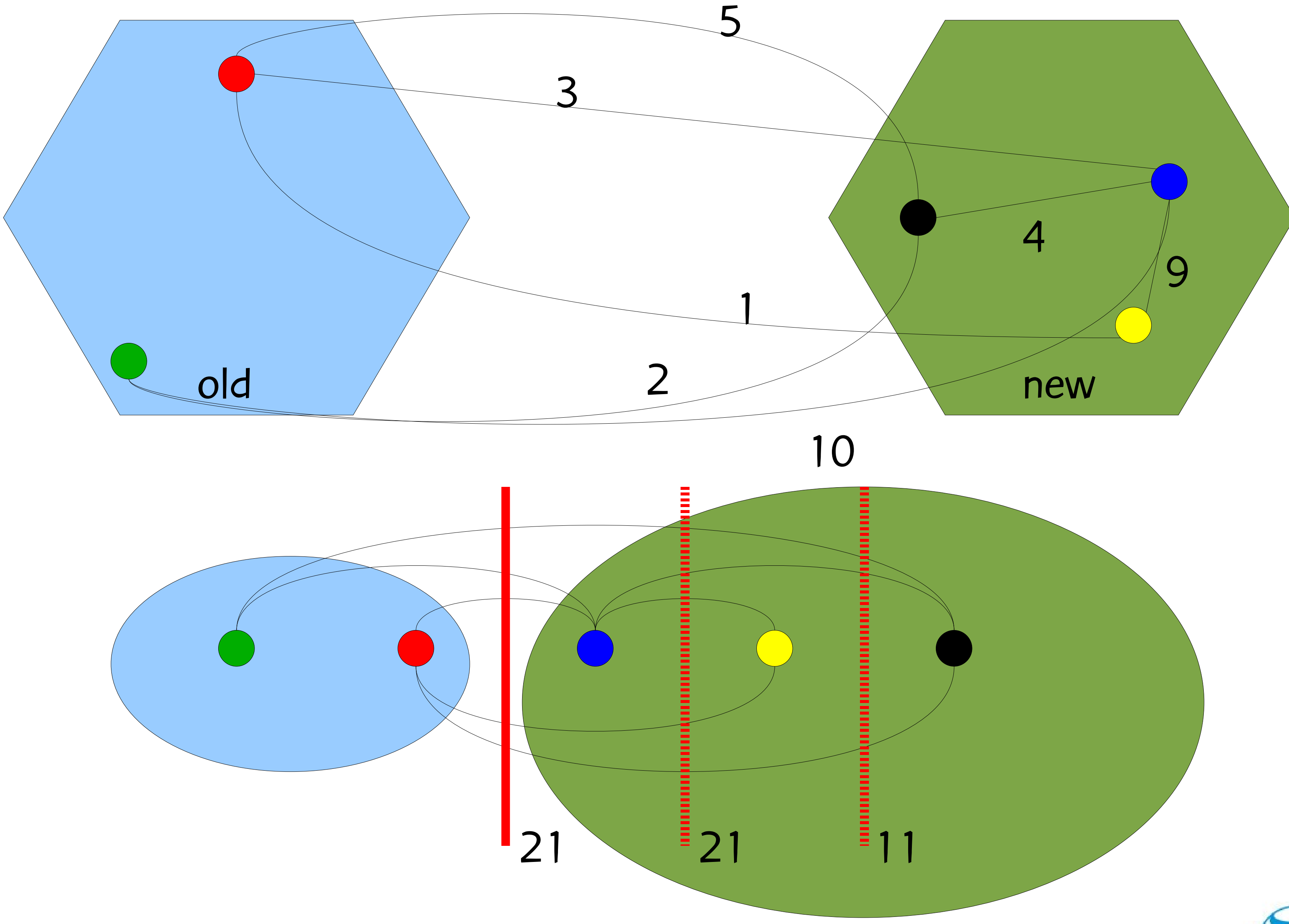


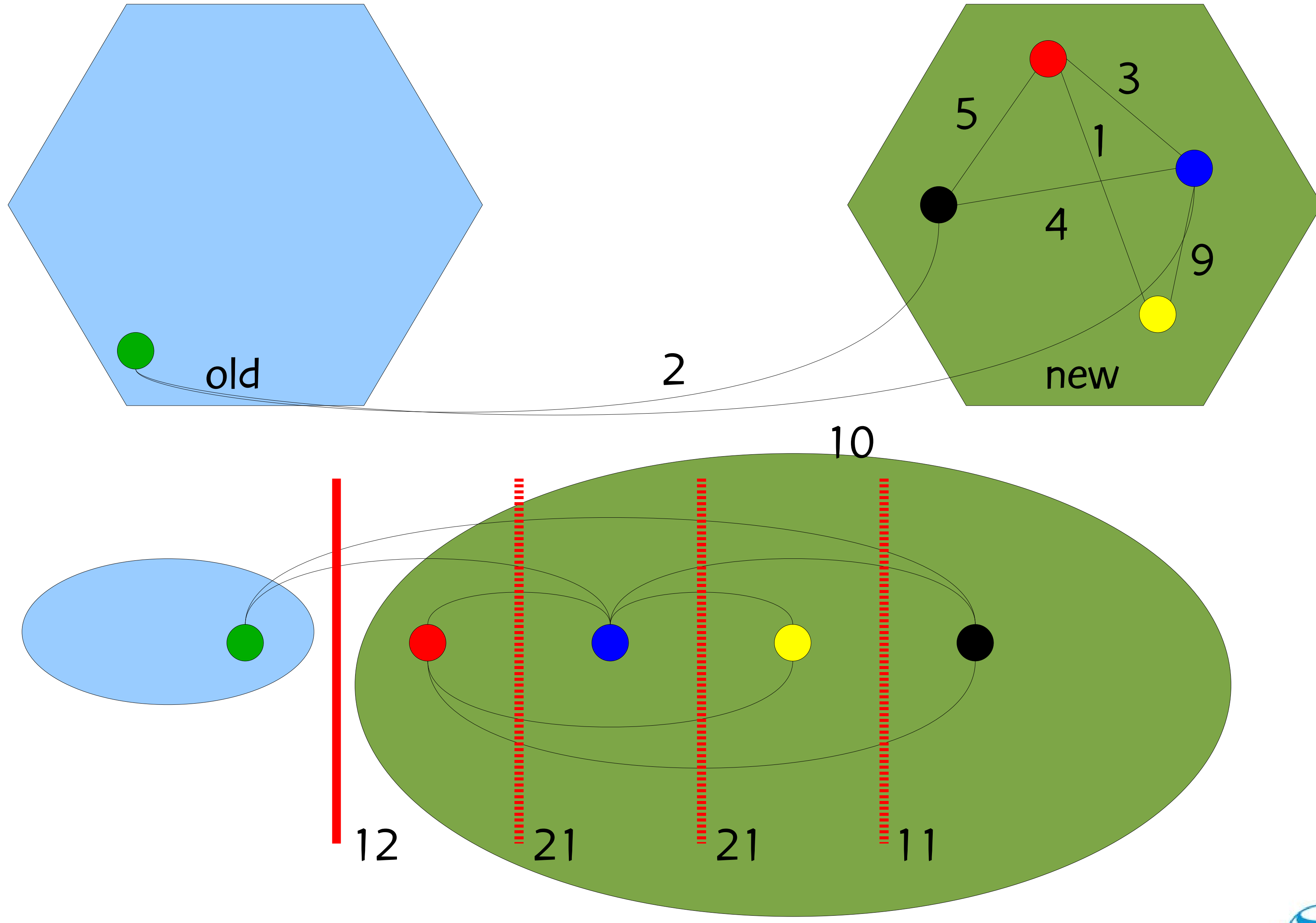
Redraw graph with nodes in line giving order in which nodes are migrated.

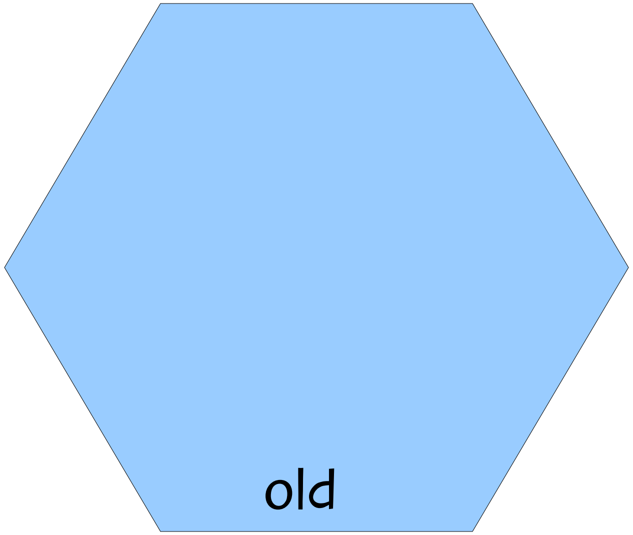

3
5
1
4
9
10
2
old
new

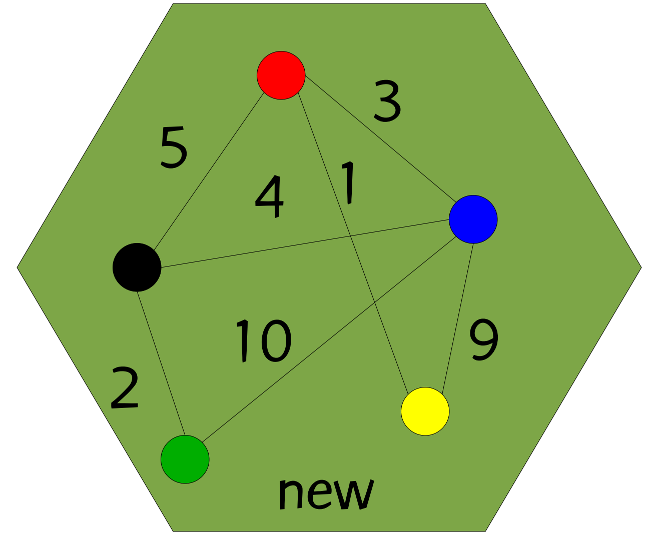

11

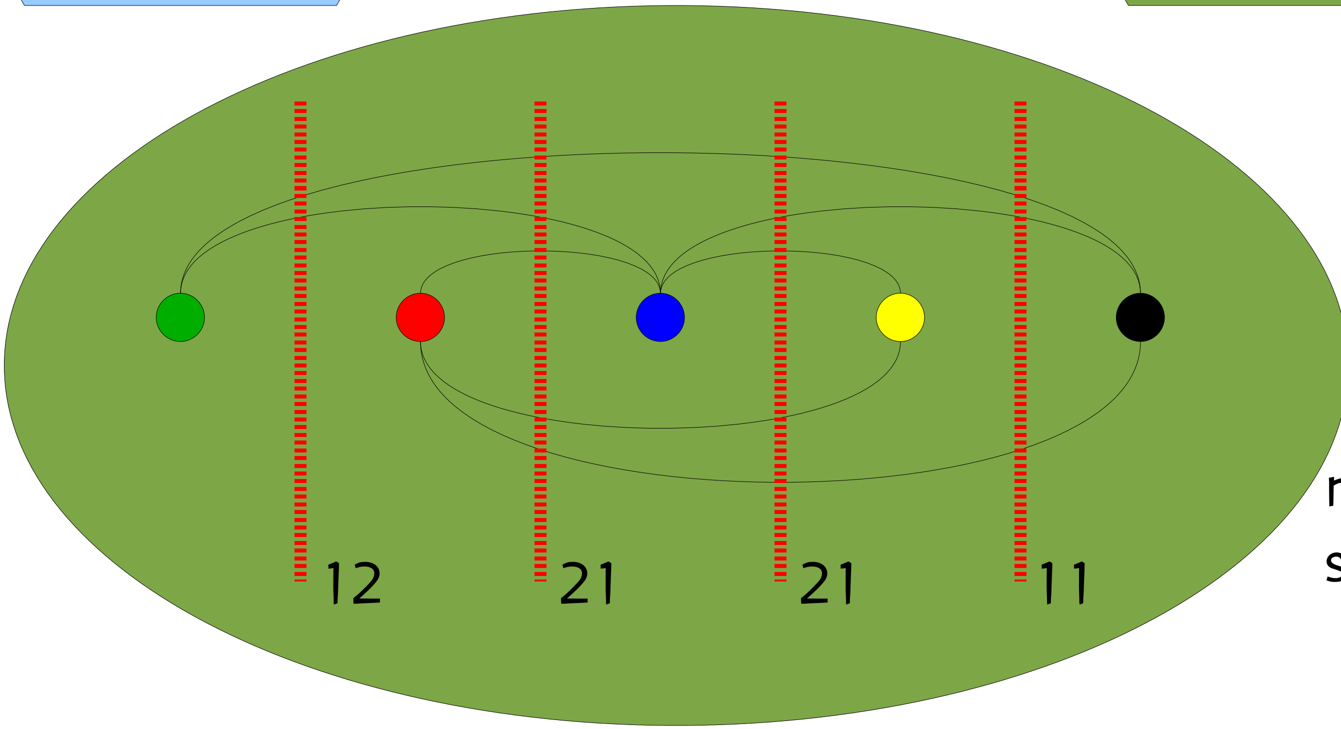

5
3
4
9
10
2
1
old
new
21
11


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


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

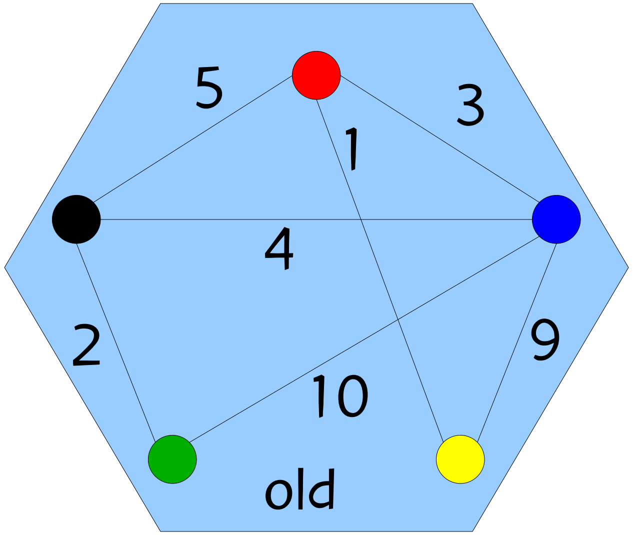




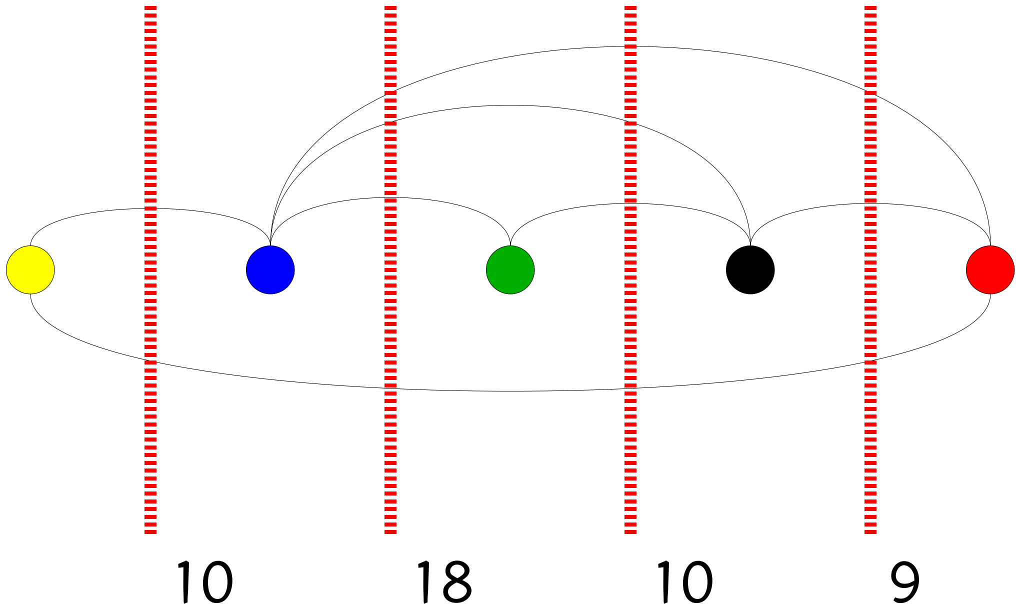


max = 21

sum = 65



Consider another ordering.

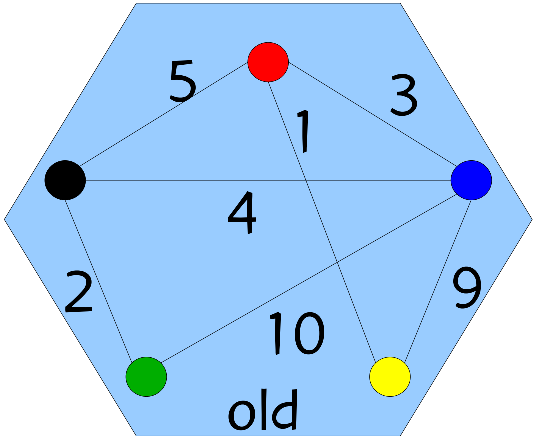


max = 18 < 21

sum = 47 < 65

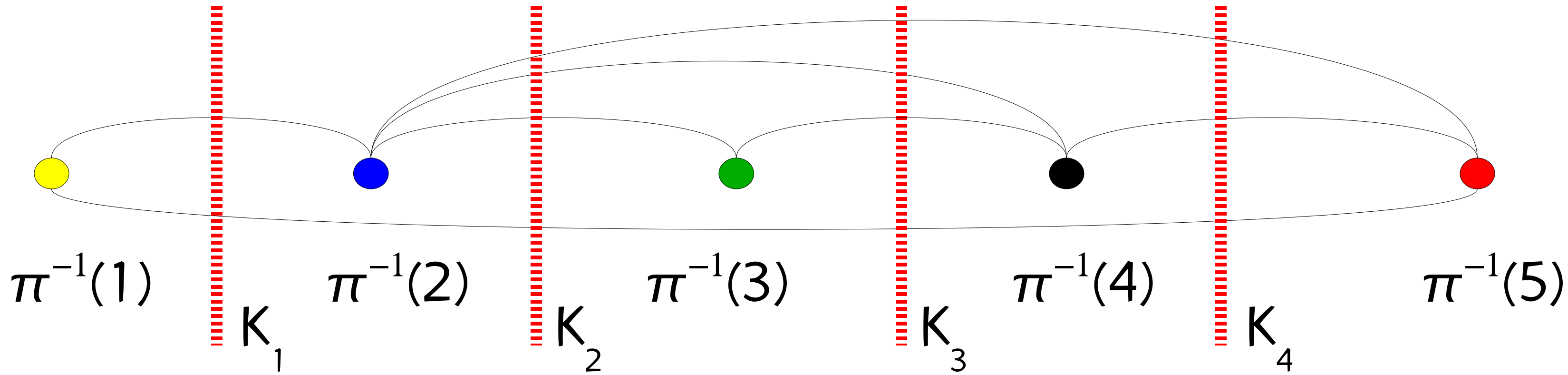
Optimization problem

Given an edge-weighted graph G = (V,E,w), where node set V is the set of switches (routers), edge set E is the set of links between switches (routers), and w is the traffic volume on the links.



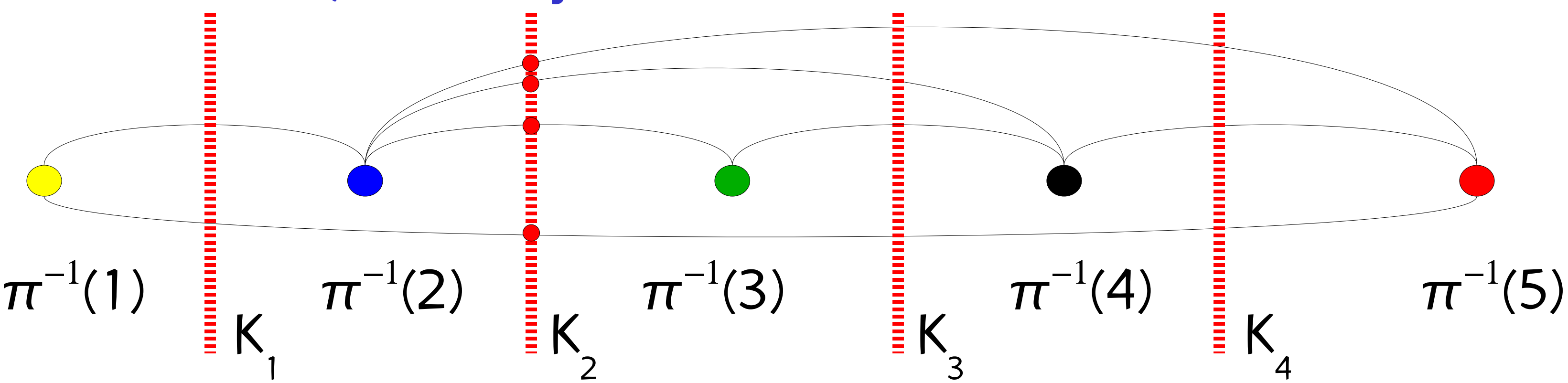
Optimization problem

- Let π: V $\rightarrow$ {1,...,n = |V| } be an ordering of the switches.
- For $1 \leq i < n$, let cut K_i between nodes $\pi^{-1}(i)$ and $\pi^{-1}(i+1)$ be the sum of the weights of all links with one endpoint in $\pi^{-1}(j)$ and the other in $\pi^{-1}(k)$, for all $j \leq i$ and all $k > i$.



Optimization problem

- Let π: $V \rightarrow \{1,\ldots,n = |V|\}$ be an ordering of the switches.
- For $1 \leq i < n$, let cut K_i between nodes $\pi^{-1}(i)$ and $\pi^{-1}(i+1)$ be the sum of the weights of all links with one endpoint in $\pi^{-1}(j)$ and the other in $\pi^{-1}(k)$, for all $j \leq i$ and all $k > i$.



Two measures of solution quality we want to optimize (minimize).

- The value K_{max} of the largest cut is the max-cut,

  i.e., $K_{max} = \max\{K_1,\ldots, K_{n-1}\}$.

- The value K_{sum} is the cut sum,

  i.e., $K_{sum} = K_1 + \cdots + K_{n-1}$.

