Survivable composite-link IP network design with OSPF routing

Talk given at
Eighth INFORMS Telecommunications Conference
Dallas, Texas
April 1, 2006

Mauricio G. C. Resende

AT&T Labs Research
Florham Park, New Jersey
mgcr@research.att.com
www.research.att.com/~mgcr

Joint work with D.V. Andrade, L.S. Buriol & M. Thorup

Summary of talk

- OSPF routing
- Survivable IP network design
- Composite-link design
- Concluding remarks

Your world. Delivered.

Summary of talk

- OSPF routing
- Survivable IP network design
- Composite-link design
- Concluding remarks

Summary of talk

- OSPF routing
- Survivable IP network design
- Composite-link design
- Concluding remarks

Your world. Delivered.

Summary of talk

- OSPF routing
- Survivable IP network design
- Composite-link design
- Concluding remarks

OSPF routing

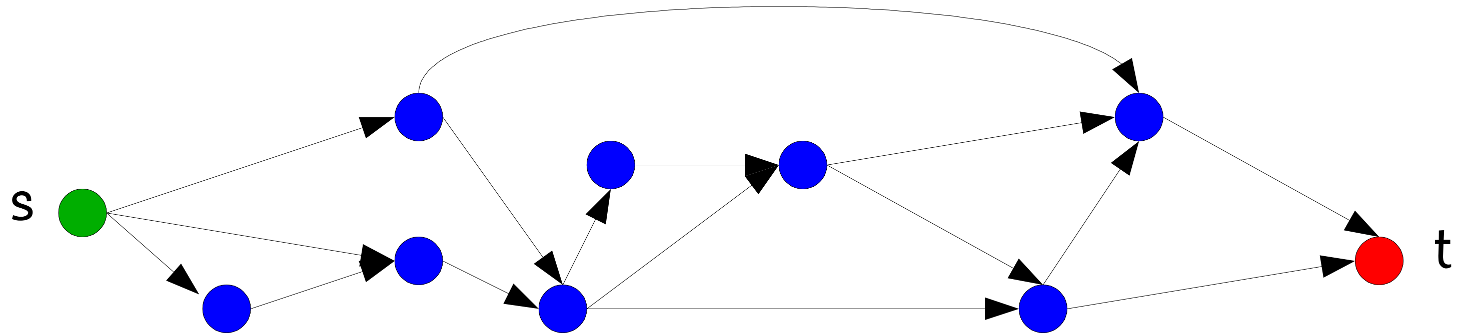
- Given a network $G = (N,A)$, where N is the set of routers and A is the set of links.

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.

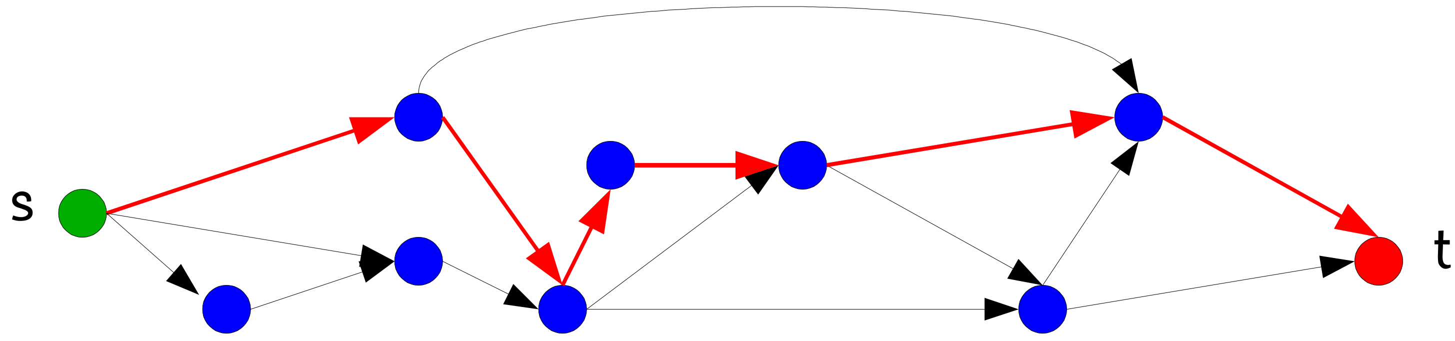
OSPF routing

- 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.



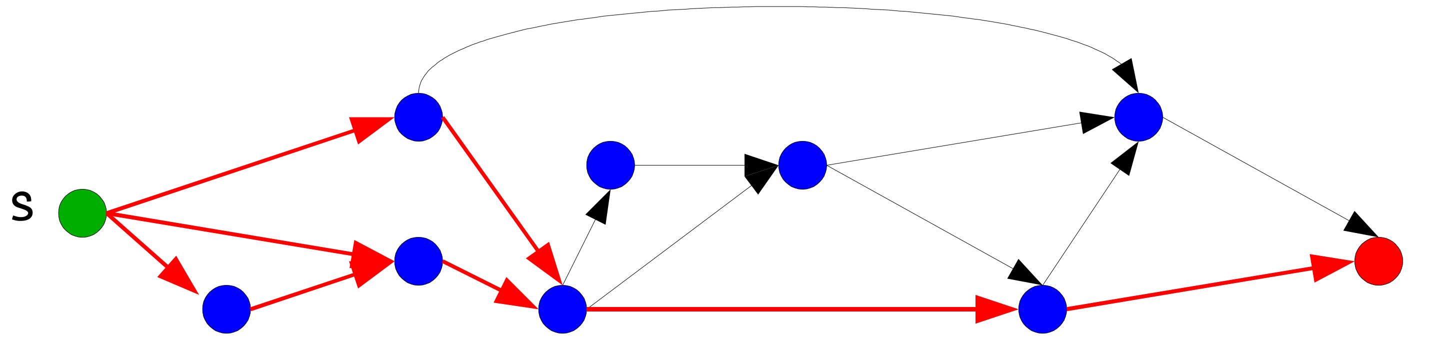
OSPF routing

- 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.



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.



Traffic splitting

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, Resende, & Pardalos (2002)
  - Buriol, Resende, 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, Resende, & Pardalos (2002)
  - Buriol, Resende, Ribeiro, & Thorup (2002, 2005)

at&t
Your world. Delivered.

Survivable IP network design

- Buriol, Resende, and Thorup (Networks, 2006) use weight setting to design survivable IP networks.

Survivable IP network design

- Buriol, Resende, and Thorup (Networks, 2006) use weight setting to design survivable IP networks.
- 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.

Survivable IP network design