Optimization problem

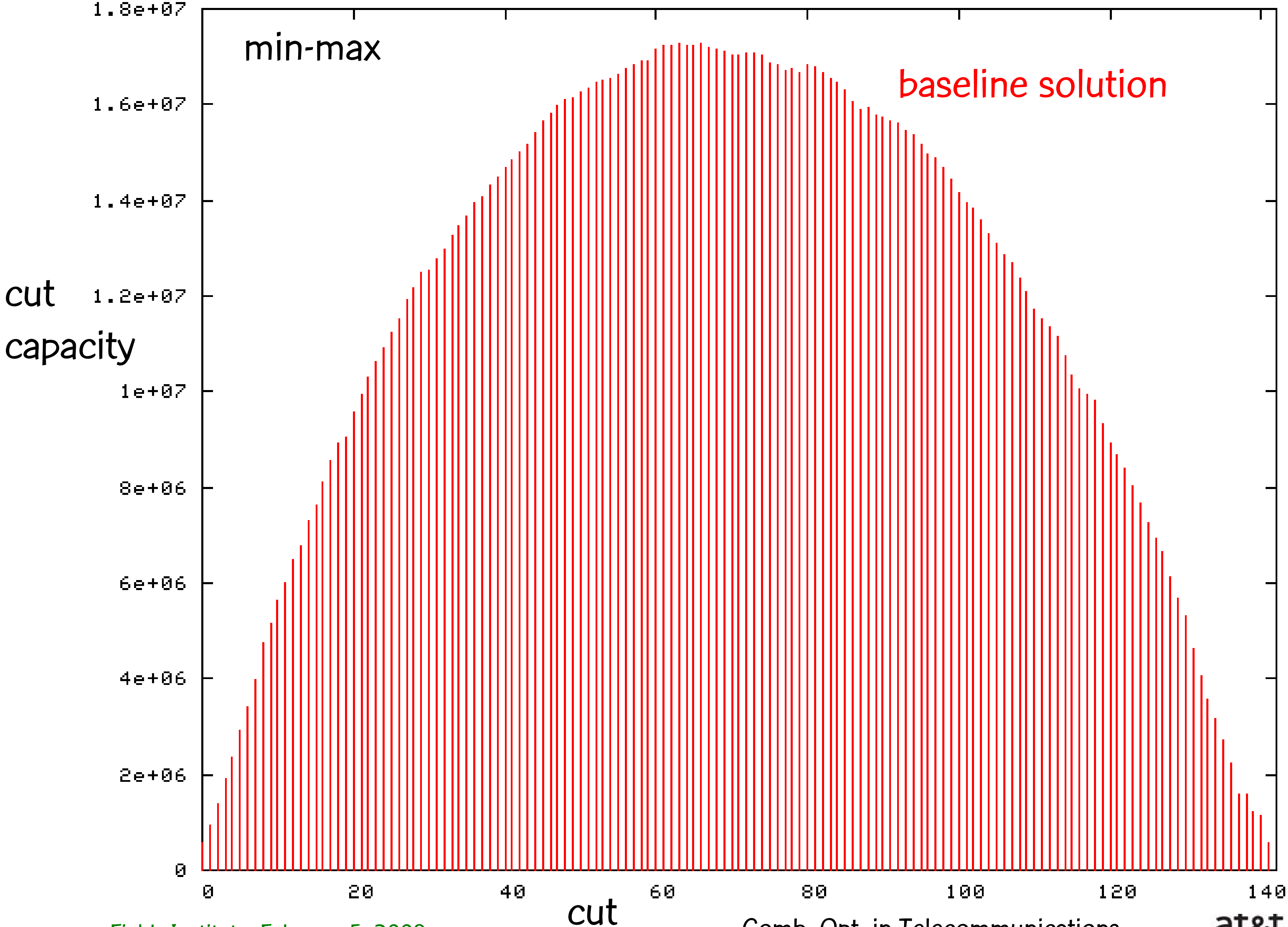
- This is an instance of batch scheduling of multi-grouped units:
  - each switch is a unit
  - each trunk $e \in E$ defines a group g_e
  - group g_e penalty is traffic w_e on trunk e
  - number of periods is $|V|$ (number of switches)
  - one switch is scheduled per period

Scheduling 4ESS deloading process

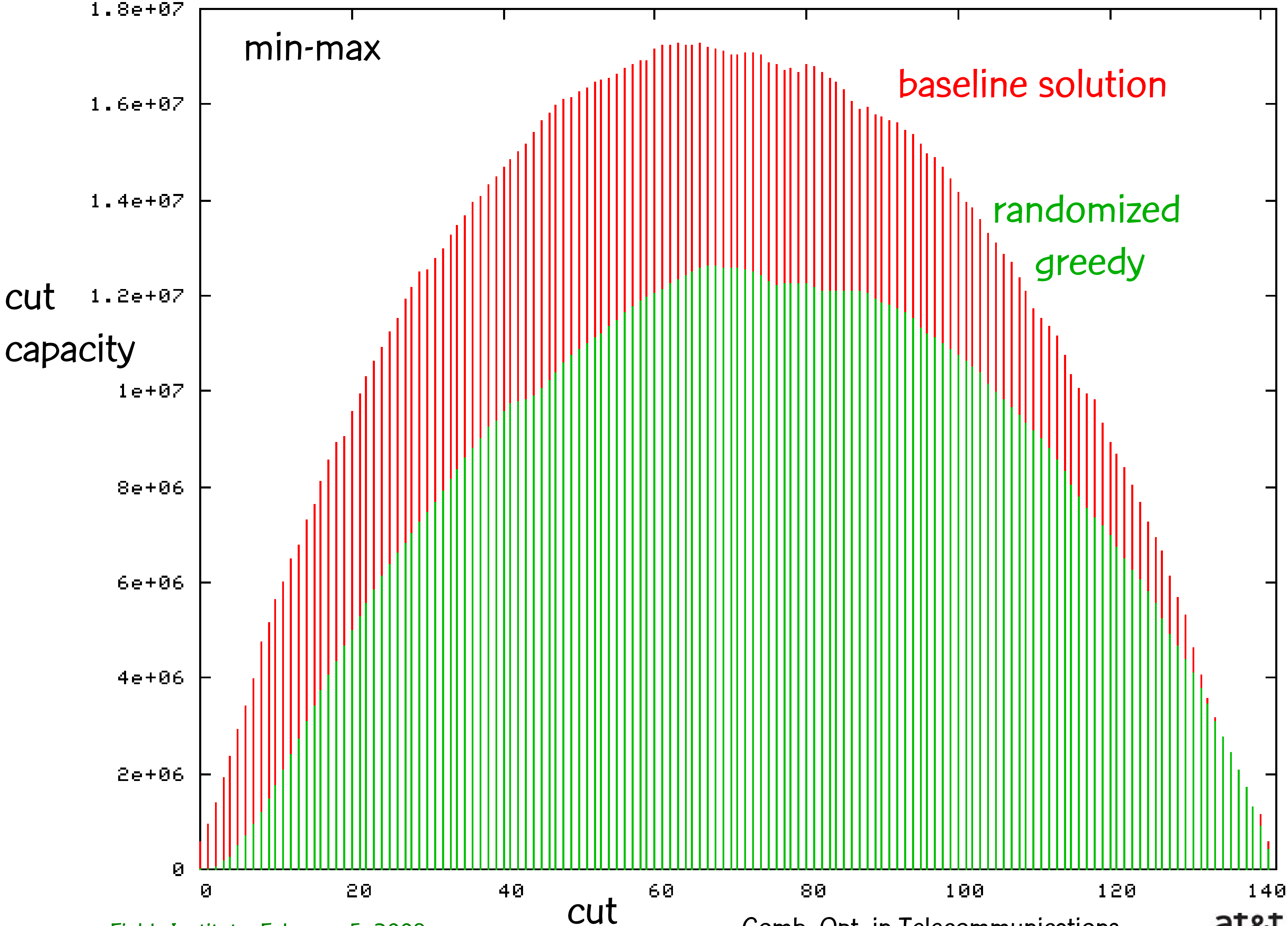
- Legacy AT&T phone network had 140 4ESS switches (nodes) and 9730 links: 100% edge density.
- 4ESS switches were to be "deloaded" and traffic moved to new IP network.
- One 4ESS switch is "deloaded" at each time period.

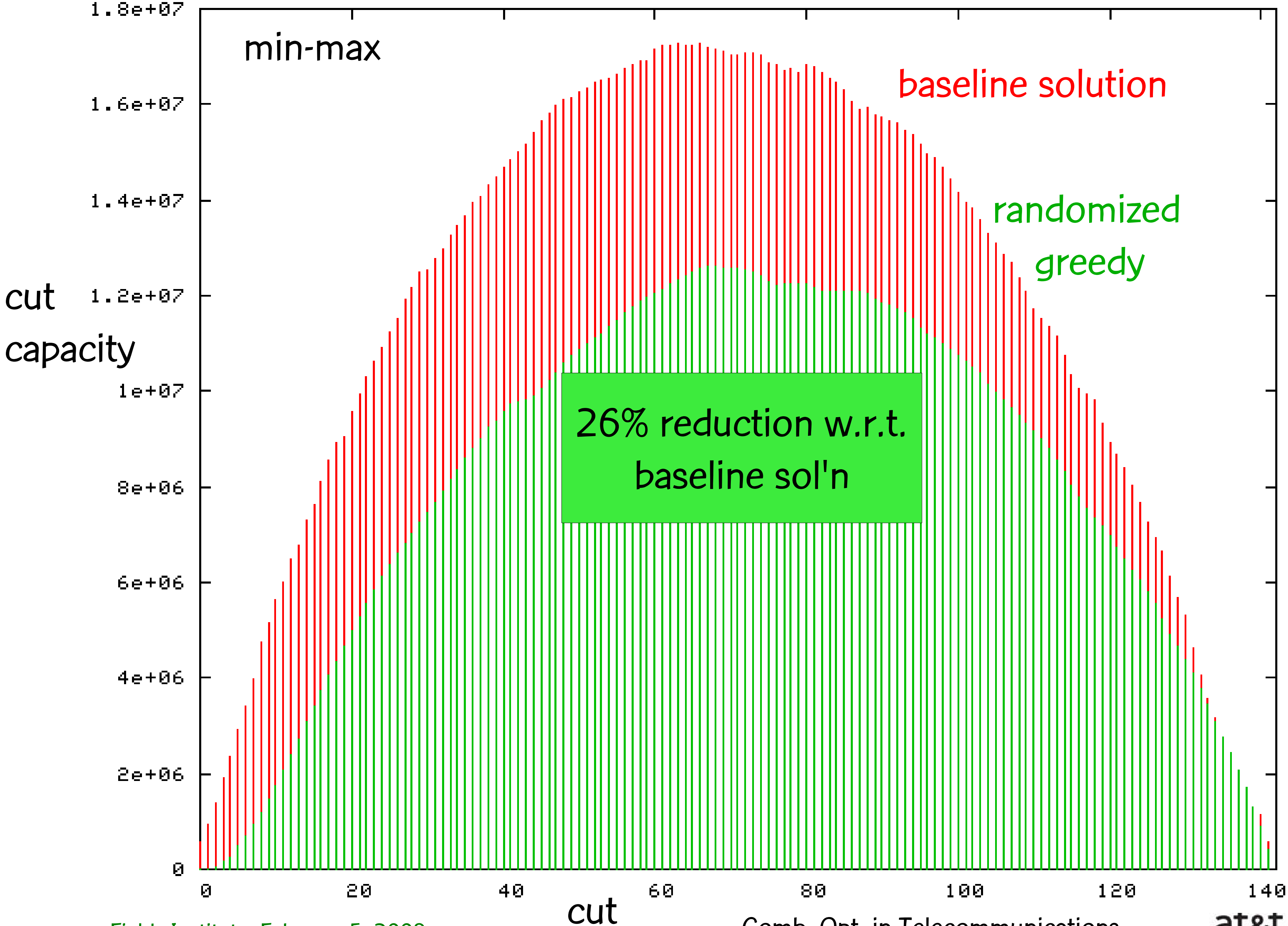
Scheduling 4ESS deloading process

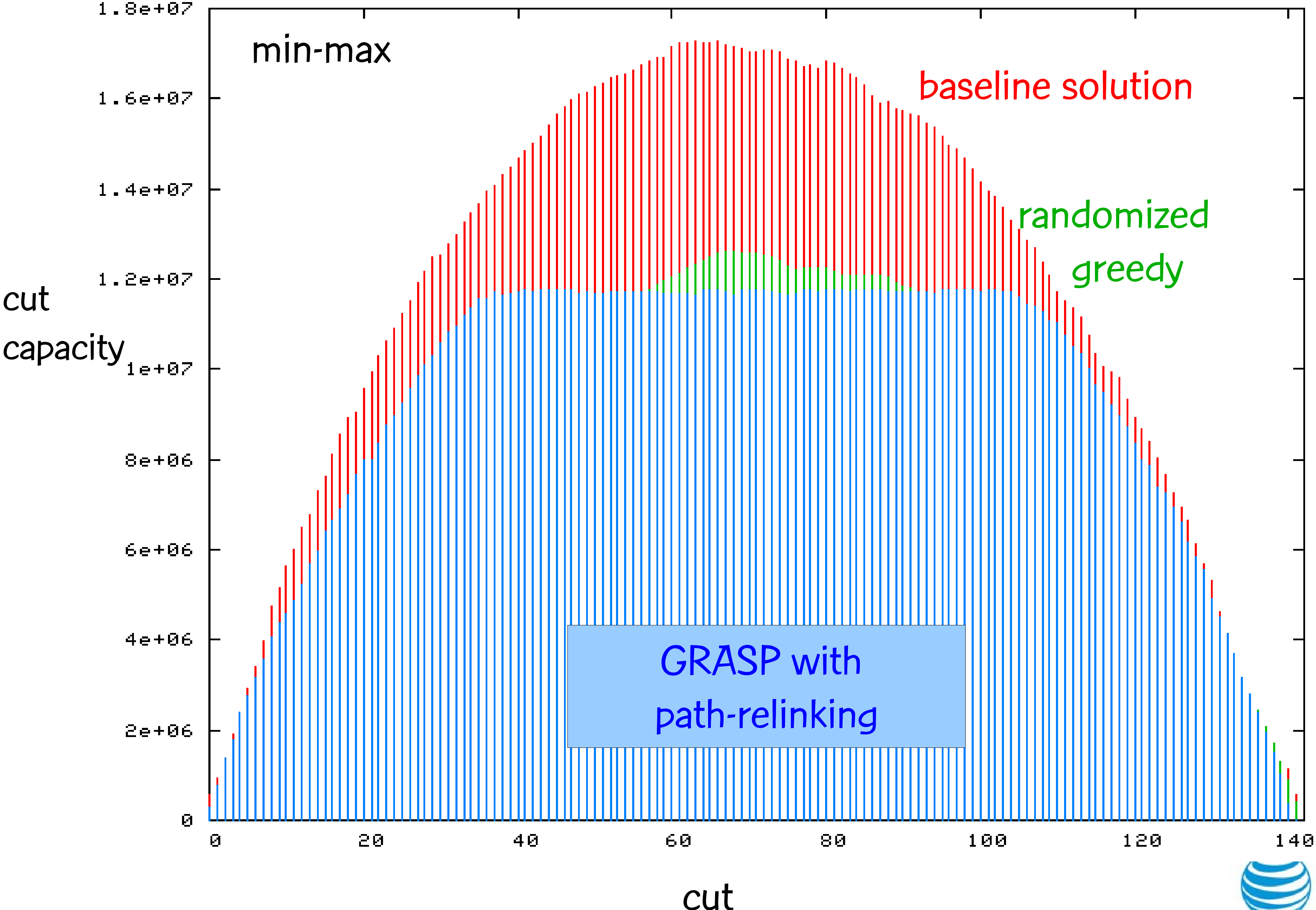
- With colleagues at AT&T, we built a web-based tool to simulate the deloading process.
- The tool estimates traffic before and after a switch is deloaded.
- Being optimizers, we asked:
  - What is the best ordering for the 4ESS deloading process?
  - We want to minimize the amount of capacity that will need to be built to accomplish the deloading.



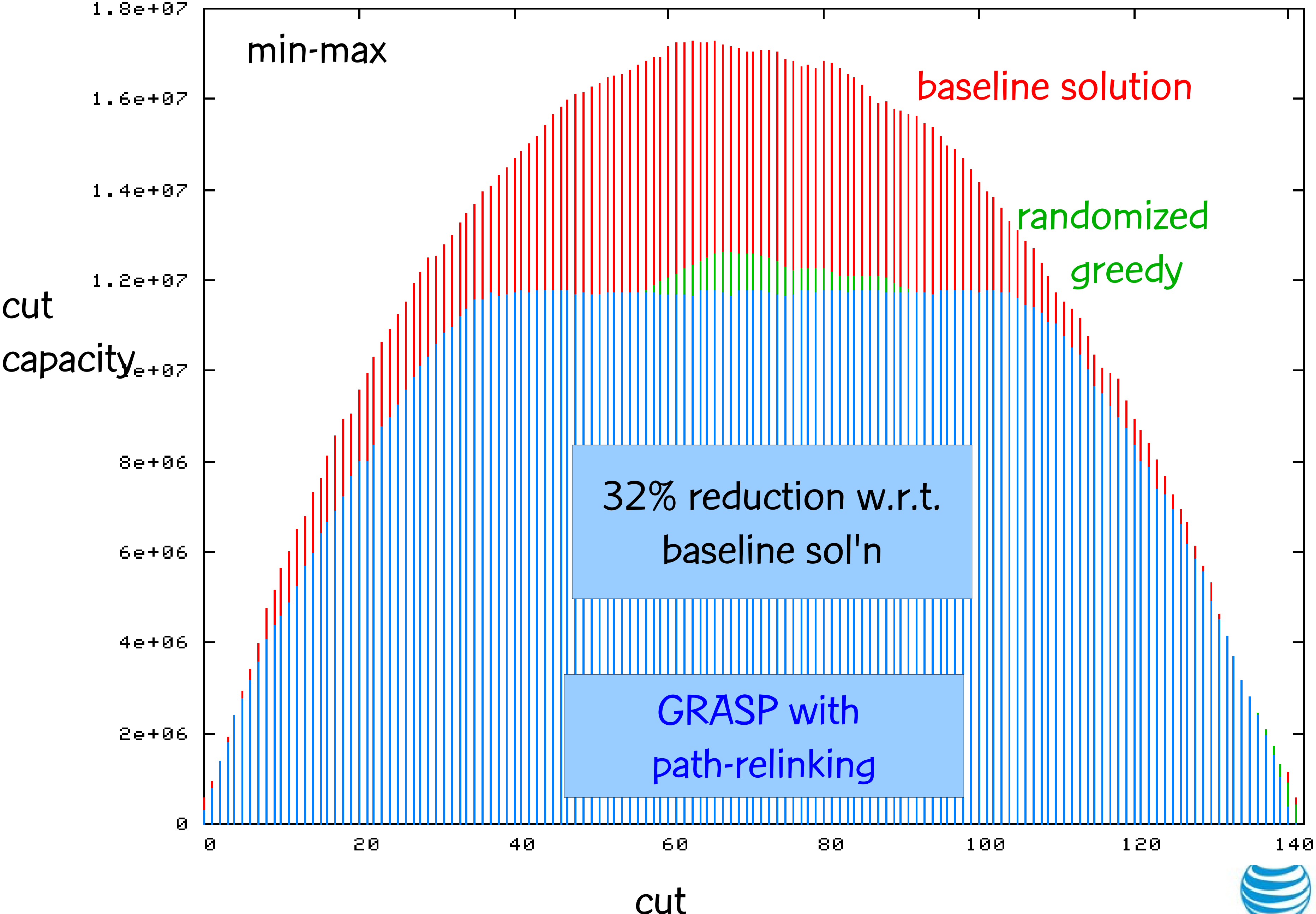
min-max
baseline solution
cut
capacity
1.8e+07
1.6e+07
1.4e+07
1.2e+07
1e+07
8e+06
6e+06
4e+06
2e+06
0
0
20
40
60
80
100
120
140
cut


min-max
baseline solution
randomized
greedy
cut
capacity
1.8e+07
1.6e+07
1.4e+07
1.2e+07
1e+07
8e+06
6e+06
4e+06
2e+06
0
0
20
40
60
80
100
120
140
cut


min-max
baseline solution
randomized
greedy
cut
capacity
26% reduction w.r.t.
baseline sol'n
cut
1.8e+07
1.6e+07
1.4e+07
1.2e+07
1e+07
8e+06
6e+06
4e+06
2e+06
0
0
20
40
60
80
100
120
140



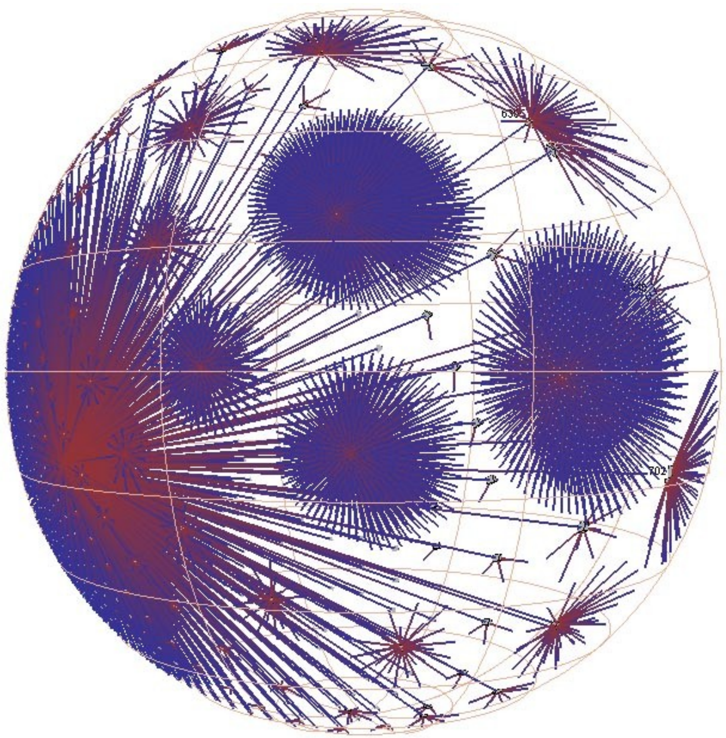
min-max
baseline solution
randomized
greedy
GRASP with
path-relinking
cut
capacity
cut
1.8e+07
1.6e+07
1.4e+07
1.2e+07
1e+07
8e+06
6e+06
4e+06
2e+06
0
0
20
40
60
80
100
120
140


min-max
baseline solution
randomized greedy
32% reduction w.r.t. baseline sol'n
GRASP with path-relinking
cut capacity
cut
1.8e+07
1.6e+07
1.4e+07
1.2e+07
8e+06
6e+06
4e+06
2e+06
0
0
20
40
60
80
100
120
140

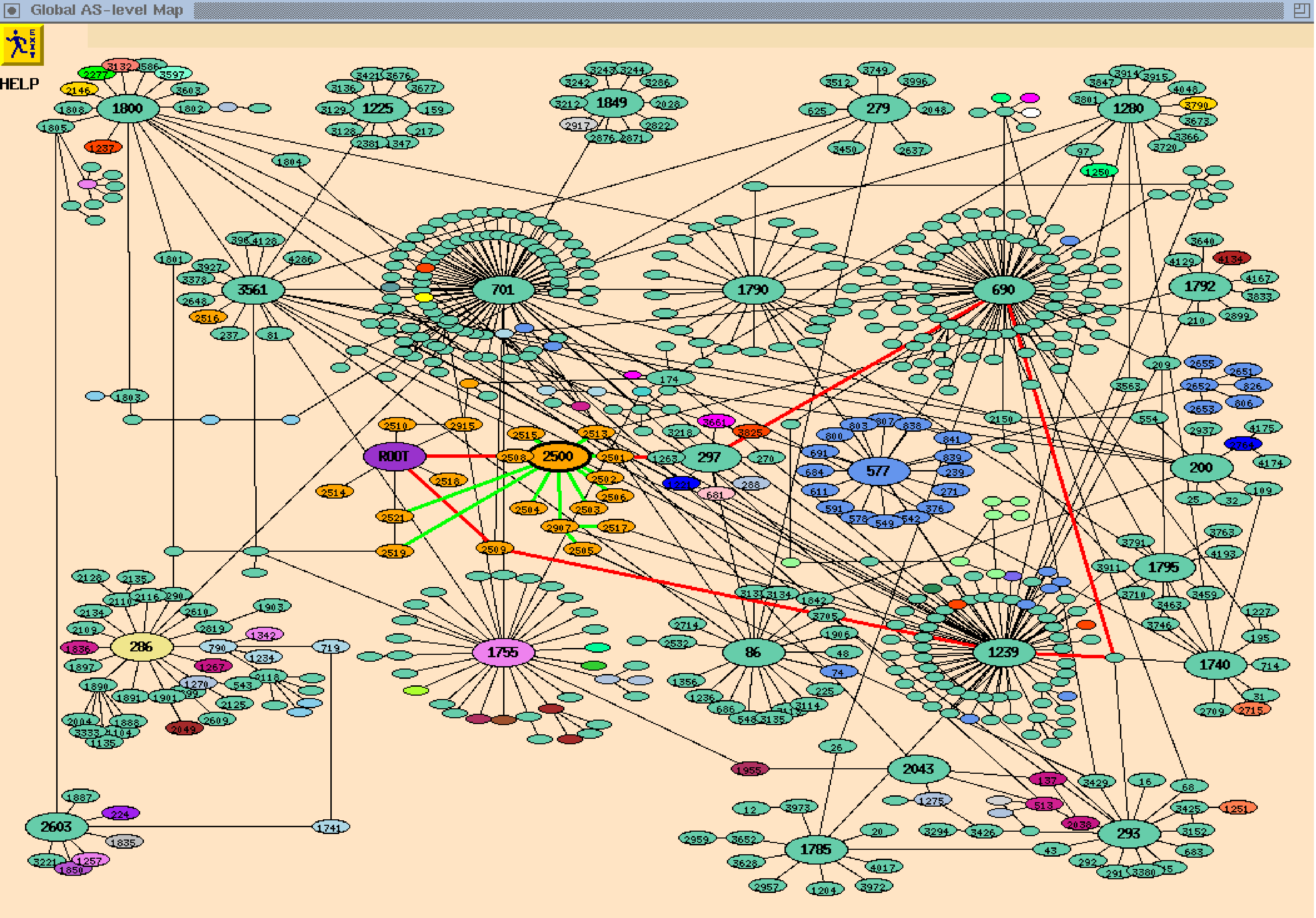
- D.V. Andrade and M.G.C.R., "GRASP with path-relinking for network migration scheduling," Proceedings of International Network Optimization Conference (INOC 2007), Spa, Belgium, 2007.

Routing in Internet Protocol (IP) Networks

The Internet



- The Internet is composed of many (inter-connected) autonomous systems (AS).
- An AS is a network controlled by a single entity, e.g. ISP, university, corporation, country, ...


Global AS-level Map
EXIT
HELP
1800
1225
1849
279
1280
3561
701
1790
690
1792
ROOT
2500
297
577
200
1795
286
1755
86
1239
1740
2043
2603
1785
293

Routing

- A packet is sent from a origination router S to a destination router T.
- S and T may be in
  - same AS:
  - different ASes:

Routing

- A packet is sent from a origination router S to a destination router T.
- S and T may be in
  - same AS: Interior Gateway Protocol (IGP) routing
  - different ASes:

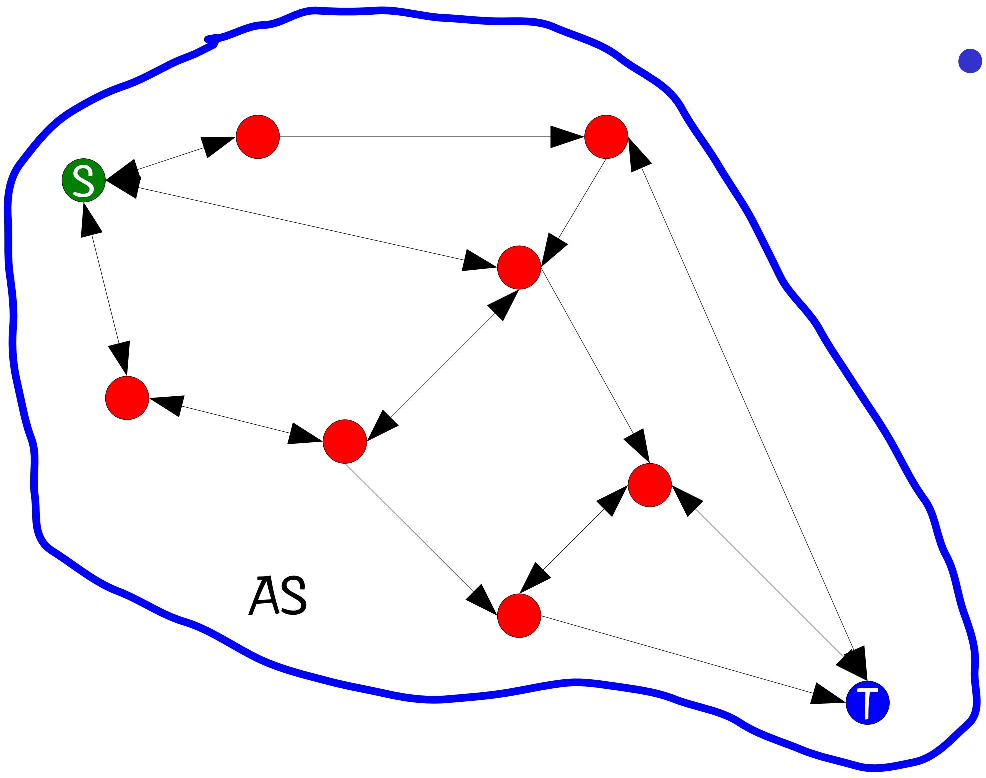
Routing

- A packet is sent from a origination router S to a destination router T.
- S and T may be in
  - same AS: Interior Gateway Protocol (IGP) routing
  - different ASes: Border Gateway Protocol (BGP) routing

Routing

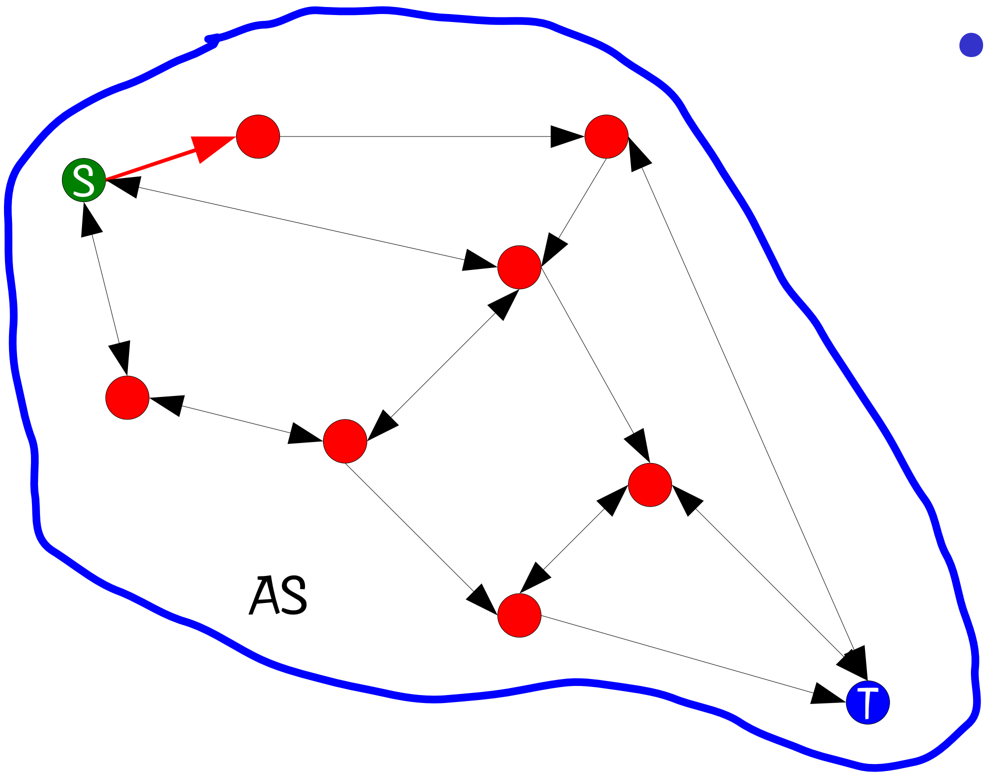
- A packet is sent from a origination router S to a destination router T.
- S and T may be in
  - same AS: Interior Gateway Protocol (IGP) routing
  - different ASes: Border Gateway Protocol (BGP) routing

IGP Routing



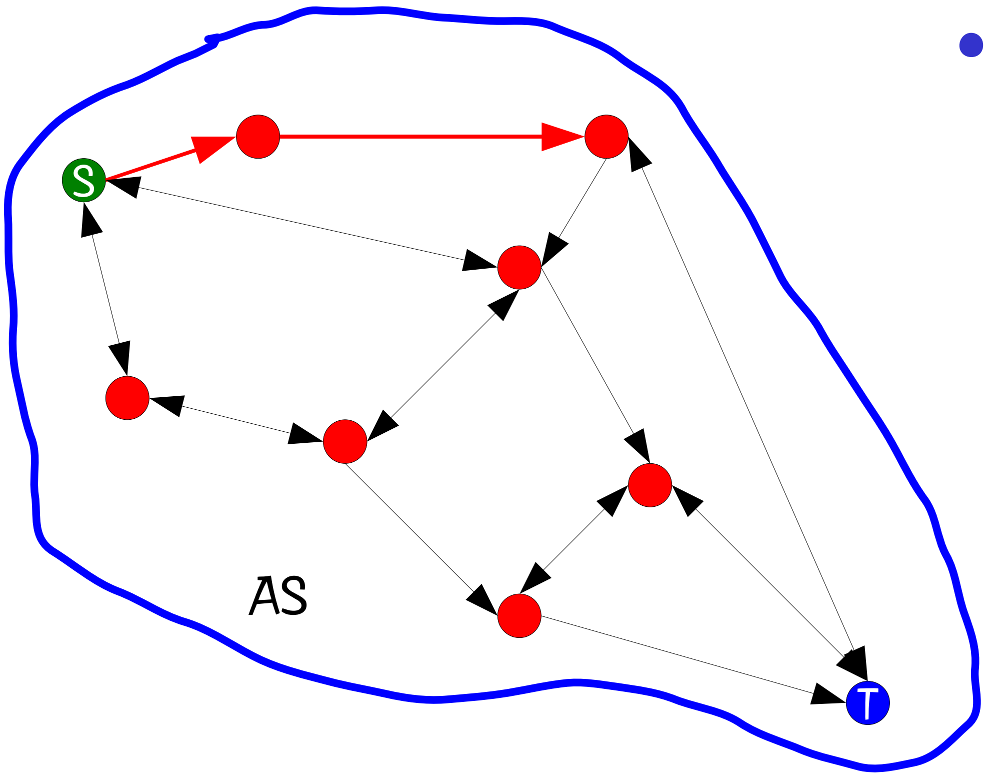
- IGP (interior gateway protocol) routing is concerned with routing within an AS.

IGP Routing



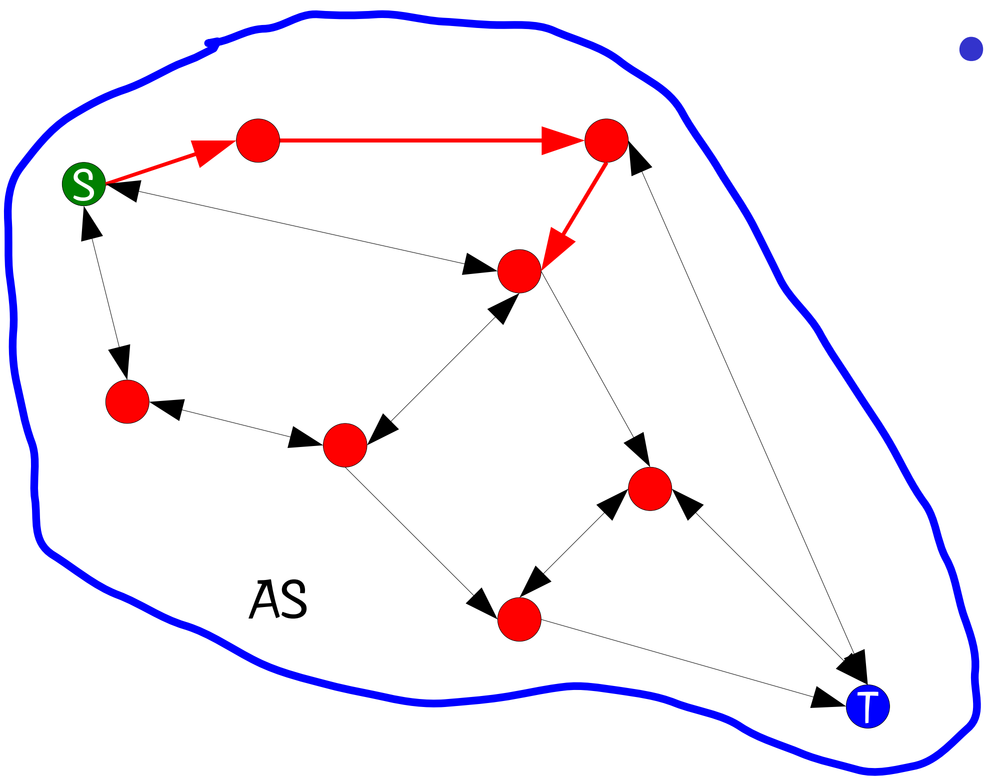
- IGP (interior gateway protocol) routing is concerned with routing within an AS.

IGP Routing



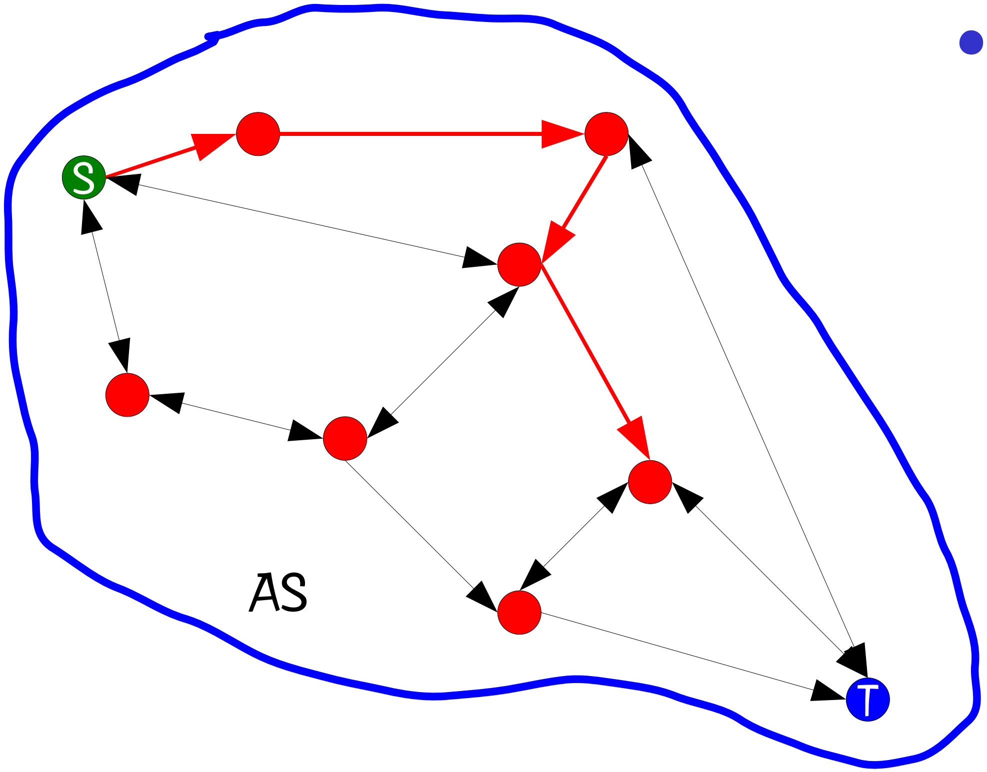
- IGP (interior gateway protocol) routing is concerned with routing within an AS.

IGP Routing



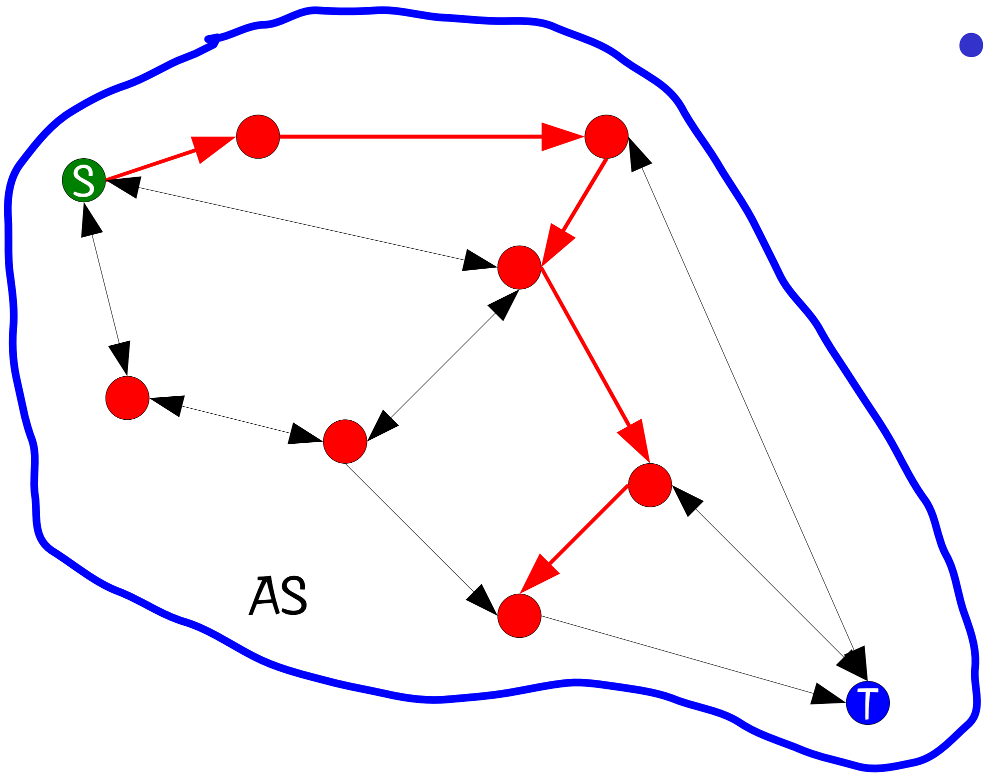
- IGP (interior gateway protocol) routing is concerned with routing within an AS.

IGP Routing



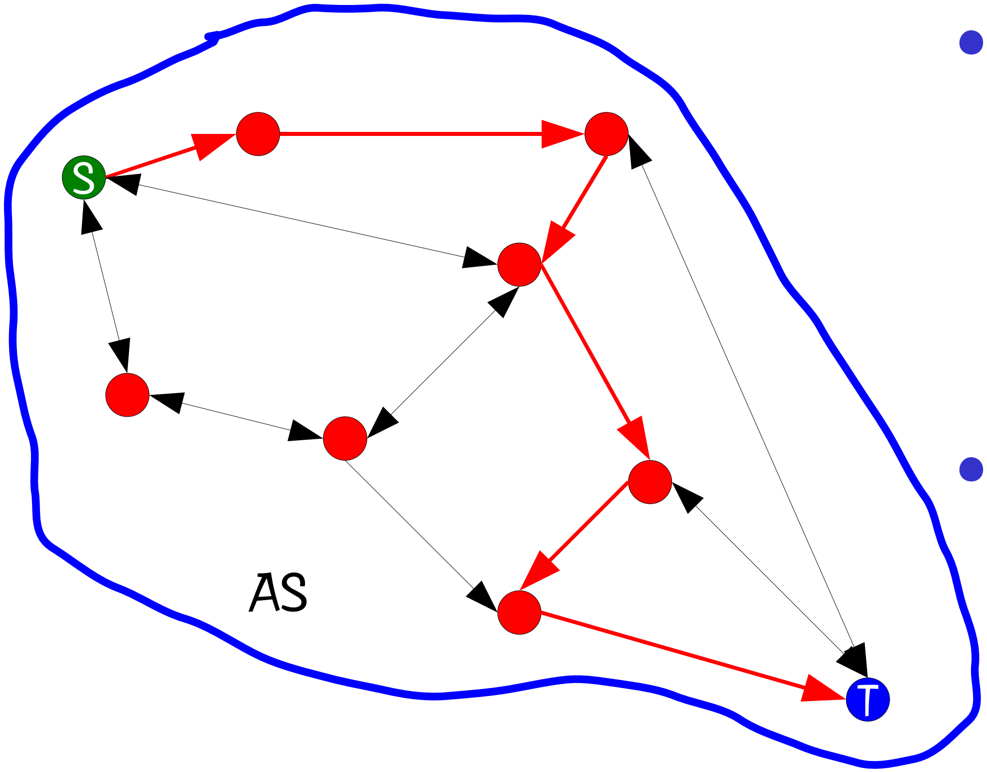
- IGP (interior gateway protocol) routing is concerned with routing within an AS.

IGP Routing



- IGP (interior gateway protocol) routing is concerned with routing within an AS.

IGP Routing



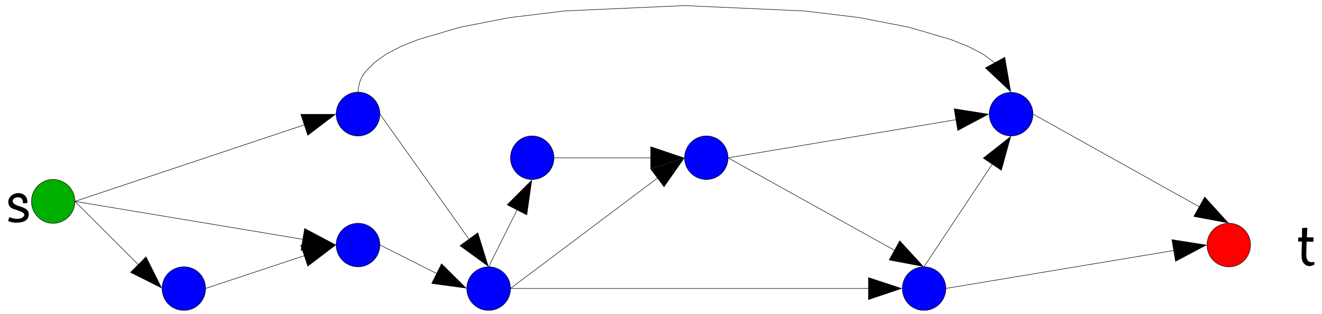
- IGP (interior gateway protocol) routing is concerned with routing within an AS.
- Routing decisions are made by AS operator.

OSPF routing

- Given a network $G = (N,A)$, where N is the set of routers and A is the set of links.

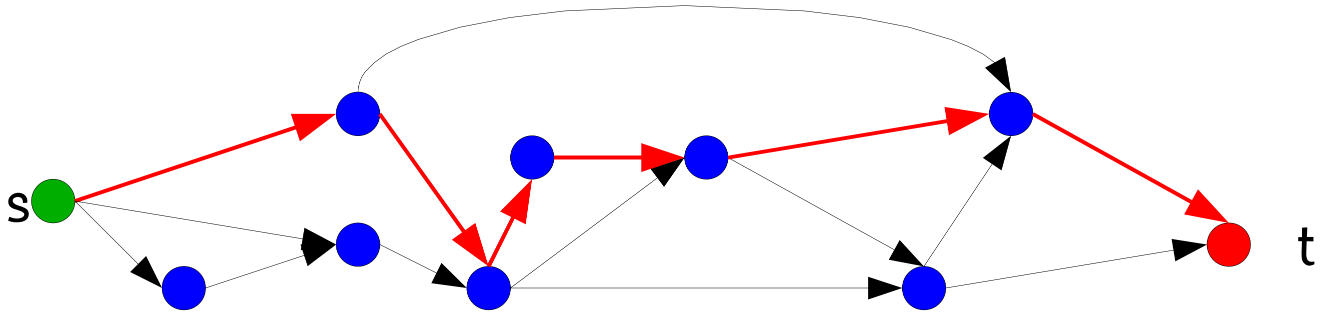
OSPF routing

- Given a network G = (N,*A*), where N is the set of routers and *A* is the set of links.
- The OSPF (open shortest path first) routing protocol assumes each link a has a weight w(a) assigned to it so that a packet from a source router s to a destination router t is routed on a shortest weight path from s to t.



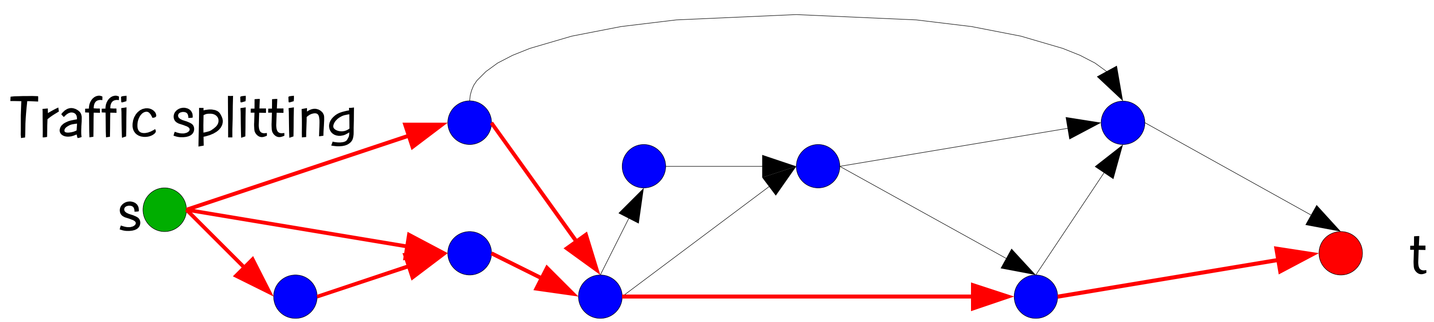
OSPF routing

- Given a network G = (N,*A*), where N is the set of routers and *A* is the set of links.
- The OSPF (open shortest path first) routing protocol assumes each link a has a weight w(a) assigned to it so that a packet from a source router s to a destination router t is routed on a shortest weight path from s to t.



OSPF routing

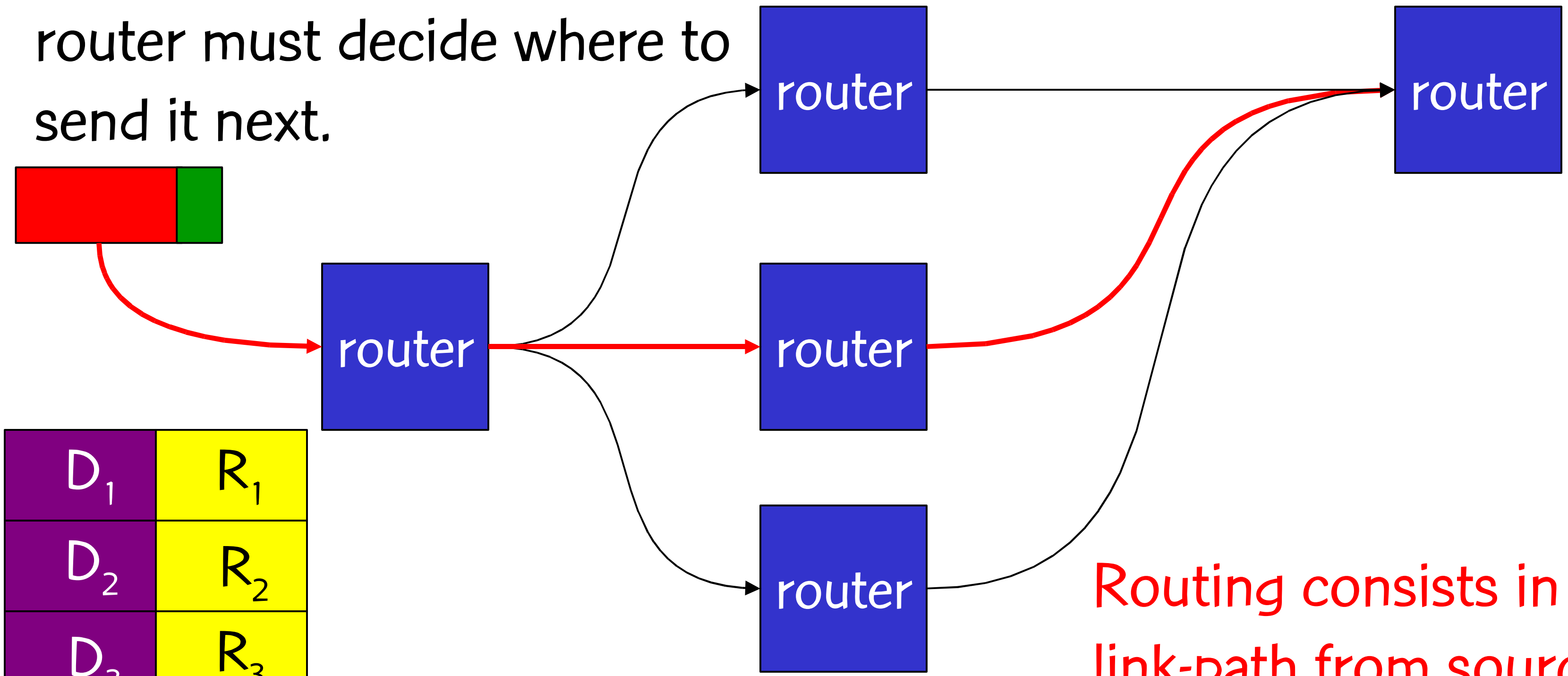
- Given a network G = (N,A), where N is the set of routers and A is the set of links.
- The OSPF (open shortest path first) routing protocol assumes each link a has a weight w(a) assigned to it so that a packet from a source router s to a destination router t is routed on a shortest weight path from s to t.



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 routing

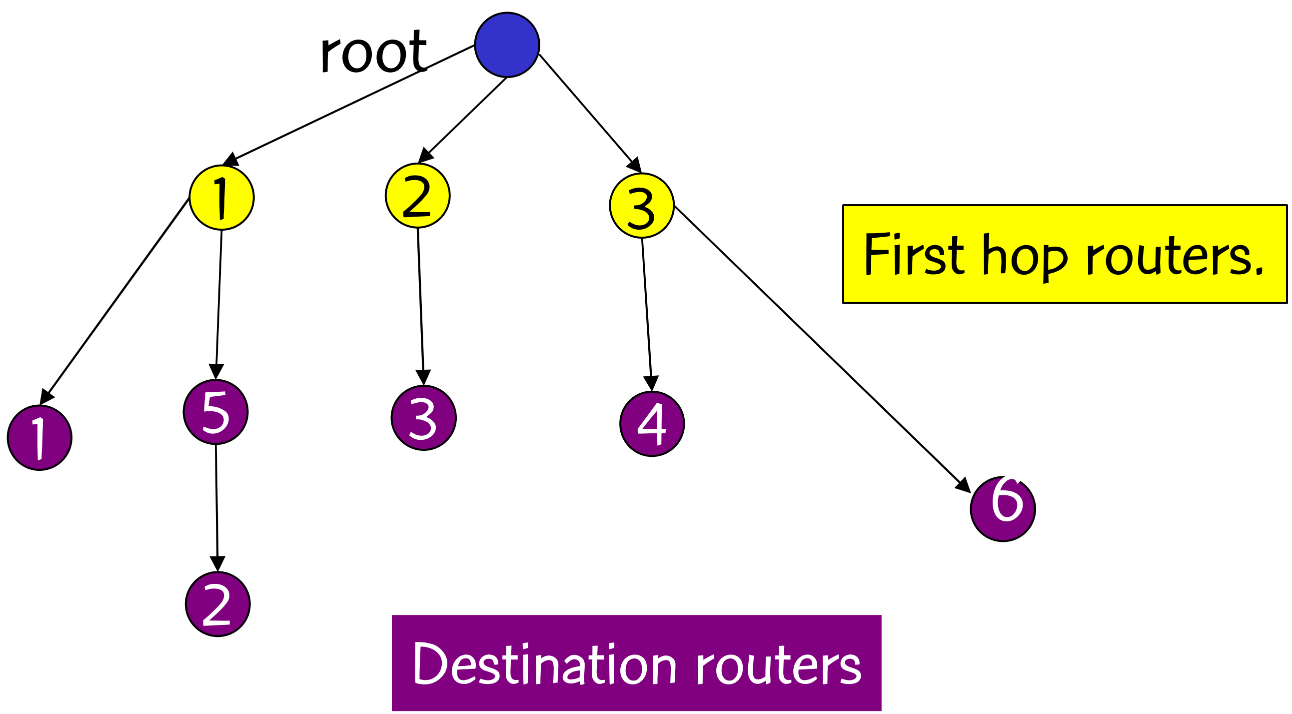
- Assign an integer weight $\in [1, w_{max}]$ to each link in AS. In general, $w_{max} = 65535 = 2^{16} - 1$.
- Each router computes tree of shortest weight paths to all other routers in the AS, with itself as the root, using Dijkstra's algorithm.

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

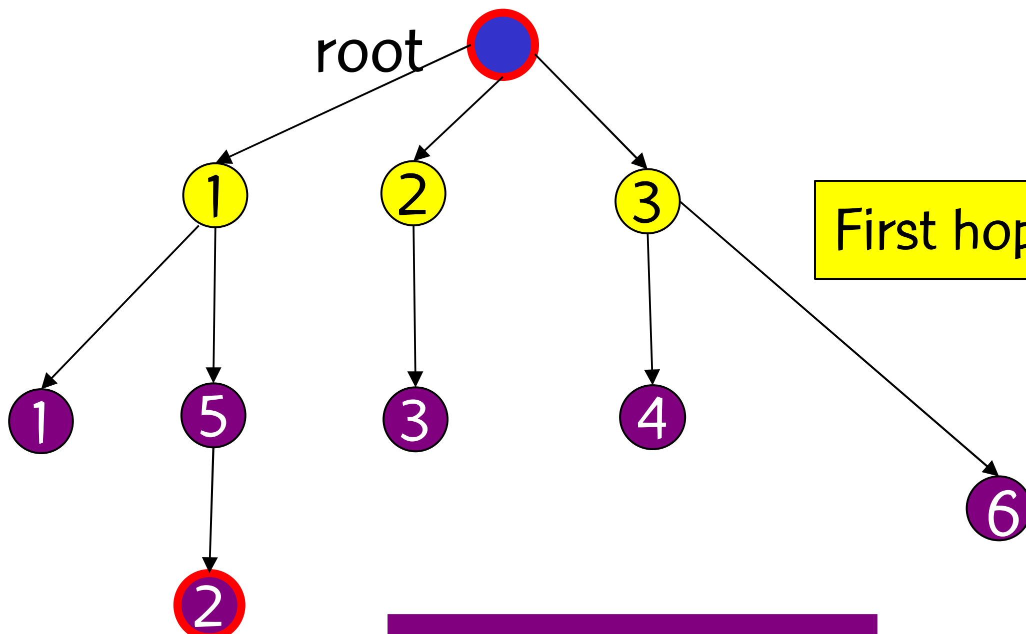


OSPF routing

Routing table

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.



First hop routers.

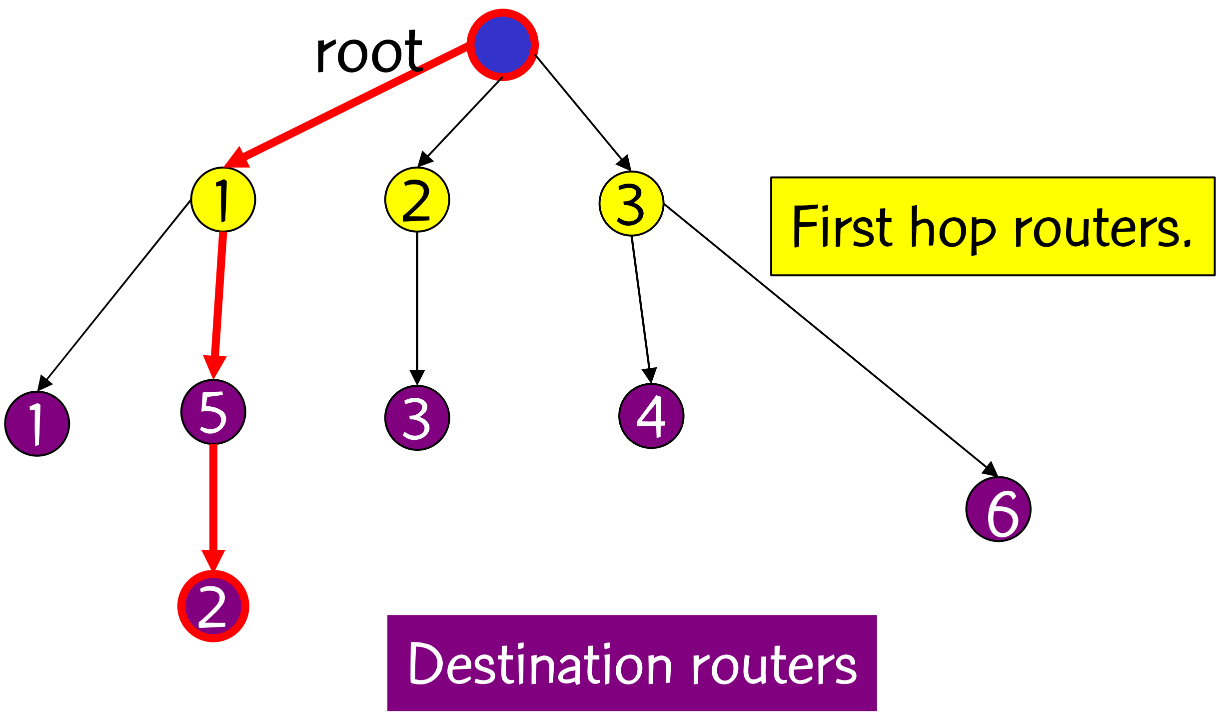
Destination routers

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

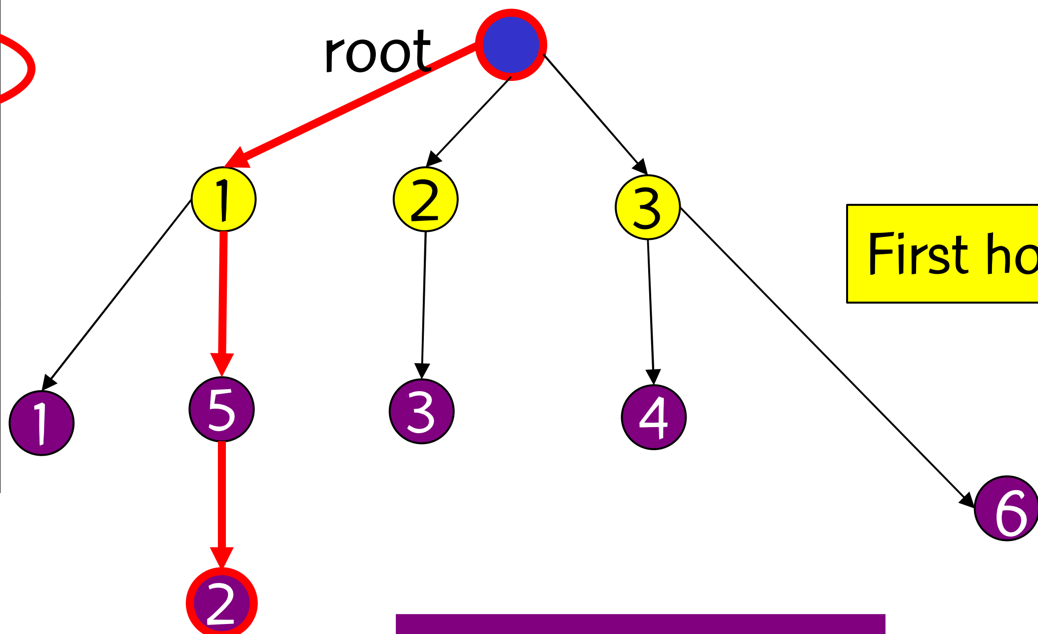


OSPF routing

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



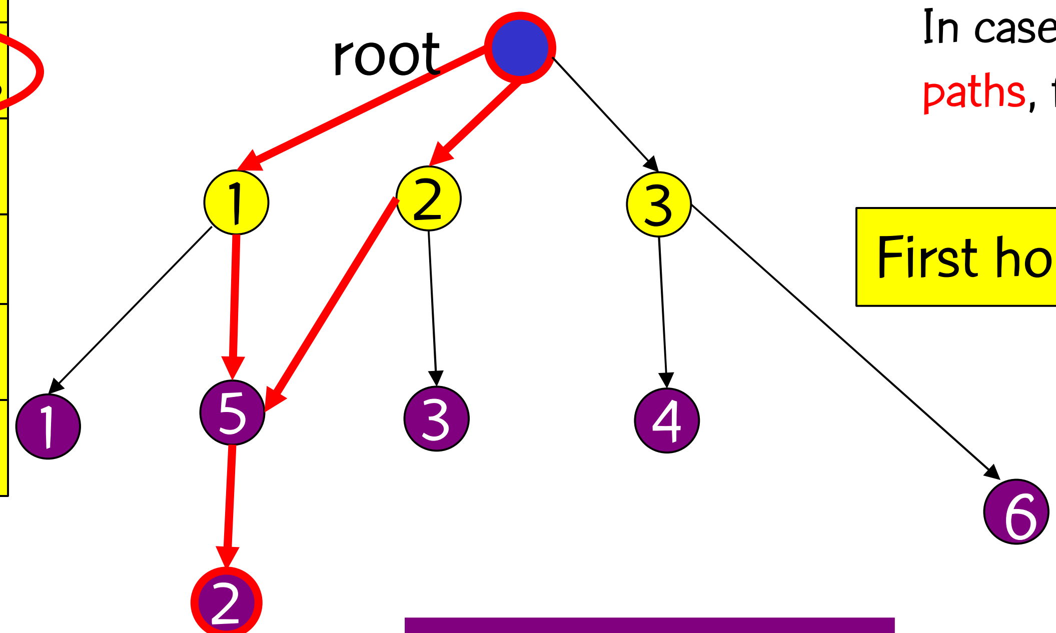
First hop routers.

Destination routers

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.

First hop routers.

Destination routers

OSPF routing

- By setting OSPF weights appropriately, one can do traffic engineering, i.e. route traffic so as to optimize some objective (e.g. minimize congestion, maximize throughput, etc.).
- Some recent papers on this topic:
  - Fortz & Thorup (2000, 2004)
  - Ramakrishnan & Rodrigues (2001)
  - Sridharan, Guérin, & Diot (2002)
  - Fortz, Rexford, & Thorup (2002)
  - Ericsson, R., & Pardalos (2002)
  - Buriol, R., Ribeiro, & Thorup (2002, 2005)

OSPF routing

- By setting OSPF weights appropriately, one can do traffic engineering, i.e. route traffic so as to optimize some objective (e.g. minimize congestion, maximize throughput, etc.).
- Some recent papers on this topic:
  - Fortz & Thorup (2000, 2004)
  - Ramakrishnan & Rodrigues (2001)
  - Sridharan, Guérin, & Diot (2002)
  - Fortz, Rexford, & Thorup (2002)
  - Ericsson, R., & Pardalos (2002)
  - Buriol, R., Ribeiro, & Thorup (2002, 2005)

OSPF routing

- By setting OSPF weights appropriately, one can do traffic engineering, i.e. route traffic so as to optimize some objective (e.g. minimize congestion, maximize throughput, etc.).
- Some recent papers on this topic:
  - Fortz & Thorup (2000, 2004)
  - Ramakrishnan & Rodrigues (2001)
  - Sridharan, Guérin, & Diot (2002)
  - Fortz, Rexford, & Thorup (2002)
  - Ericsson, R., & Pardalos (2002)
  - Buriol, R., Ribeiro, & Thorup (2002, 2005)

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 evolutionary algorithms to find good OSPF weights.
  - Genetic algorithm
  - 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:

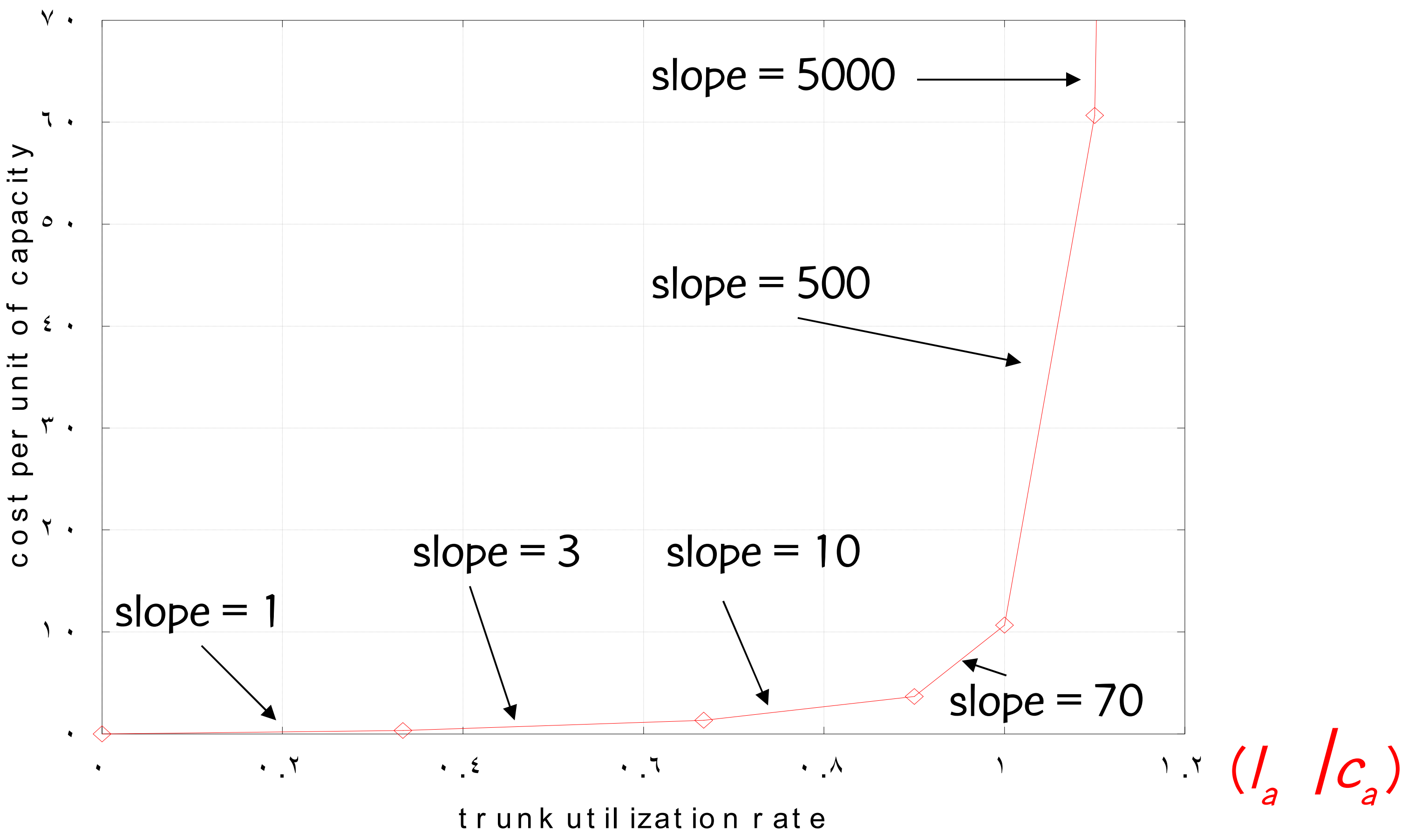
$$\Phi = \Phi_1(l_1) + \Phi_2(l_2) + \ldots + \Phi_{|A|}(l_{|A|})$$

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

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

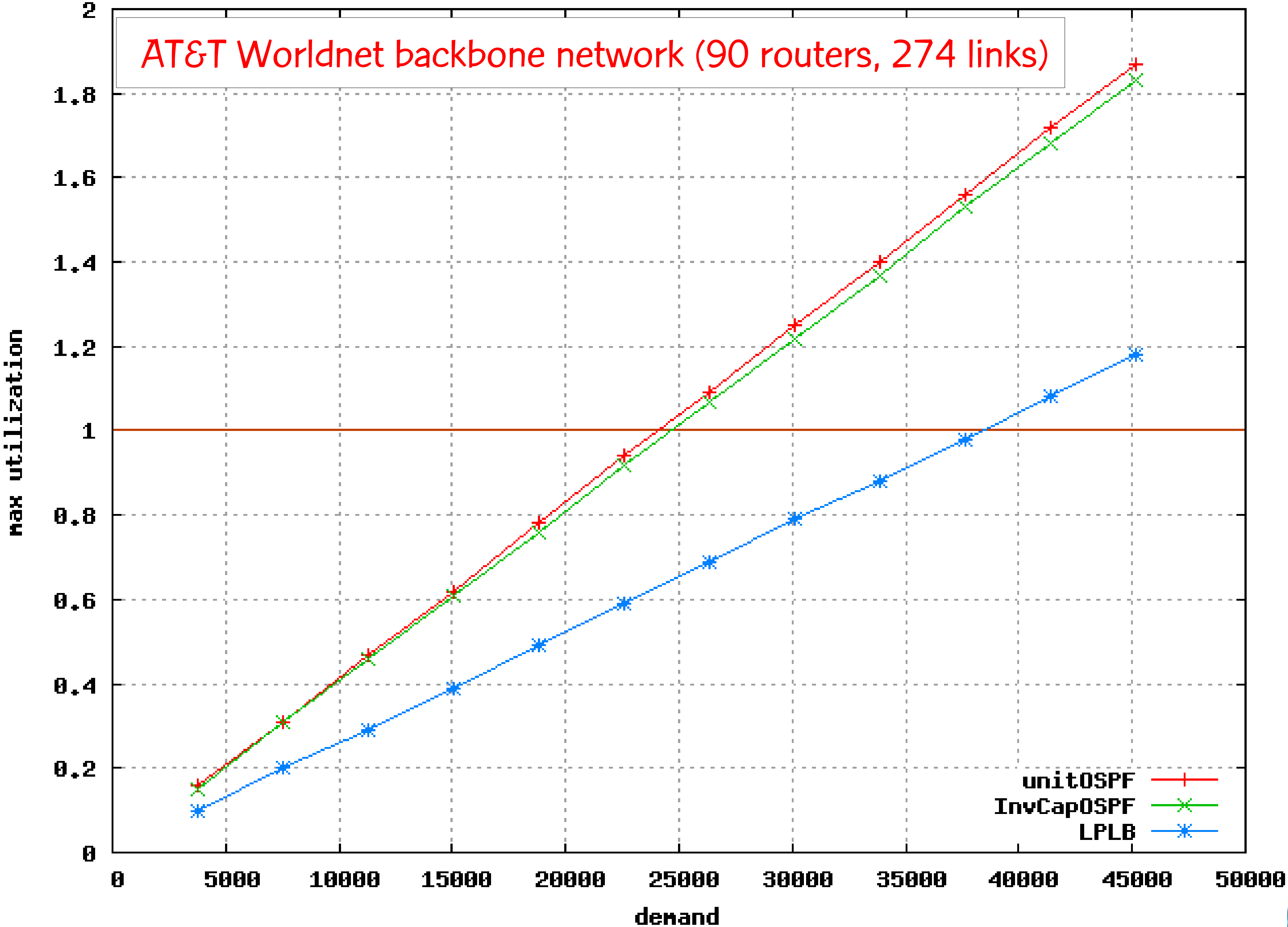
$\Phi_a(0) = 0$, for all $a \in A$.

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.



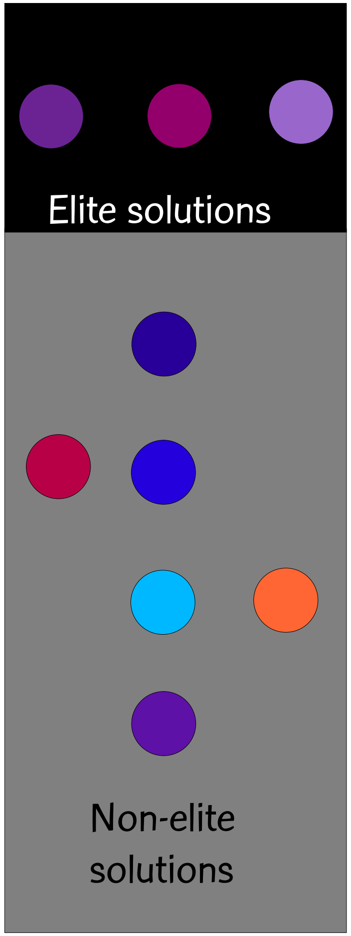
AT&T Worldnet backbone network (90 routers, 274 links)
max utilization
demand
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
0
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000
unitOSPF
InvCapOSPF
LPLB

Genetic and hybrid genetic algorithms for OSPF weight setting problem

- Genetic
  - M. Ericsson, M.G.C.R., & P.M. Pardalos, "A genetic algorithm for the weight setting problem in OSPF routing," J. of Combinatorial Optimization, vol. 6, pp. 299-333, 2002.
- Hybrid genetic
  - L.S. Buriol, M.G.C.R., C.C. Ribeiro, & M. Thorup, "A hybrid genetic algorithm for the weight setting problem in OSPF/IS-IS routing," Networks, vol. 46, pp. 36-56, 2005.

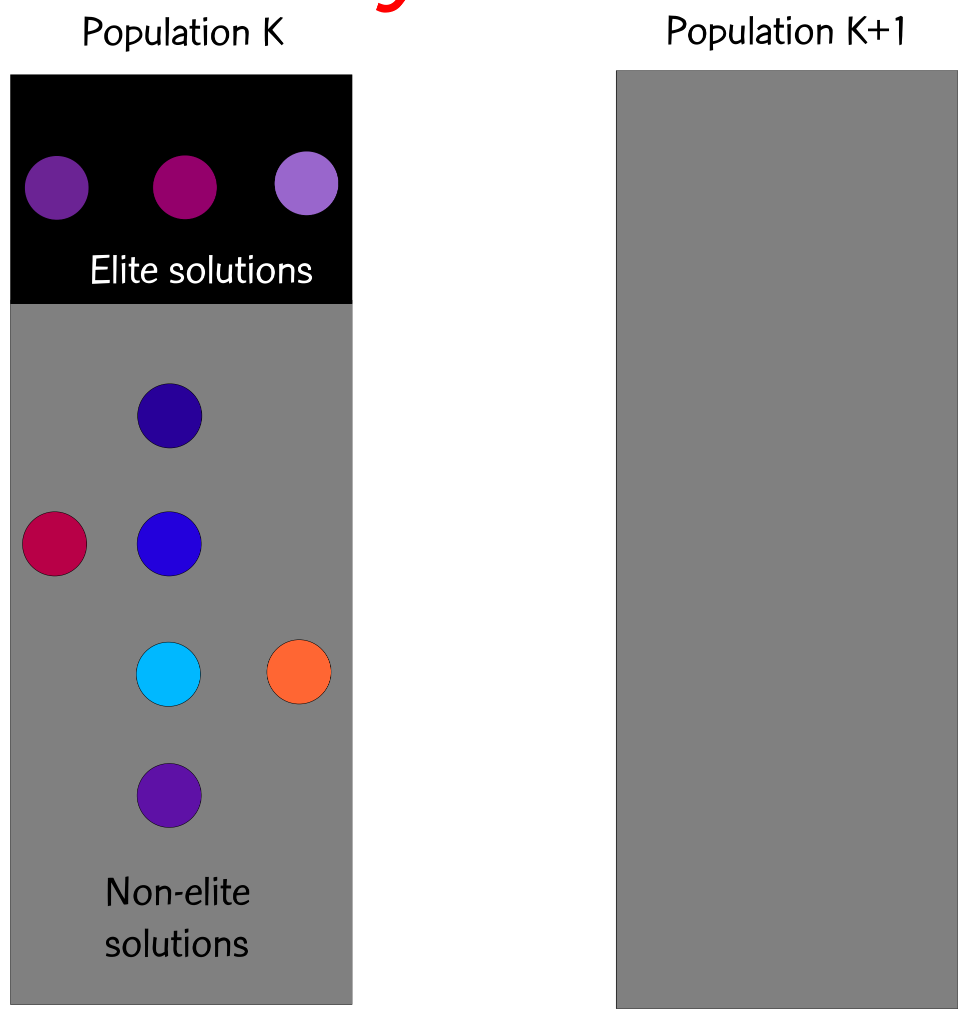
GAs and random keys

- Introduced by Bean (1994) for sequencing problems.
- At the K-th generation, compute the cost of each solution and partition the solutions into two sets: elite solutions, non-elite solutions. Elite set should be smaller of the two sets and contain best solutions.



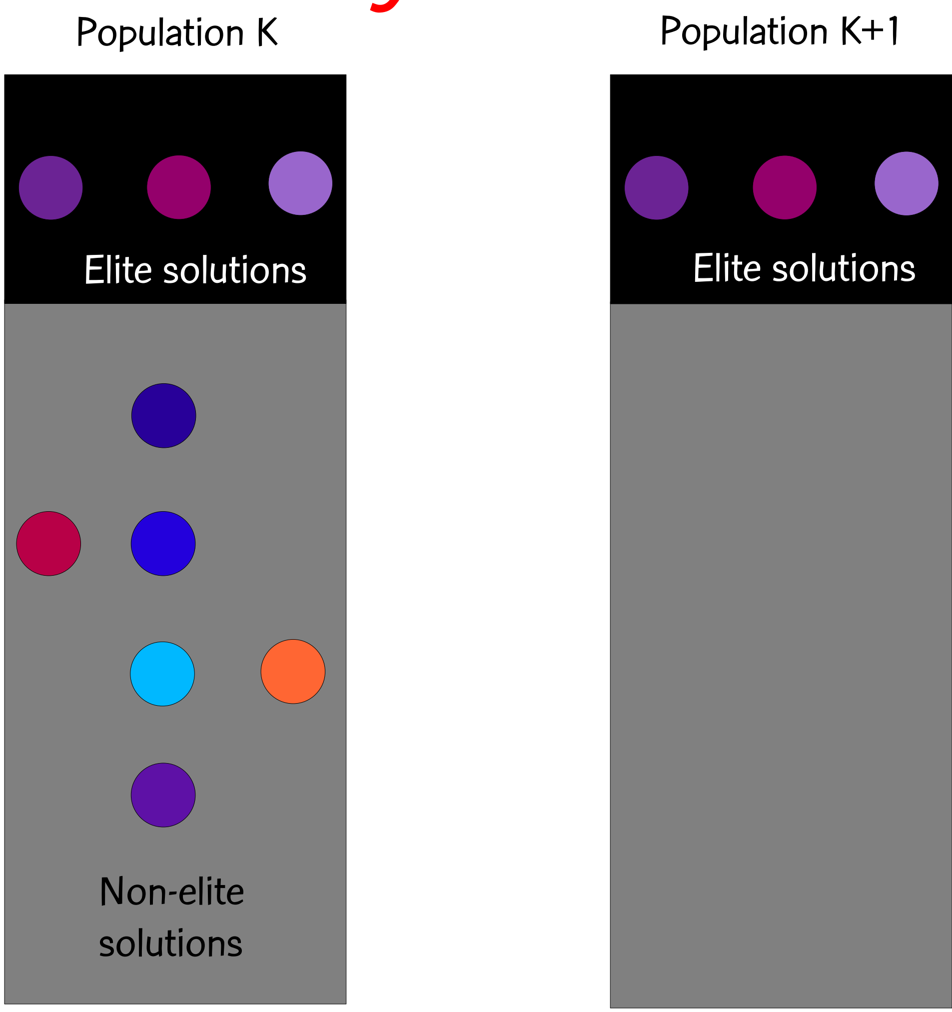
GAs and random keys

- Introduced by Bean (1994) for sequencing problems.
- Evolutionary dynamics



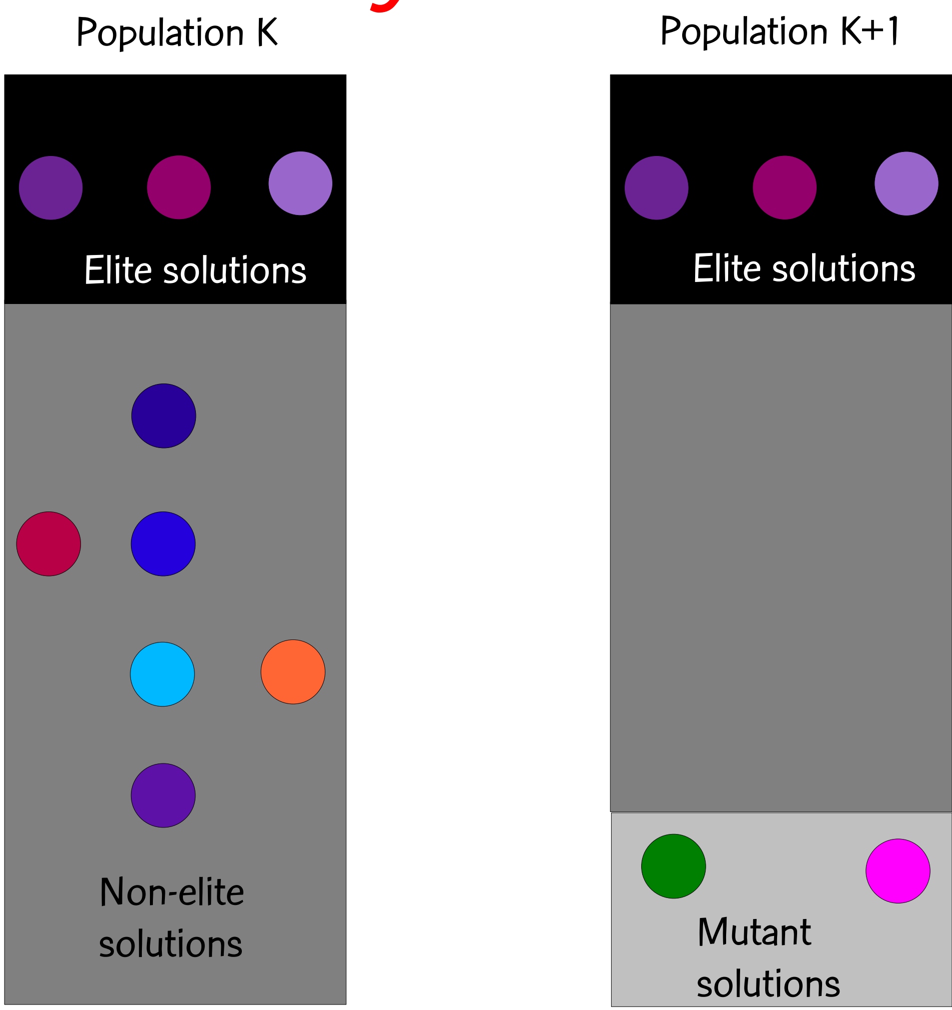
GAs and random keys

- Introduced by Bean (1994) for sequencing problems.
- Evolutionary dynamics
  - Copy elite solutions from population K to population K+1



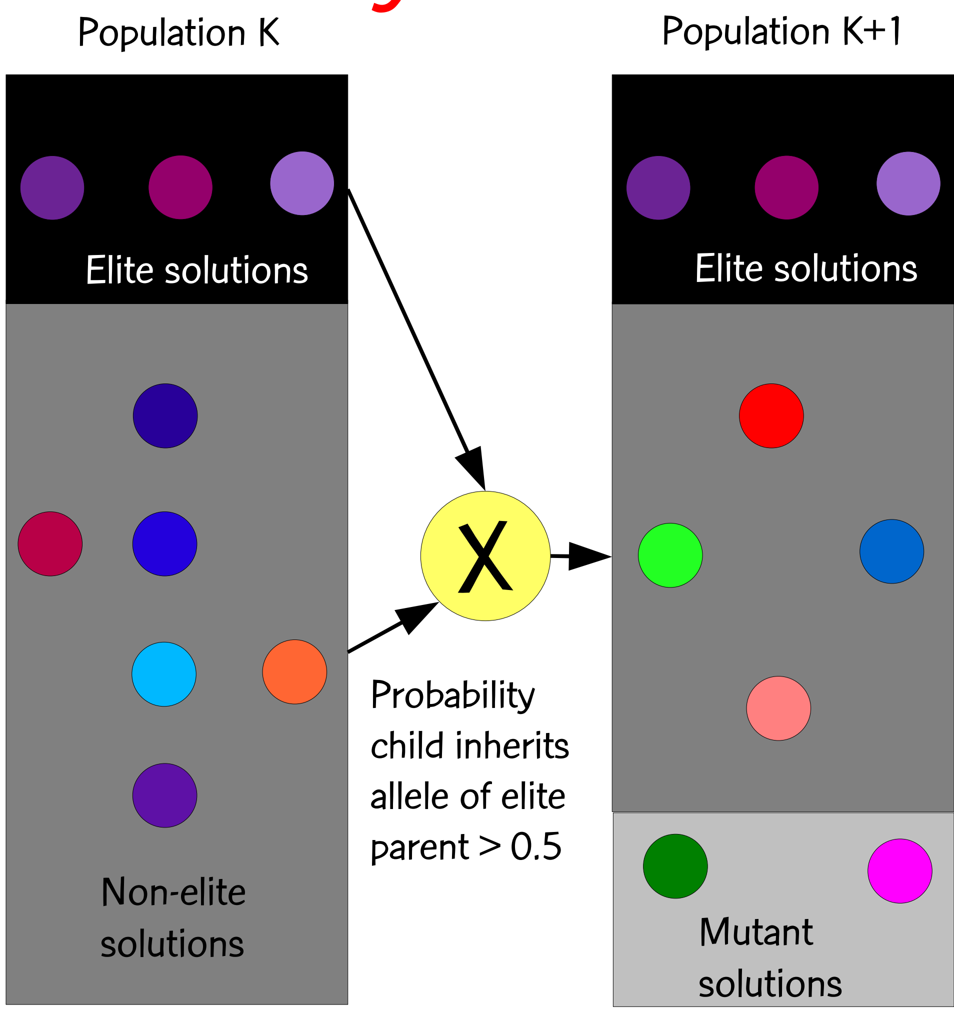
GAs and random keys

- Introduced by Bean (1994) for sequencing problems.
- Evolutionary dynamics
  - Copy elite solutions from population K to population K+1
  - Add R random solutions (mutants) to population K+1



GAs and random keys

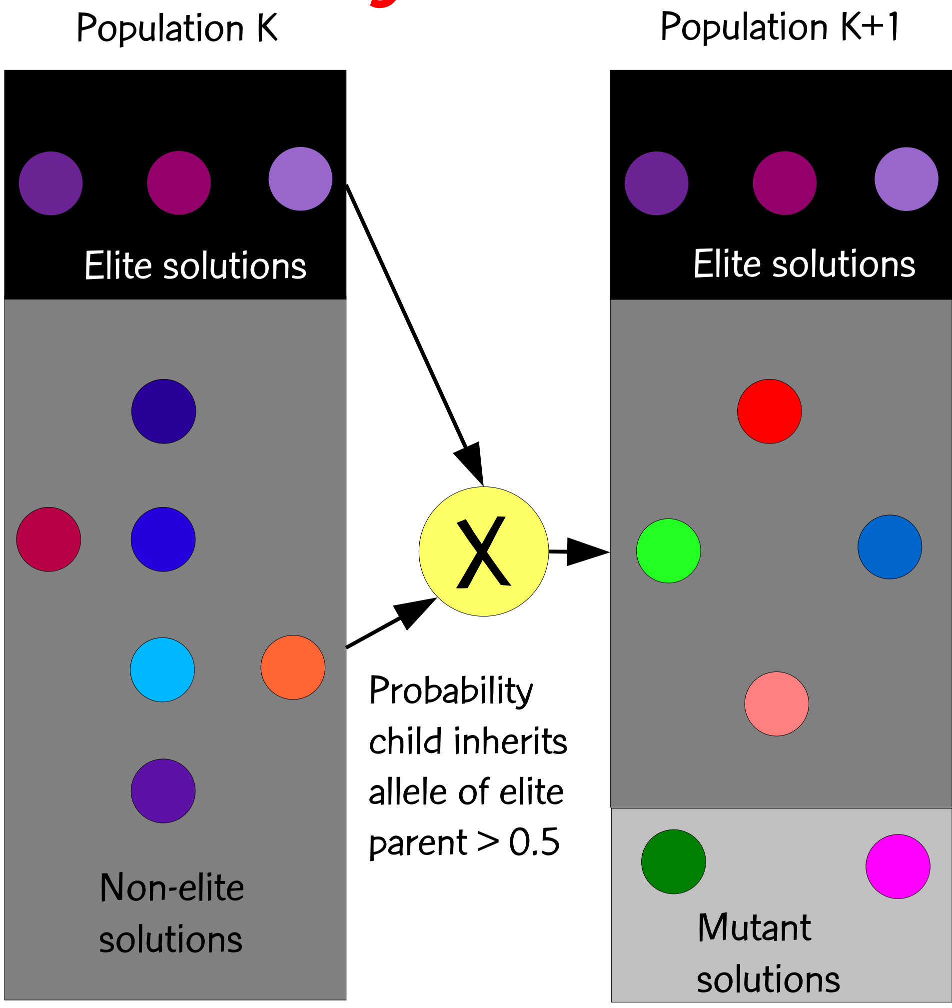
- Introduced by Bean (1994) for sequencing problems.
- Evolutionary dynamics
  - Copy elite solutions from population K to population K+1
  - Add R random solutions (mutants) to population K+1
  - While K+1-th population < P
    - Mate elite solution with non elite to produce child in population K+1. Mates are chosen at random.



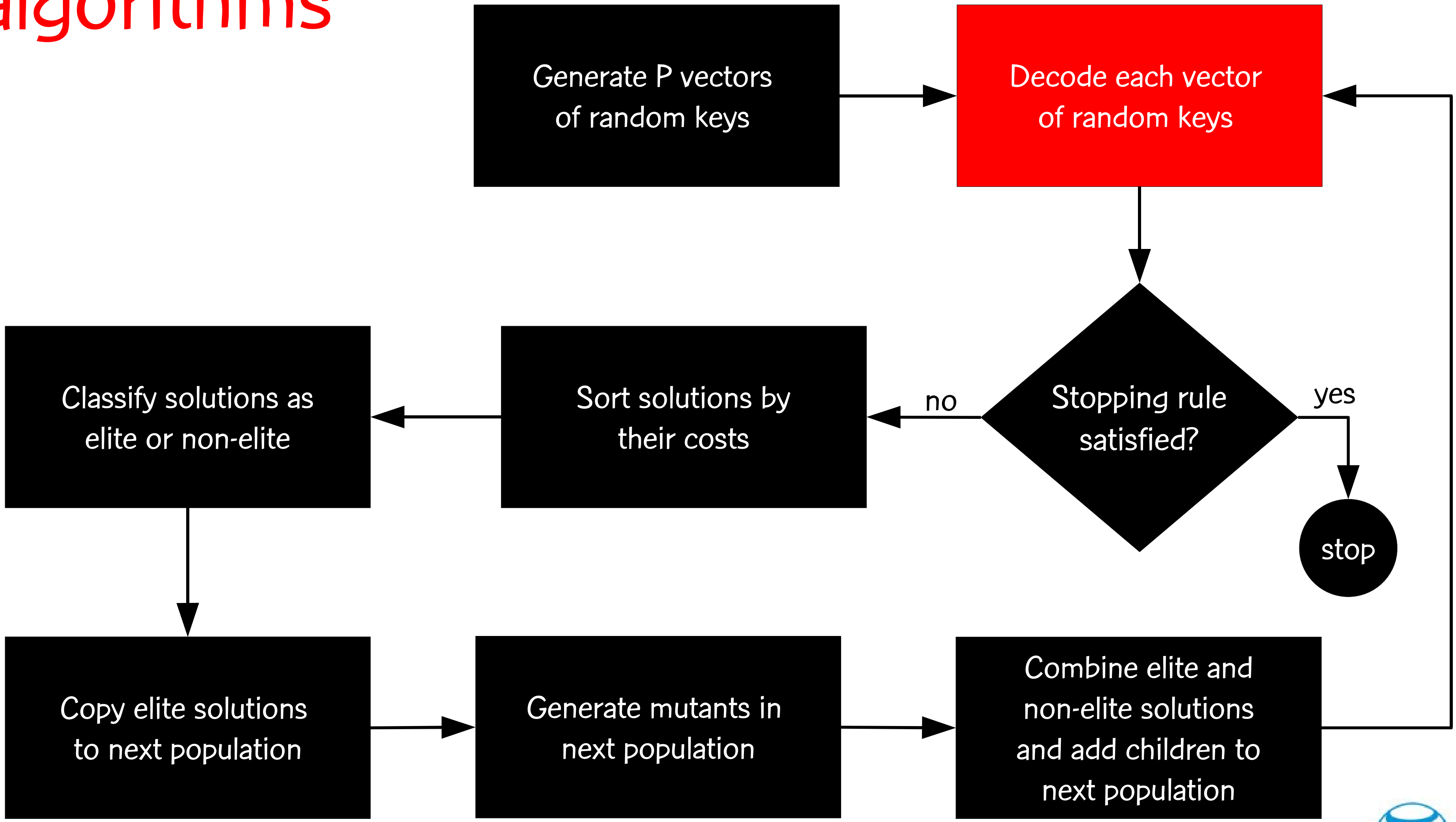
at&t Your world. Delivered.

GAs and random keys

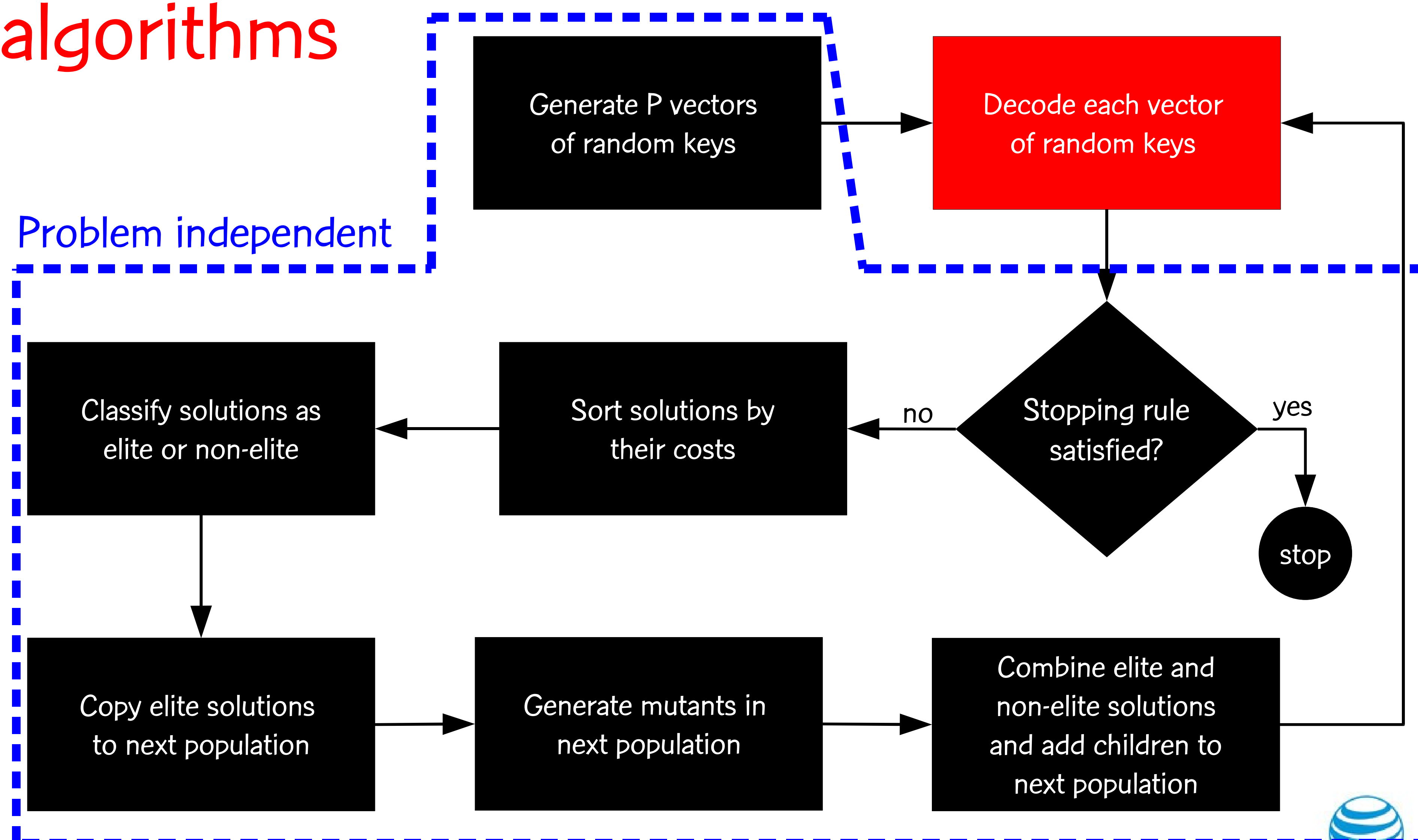
- Introduced by Bean (1994) for sequencing problems.
- In practice, good choices are:
  - Elite solutions are top 10 to 20% of population
  - Mutants are 5 to15% of population
  - Probability child inherits allele of elite parent is 60 to 80%



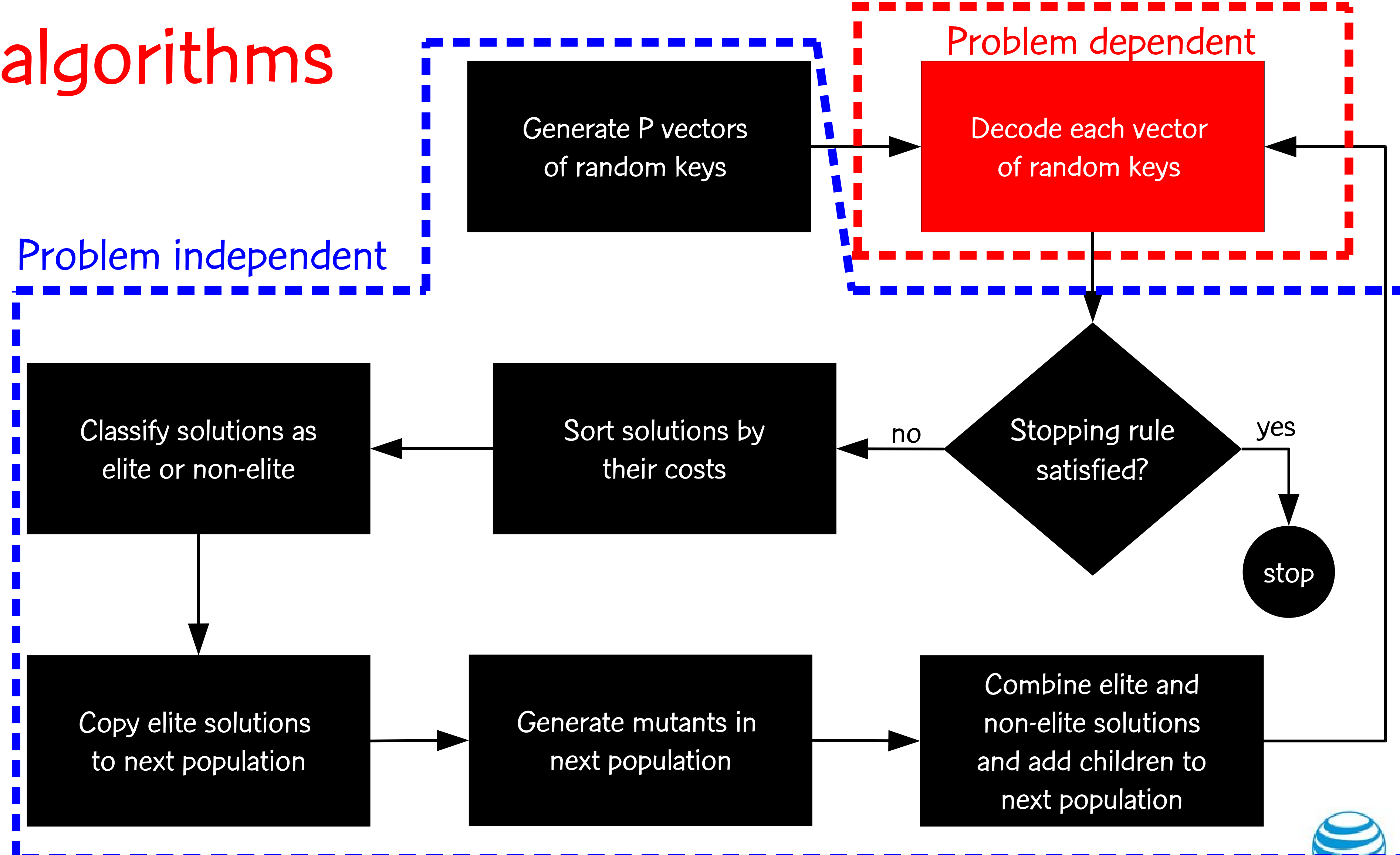
Framework for random-key genetic algorithms

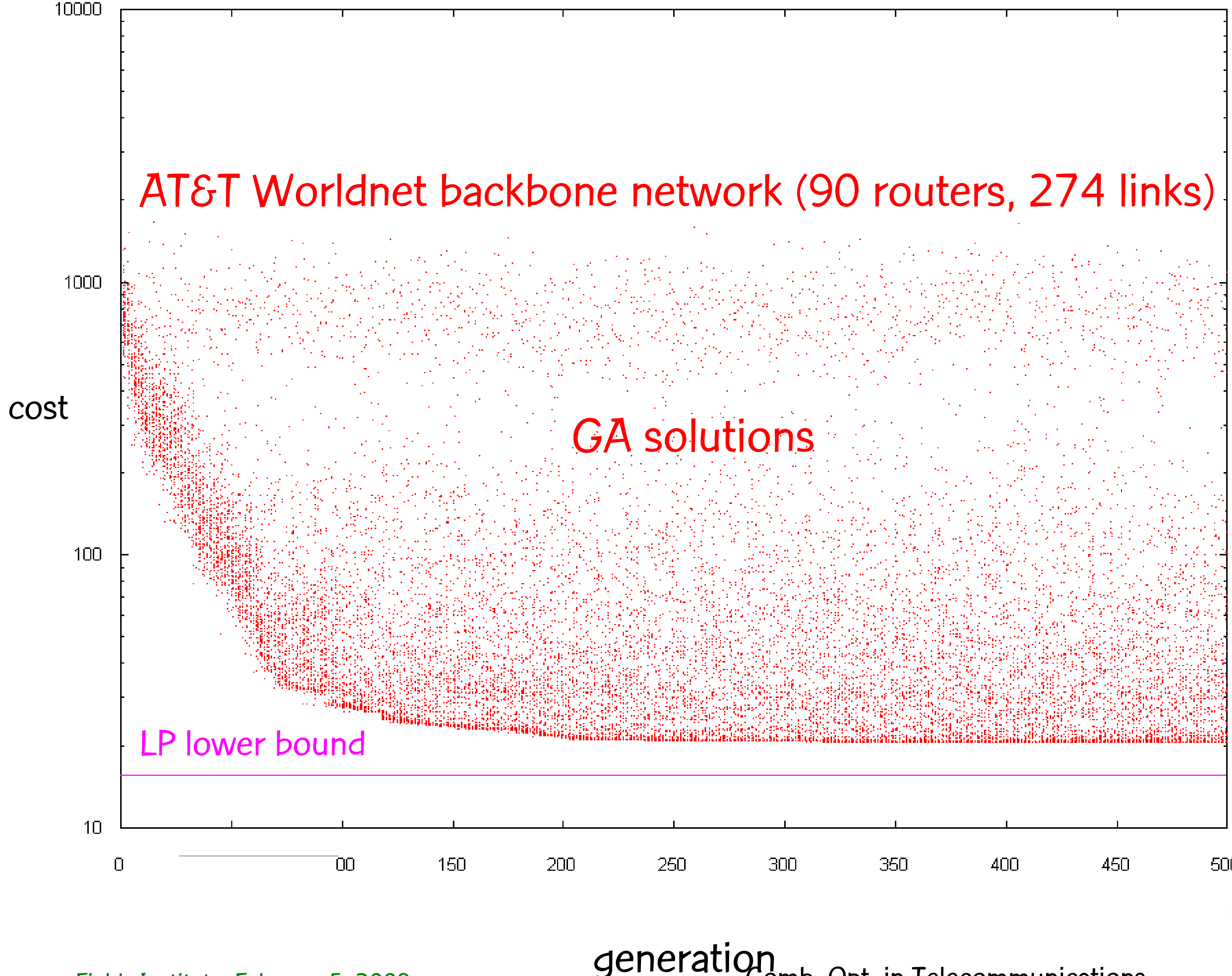


Framework for random-key genetic algorithms

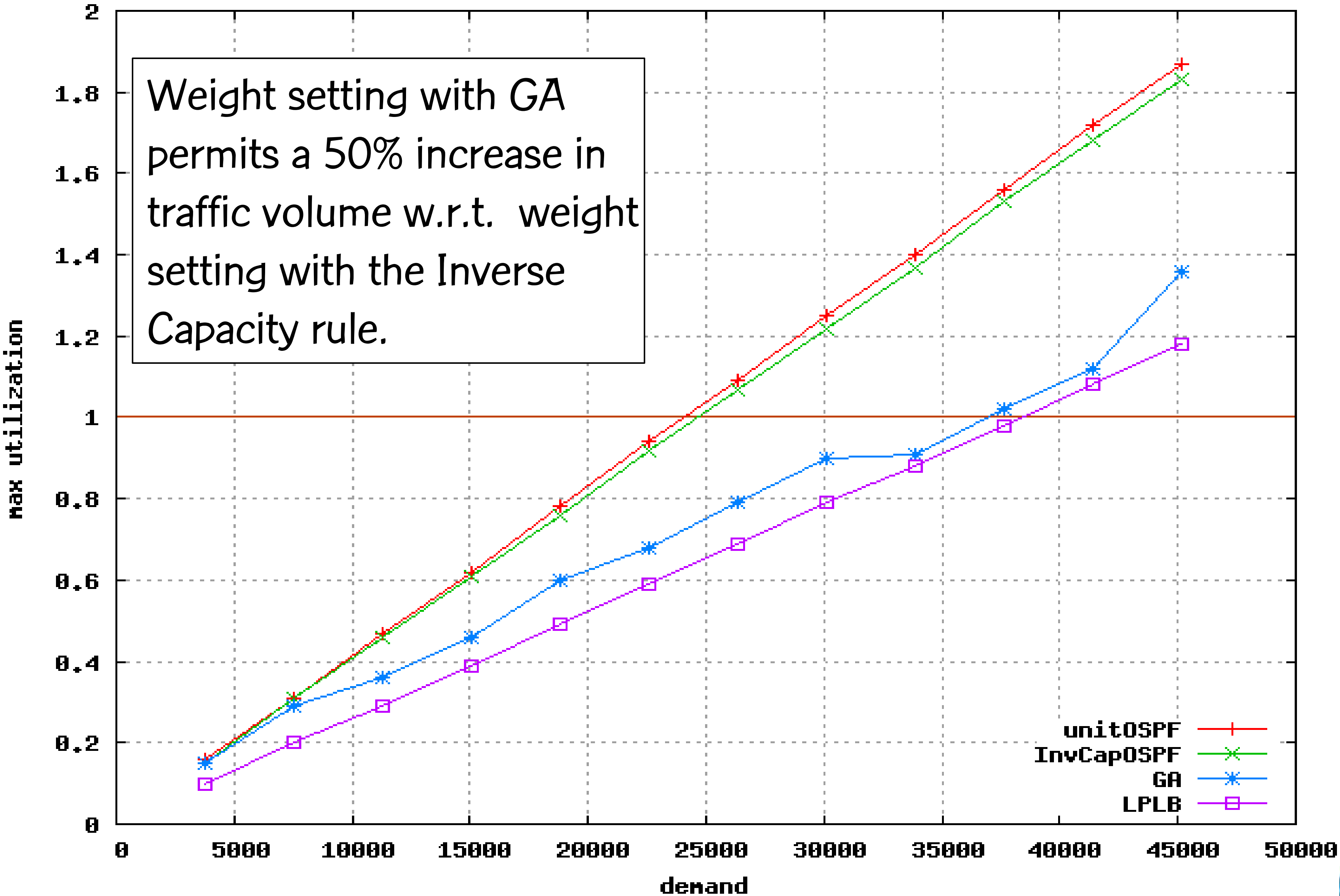


Framework for random-key genetic algorithms

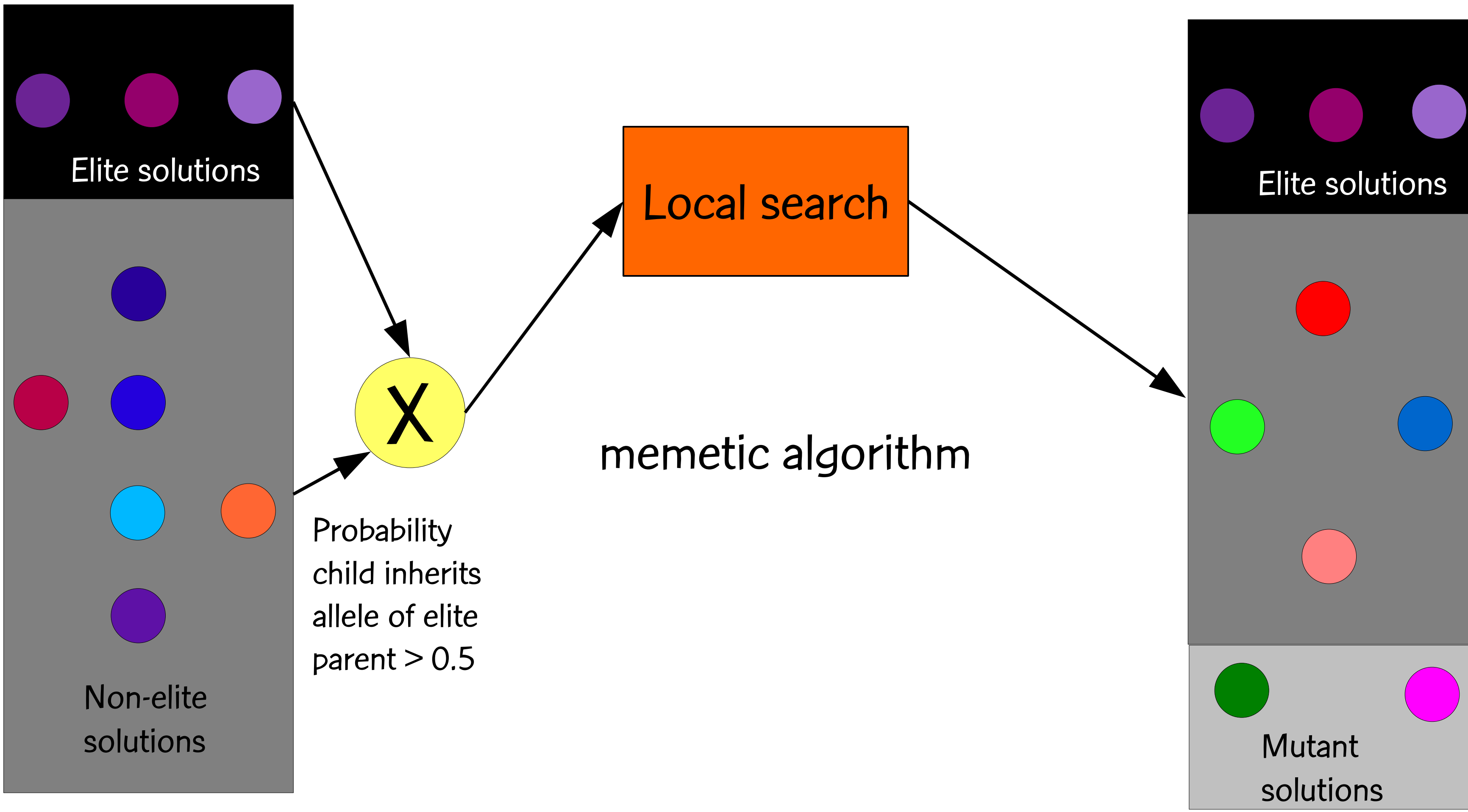



AT&T Worldnet backbone network (90 routers, 274 links)
GA solutions
LP lower bound
cost
generation
10000
1000
100
10
0
00
150
200
250
300
350
400
450
500

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



Optimized crossover = crossover + local search



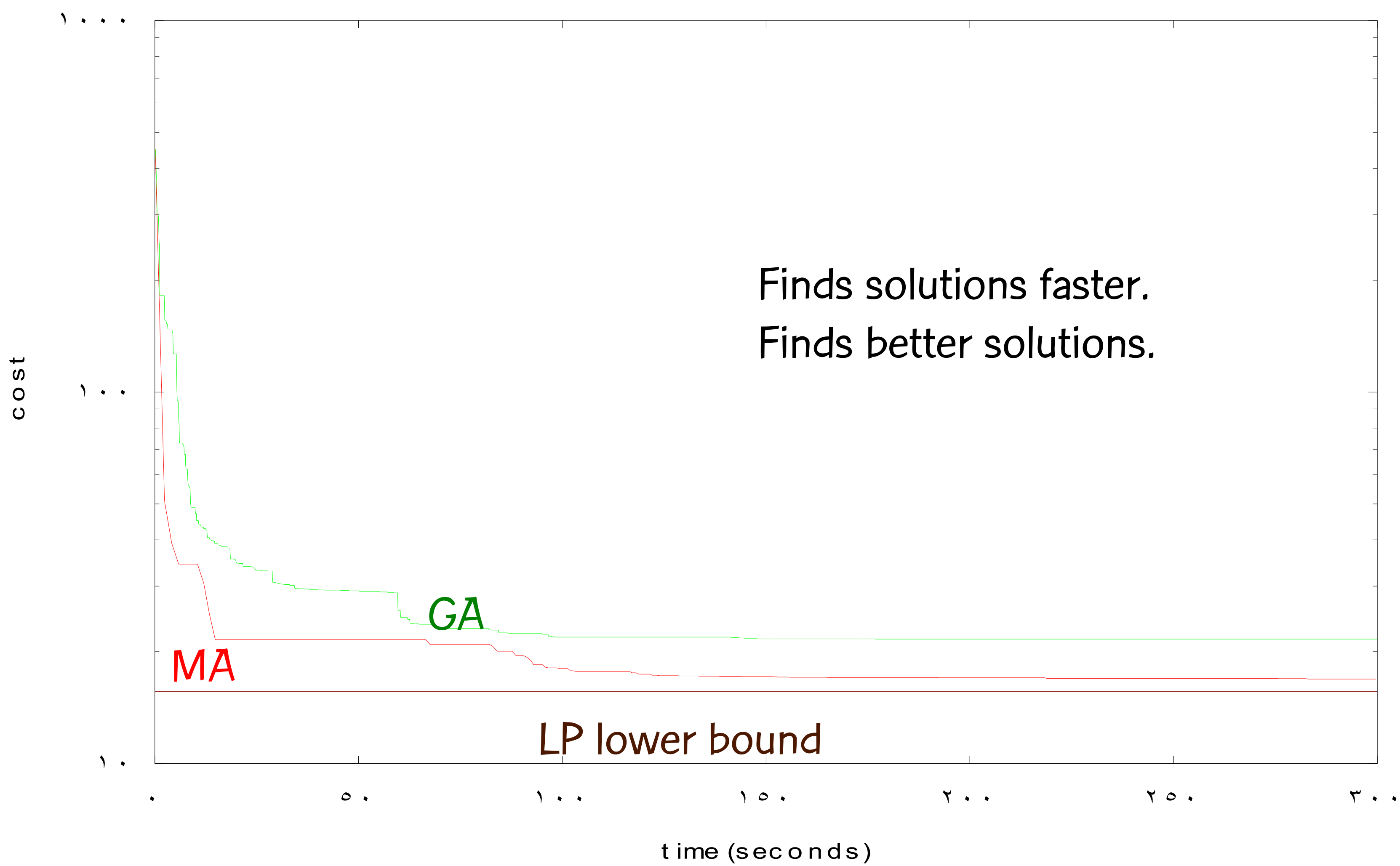
Fast local search

- 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

$$\left[w_a \,,\, w_a + \lceil (w_{max} - w_a)/4 \rceil \right]$$

  - If total cost Φ is reduced, restart local search.

Memetic algorithm (MA) improves over pure genetic algorithm in two ways:



Survivable IP network design

Survivable IP network design

Buriol, R., & Thorup (Networks, 2007)

- Given
  - directed graph $G = (N,A)$, where N is the set of routers, A is the set of potential arcs where capacity can be installed,
  - a demand matrix D that for each pair $(s,t) \in N \times N$, specifies the demand $D(s,t)$ between s and t,
  - a cost $K(a)$ to lay fiber on arc a
  - a capacity increment C for the fiber.
- Determine
  - OSPF weight $w(a)$ to assign to each arc $a \in A$,
  - which arcs should be used to deploy fiber and how many units (multiplicities) $M(a)$ of capacity C should be installed on each arc $a \in A$,
- such that all the demand can be OSPF routed on the network even when any single arc fails.
- Min total design cost $= \sum_{a \in A} M(a) \times K(a)$.

Survivable IP network design

- Chromosome:
  - A vector X of N random keys, where N is the number of links. The i-th random key corresponds to the i-th link weight.
- Decoder:
  - For i = 1,N: set $w(i) = \text{ceil}(X(i) \times w_{max})$
  - For each failure mode: route demand according to OSPF and for each link $i \in A$ determine load on link i.
  - For each link $i \in A$, compute multiplicity M(i) needed to accommodate maximum load over all failure modes.

  iterate

  - Network design cost = $\sum_{i \in A} M(i) \times K(i)$.

Survivable composite link IP network design

Andrade, Buriol, R., & Thorup (INFORMS Telecom. Conf., 2006)

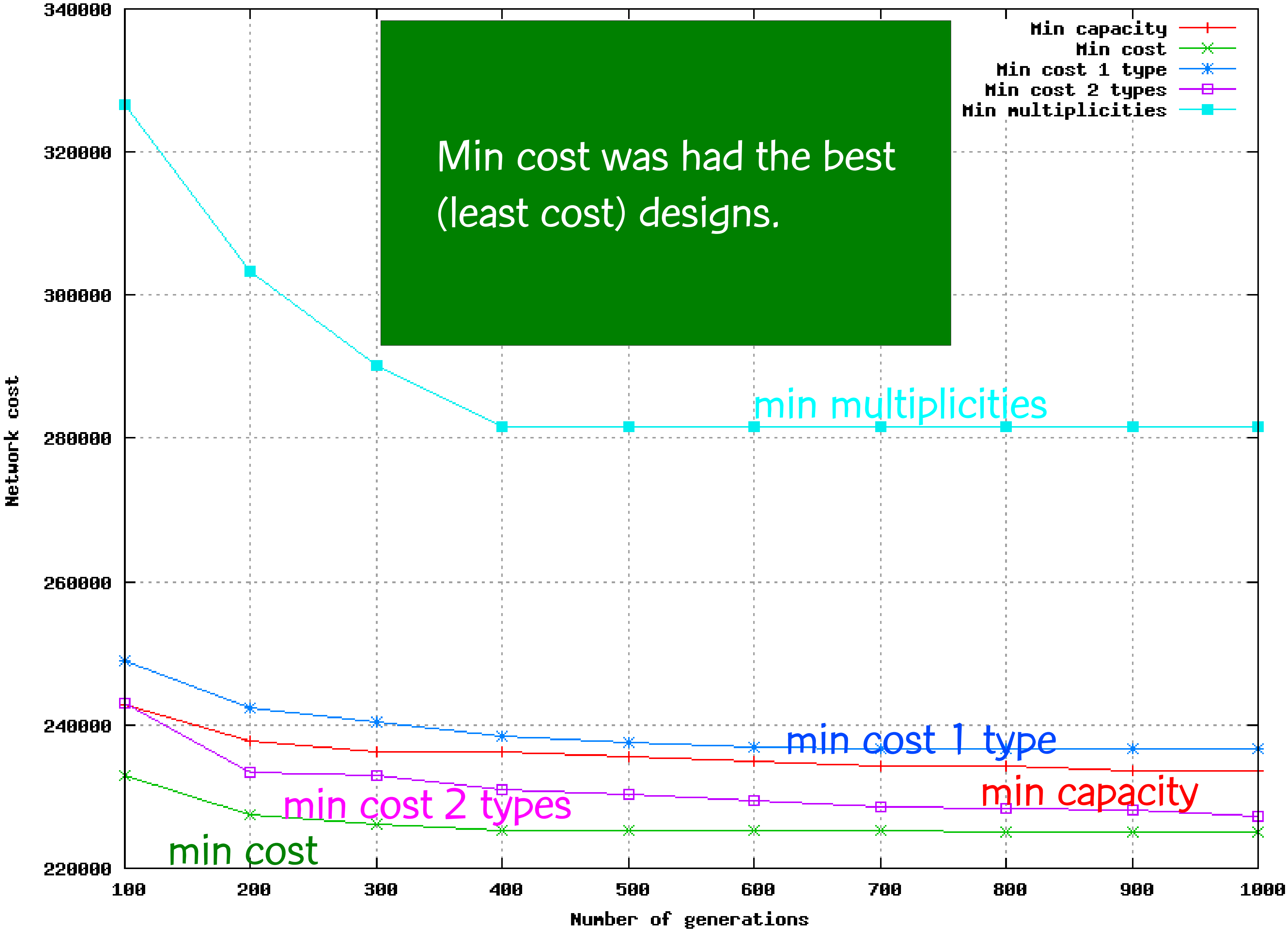
- Given a load $L(a)$ on arc a, we can compose several different link types that sum up to the needed capacity $c(a) \geq L(a)$:
  - $c(a) = \sum_{t \text{ used in arc } a} M(t,a) \times \gamma(t)$, where
  - $M(t,a)$ is the multiplicity of link type $t \in \{ 1, 2, \ldots, T \}$ on arc a
  - $\gamma(t)$ is the capacity of link type t: $\{ \gamma(1), \gamma(2), \ldots, \gamma(T) \}$ such that $\gamma(i) < \gamma(i+1)$
- Assumptions
  - Prices / unit length = $\{ p(1), p(2), \ldots, p(T) \}$: $p(i) < p(i+1)$
  - $[p(T)/\gamma(T)] < [p(T-1)/\gamma(T-1)] < \cdots < [p(1)/\gamma(1)]$: economies of scale
  - $\gamma(i) = \alpha \times \gamma(i-1)$, for $\alpha \in N$, $\alpha > 1$, e.g.
    - $\gamma(OC192) = 4 \times \gamma(OC48)$
    - $\gamma(OC48) = 4 \times \gamma(OC12)$
    - $\gamma(OC12) = 4 \times \gamma(OC3)$

Survivable composite link IP network design

- Chromosome:
  - A vector X of N random keys, where N is the number of links. The i-th random key corresponds to the i-th link weight.
- Decoder:
  - For i = 1,N: set $w(i) = \text{ceil}\,(X(i) \times w_{max})$
  - For each failure mode: route demand according to OSPF and for each arc $i \in A$ determine the load on arc i.
  - For each arc $i \in A$, determine the multiplicity M(t,i) for each link type t using the maximum load for that arc over all failure modes.
  - Network design cost = $\sum_{i \in A} \sum_{t \text{ used in arc } i} M(t,i) \times p(t)$

iterate

Min cost was had the best (least cost) designs.



IP network design with OSPF routing

- Simple link design
  - L.S. Buriol, M.G.C.R., and M. Thorup, "Survivable IP network design with OSPF routing," Networks, vol. 49, pp. 51-64, 2007.
- Composite link design
  - D.V. Andrade, L.S. Buriol, M.G.C.R., and M. Thorup, "Survivable composite-link IP network design with OSPF routing," Proceedings of The Eighth INFORMS Telecommunications Conference, Dallas, Texas, April 2006.

Covering by pairs

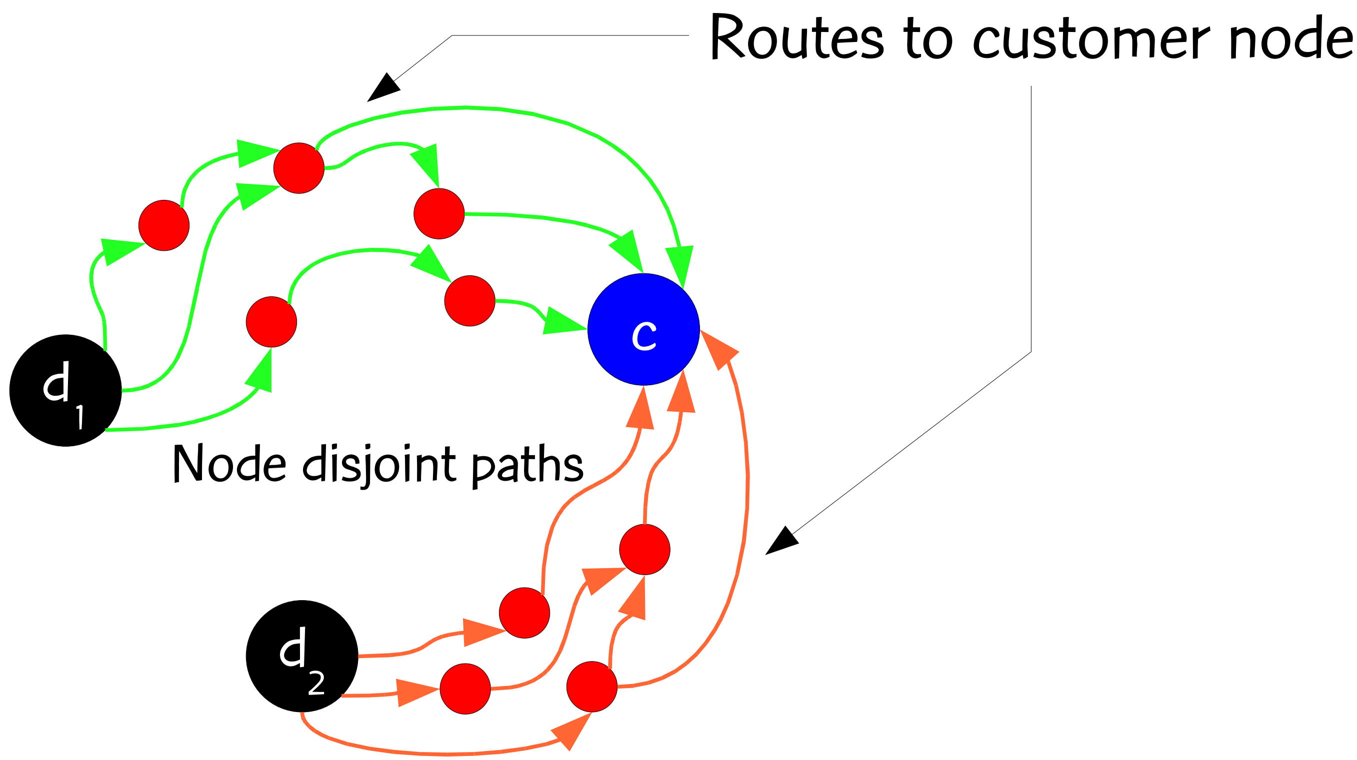
Covering by pairs

- Application to location of IPTV data centers.
- L. Breslau, I. Diakonikolas, N. Duffield, Y. Gu, M. Hajiaghayi, D.S. Johnson, H, Karloff, M.G.C.R., S. Sen, and D. Towsley, "Optimal Node Placement for Path Disjoint Network Monitoring," AT&T Labs Research Technical Report TD-7945U7, November 18, 2007.

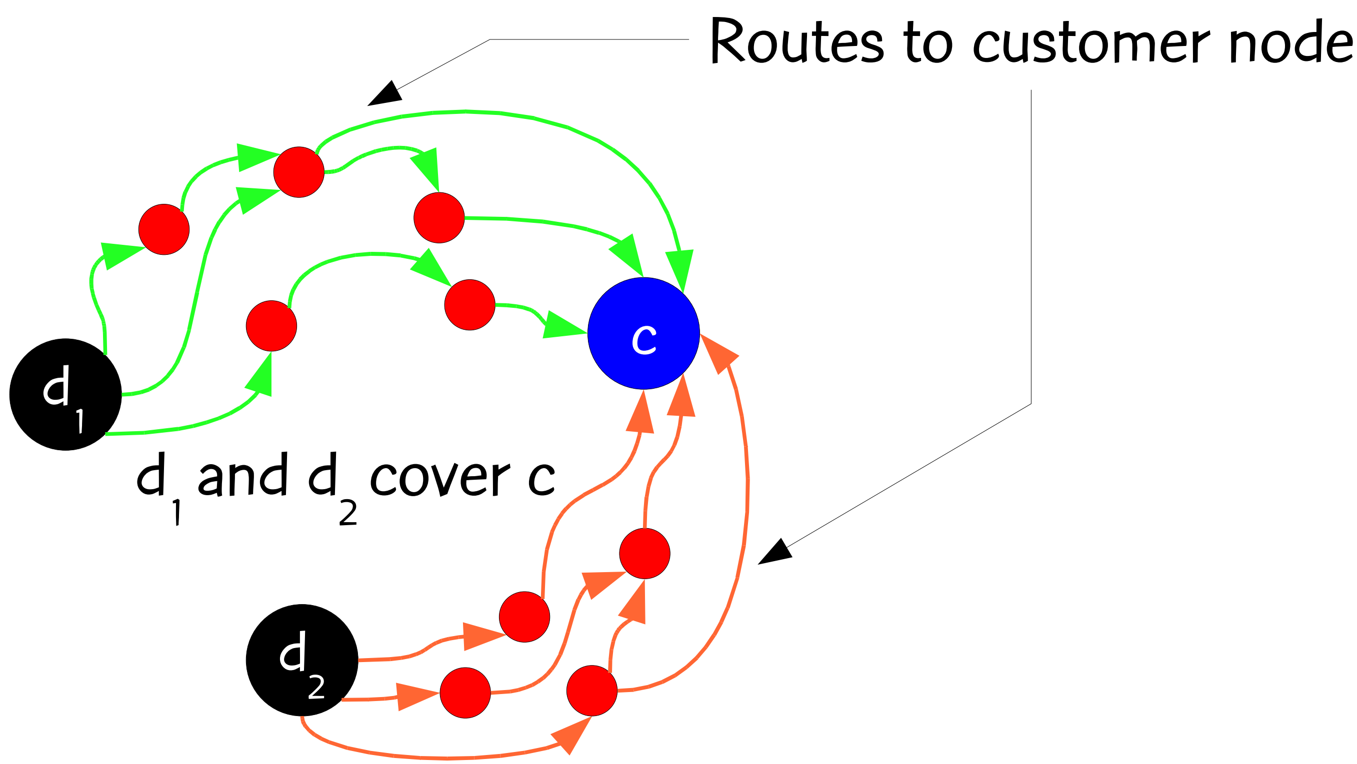
Covering by pairs

- Given a directed network $G = (N,V)$, one or more paths between all pairs of nodes in N, a subset D of nodes called (potential) data nodes, a subset C of nodes called customer nodes, we wish to select the smallest number of data nodes nodes such that for each customer node $c \in C$, either:
  - There are at least two data nodes d_1 and $d_2 \in D$ such that all routes from d_1 to c are node disjoint with all routes from d_2 to c, or
  - Node c is one of the selected data nodes (c covers itself)

Covering by pairs

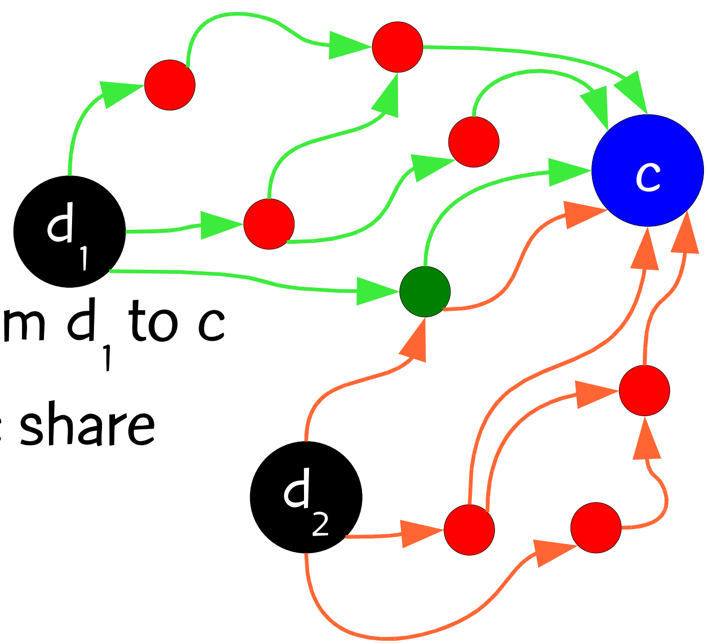


Covering by pairs



Covering by pairs

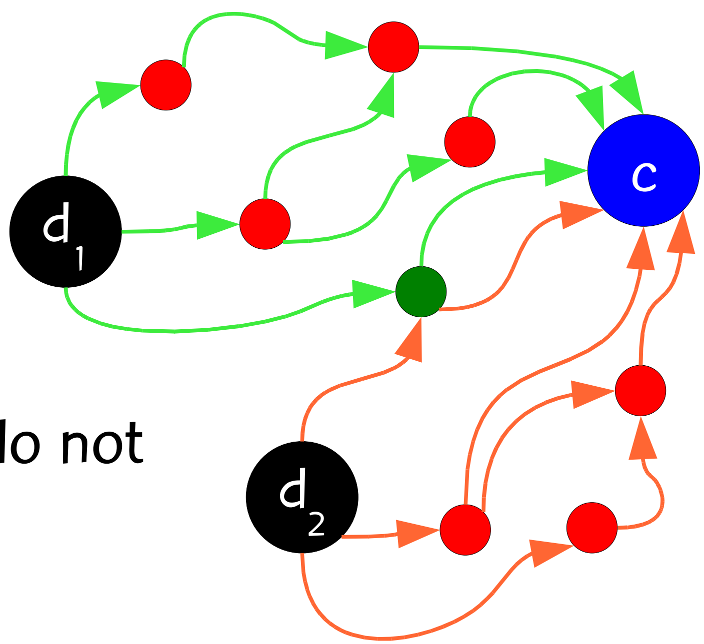
Routes to customer node



Routes from d_1 to c and d_2 to c share node

Covering by pairs

Routes to customer node



d_1 and d_2 do not cover c

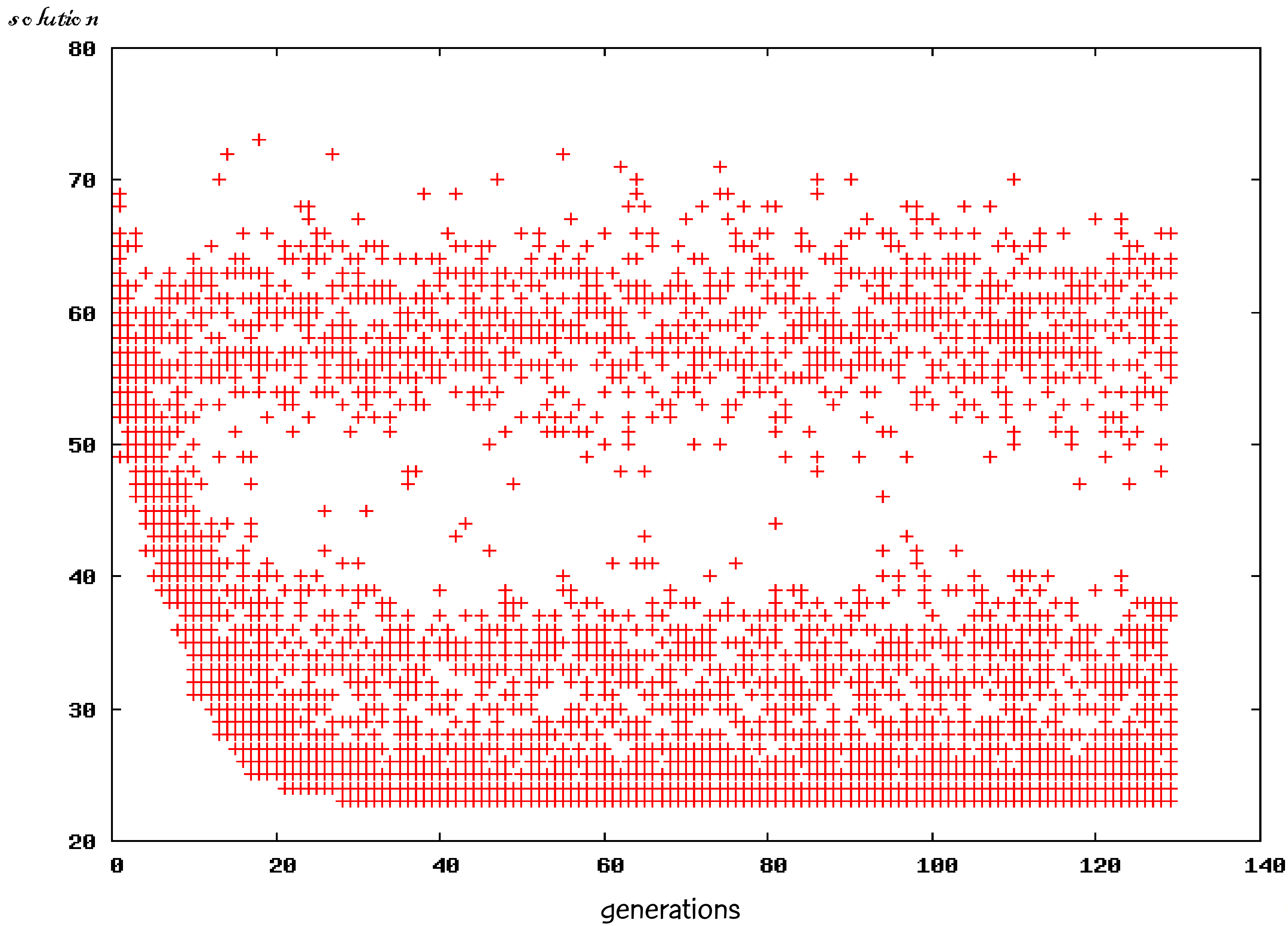
Covering by pairs

- Chromosome:
  - A vector X of N random 0-1values (random keys), where N is the number of data nodes. The i-th random key corresponds to the i-th data node.
- Decoder:
  - For i = 1,N: if X(i) = 1, add i-th data node to solution
  - If solution is feasible, i.e. all customer nodes are covered: STOP
  - Else, apply greedy algorithm to cover uncovered customer nodes.

Covering by pairs

- Size of population: N (number of data nodes)
- Size of elite set: 15% of N
- Size of mutant set: 10% of N
- Biased coin probability: 70%
- Stop after N generations without improvement of best found solution

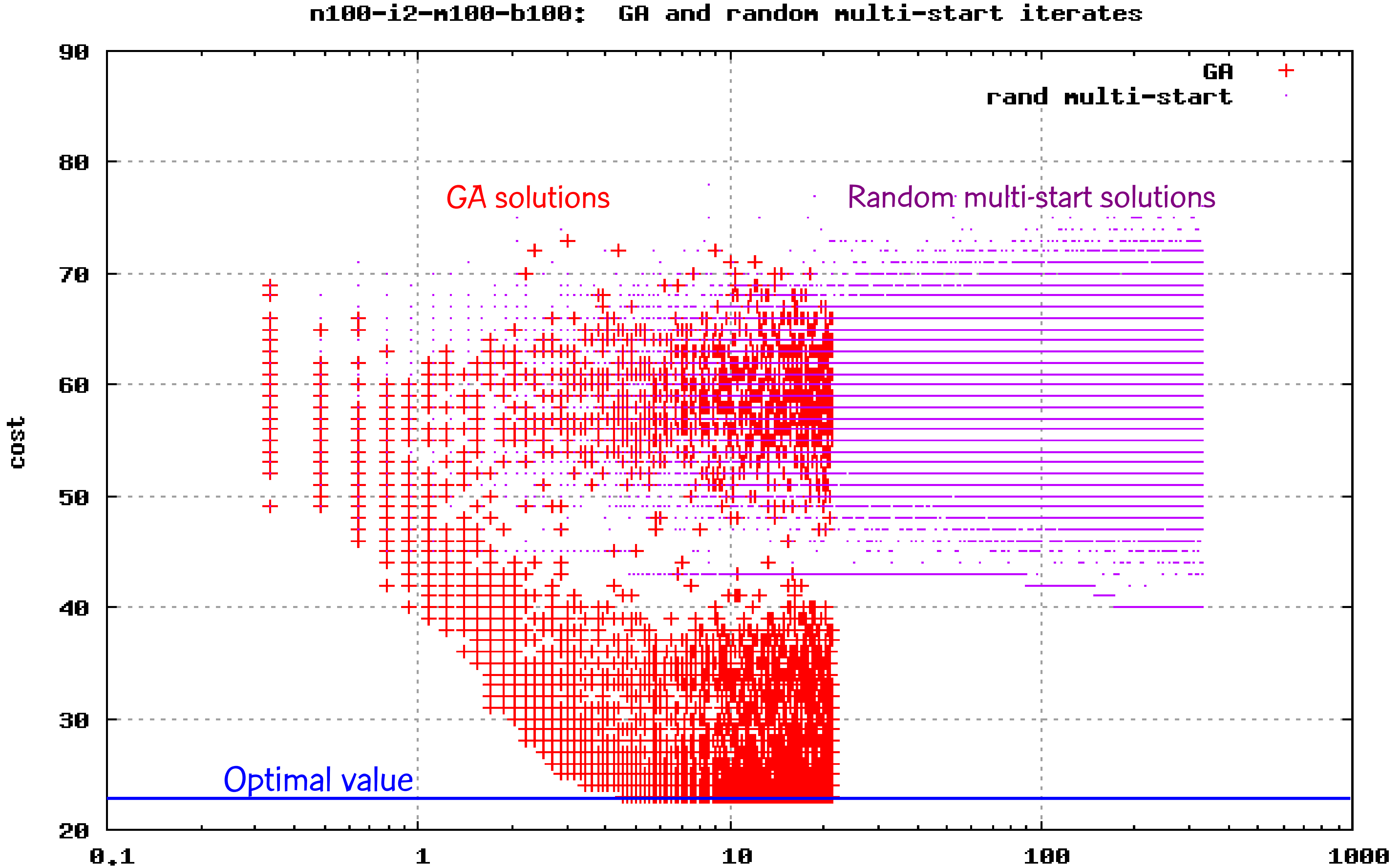
n100-i2-m100-b100 (opt = 23)



at&t
Your world. Delivered.

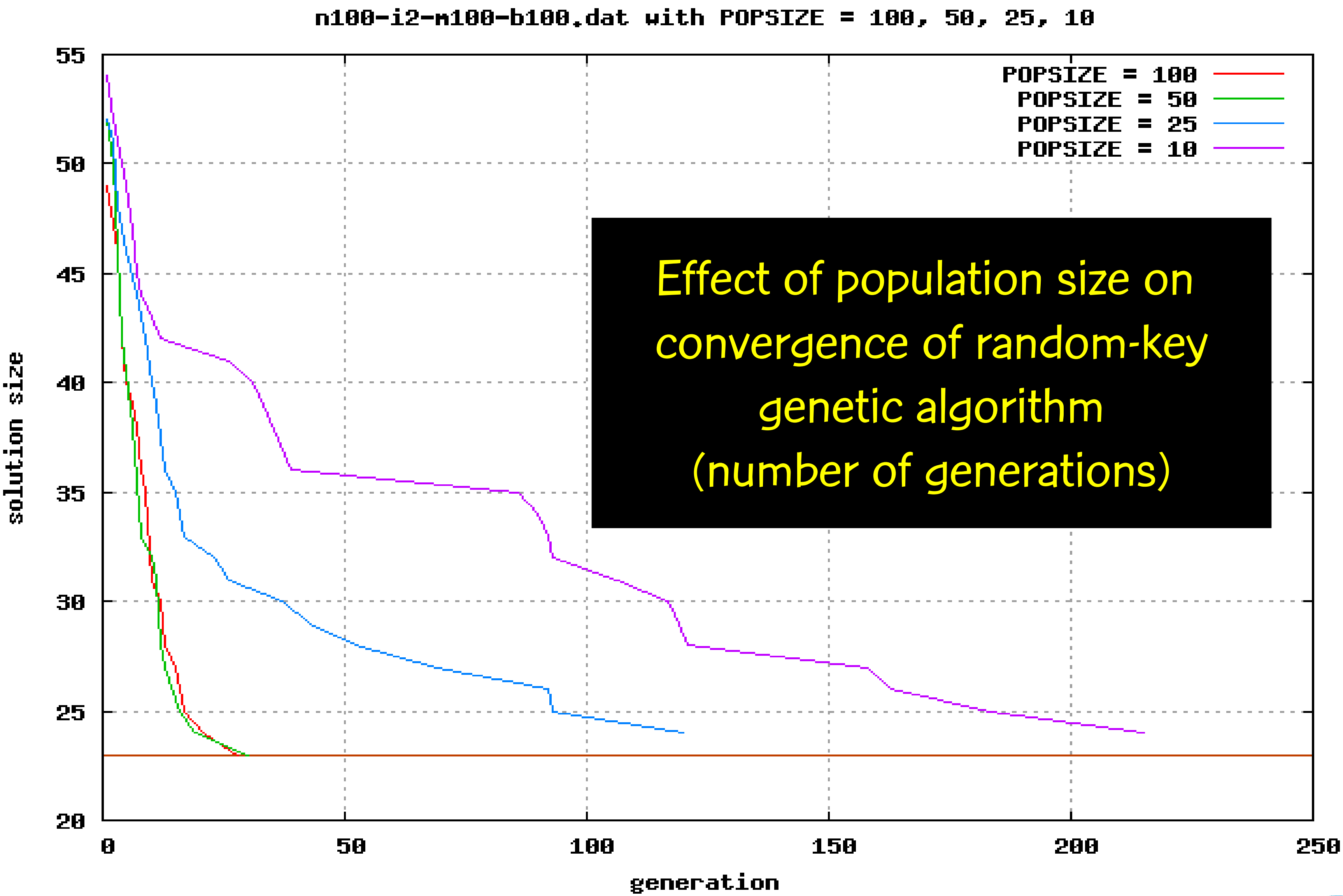
n100-i2-m100-b100 (opt = 23)

solution



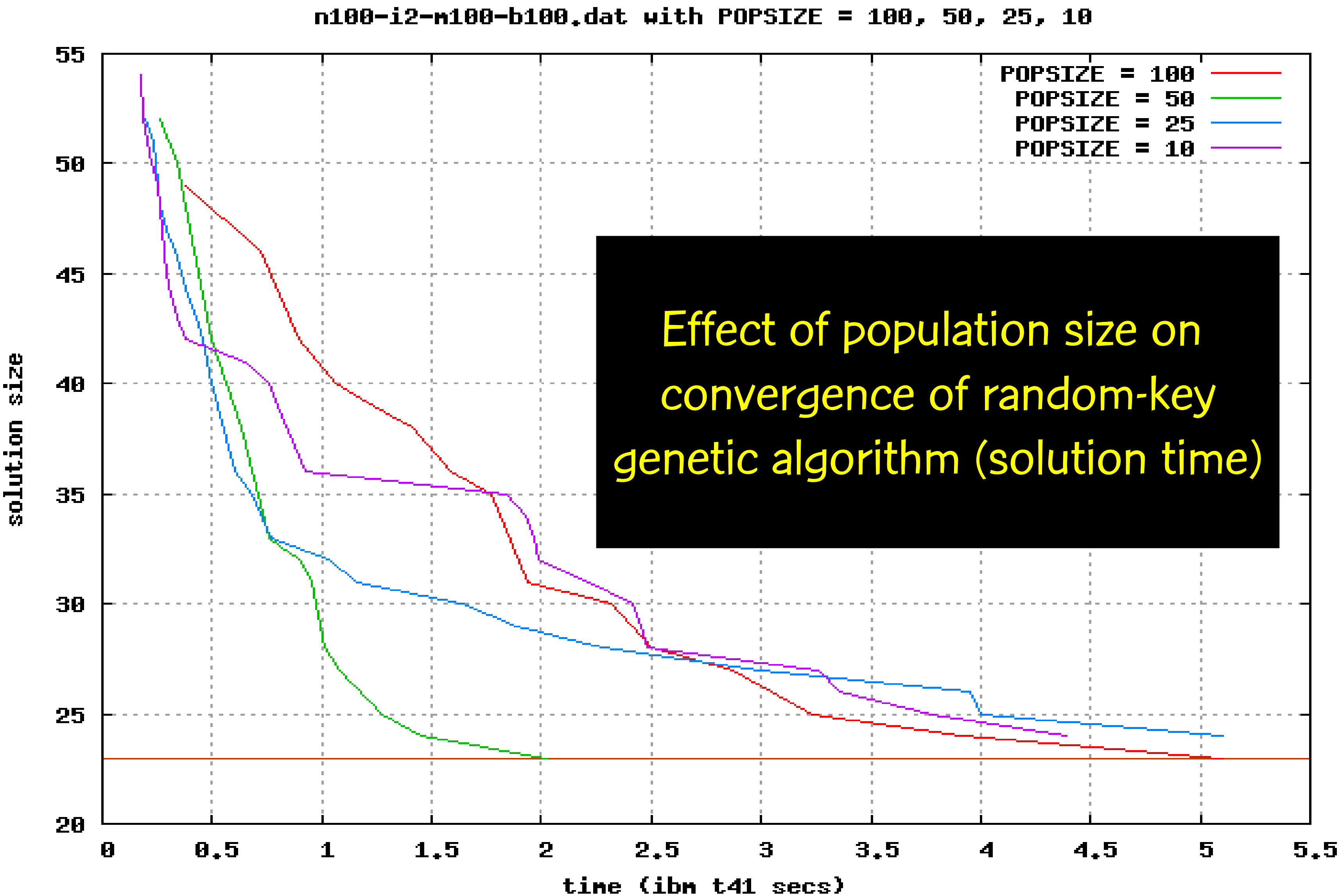
Time (ibm t41 secs)

n100-i2-m100-b100 (opt = 23)



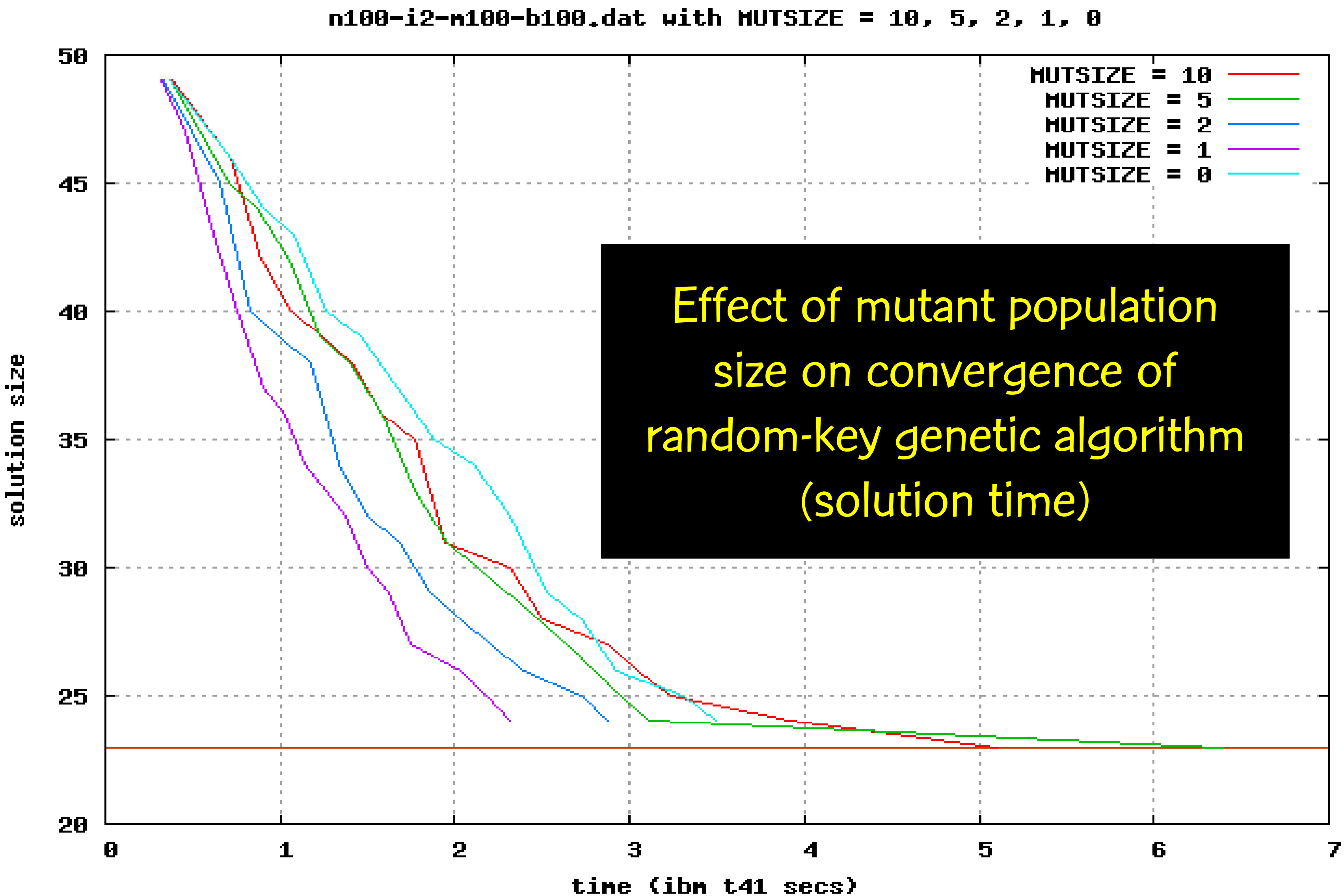
at&t Your world. Delivered.

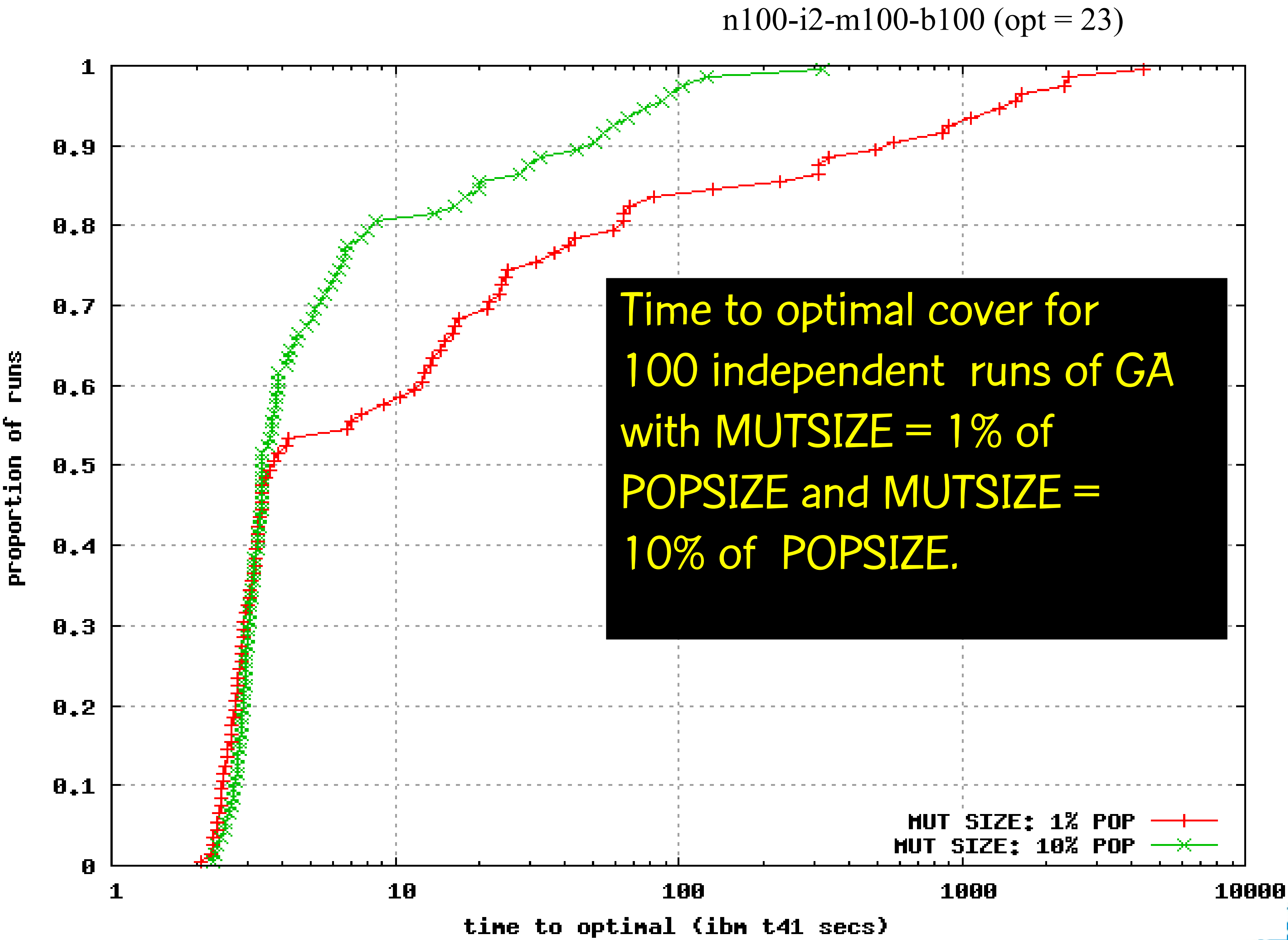
n100-i2-m100-b100 (opt = 23)



Effect of population size on convergence of random-key genetic algorithm (solution time)

n100-i2-m100-b100 (opt = 23)





n100-i2-m100-b100 (opt = 23)
Time to optimal cover for 100 independent runs of GA with MUTSIZE = 1% of POPSIZE and MUTSIZE = 10% of POPSIZE.
proportion of runs
time to optimal (ibm t41 secs)
MUT SIZE: 1% POP
MUT SIZE: 10% POP
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1
10
100
1000
10000

Experimental results

- 560 instances, with 25, 50, 100, 190, 220, 250, 300, and 558 nodes.
- 324 of these 560 were solved optimally with CPLEX. Running GA a single time, we found optimal solutions in 318 of these instances.
- Of the 236 that CPLEX could not solve, GA matched a lower bound in 166.
- In all, the GA found optimal solutions for 484 of the 560 instances (86.4%)

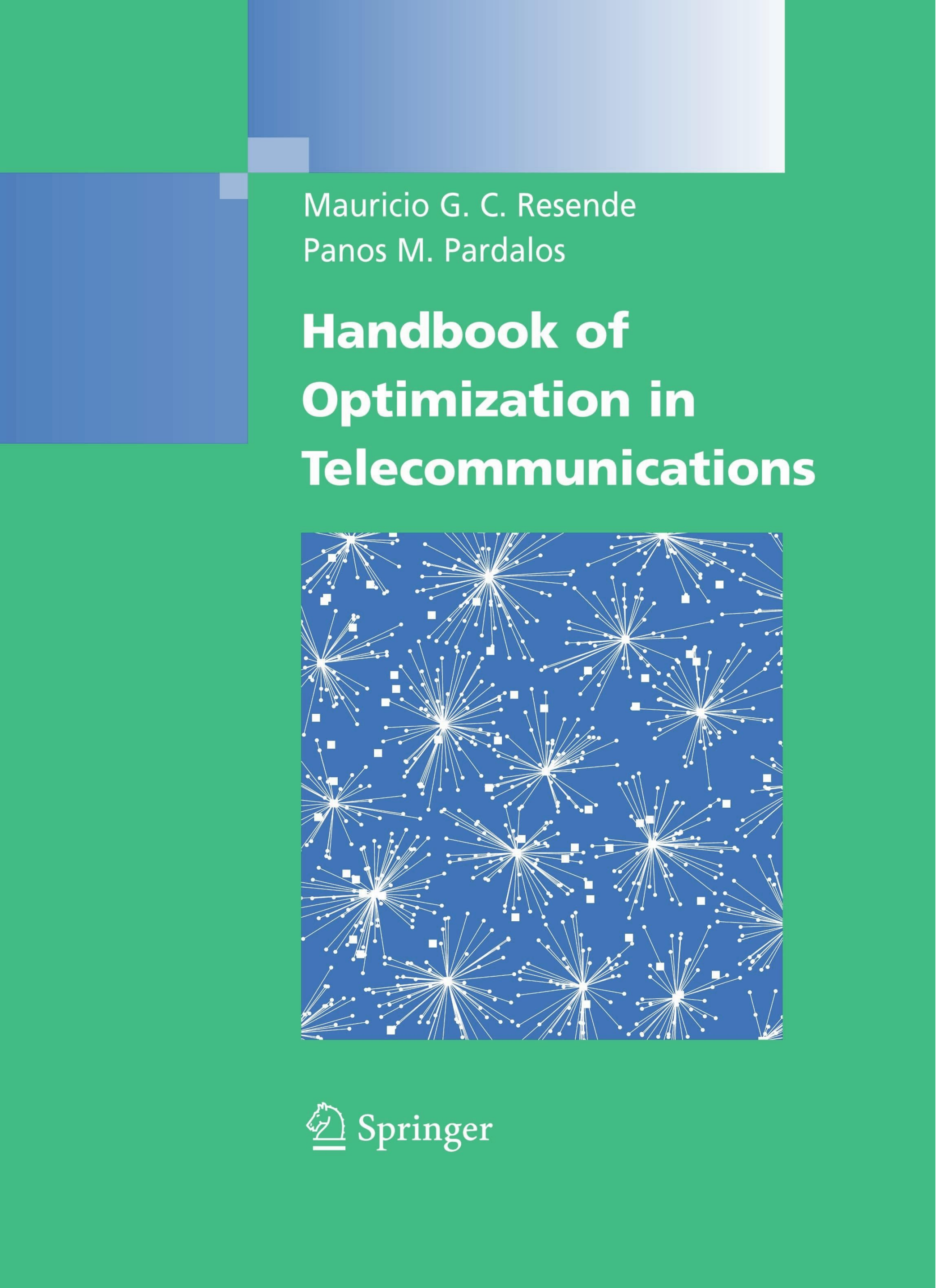
Experimental results

- The paper describes the double hitting set heuristic (HH) proposed by H. Karloff. This heuristic makes use of the OSPF paths and is very fast and effective.
- In 482 of the 560 instances (86.1%) the GA and HH found solutions with the same cost.
- In 68 instances (12.1%) GA found a better solution than HH.
- In 10 instances (2%) HH found a better solution than GA.
- In only 12 instances (2.1%) was the solution found by GA not minimal.

Concluding remarks

Concluding remarks

- We have seen just a few examples of combinatorial optimization problems that arise in telecommunications.
- The field is fertile ground for optimization research involving
  - exact methods,
  - heuristics,
  - approximation algorithms, ...



Published by Springer in April 2006

1134 pages
37 chapters in five parts:

Optimization algorithms
Planning and design
Routing
Wireless
The web and beyond

The End

These slides and all of my papers cited in this talk can be downloaded from my homepage:

http://mauricioresende.com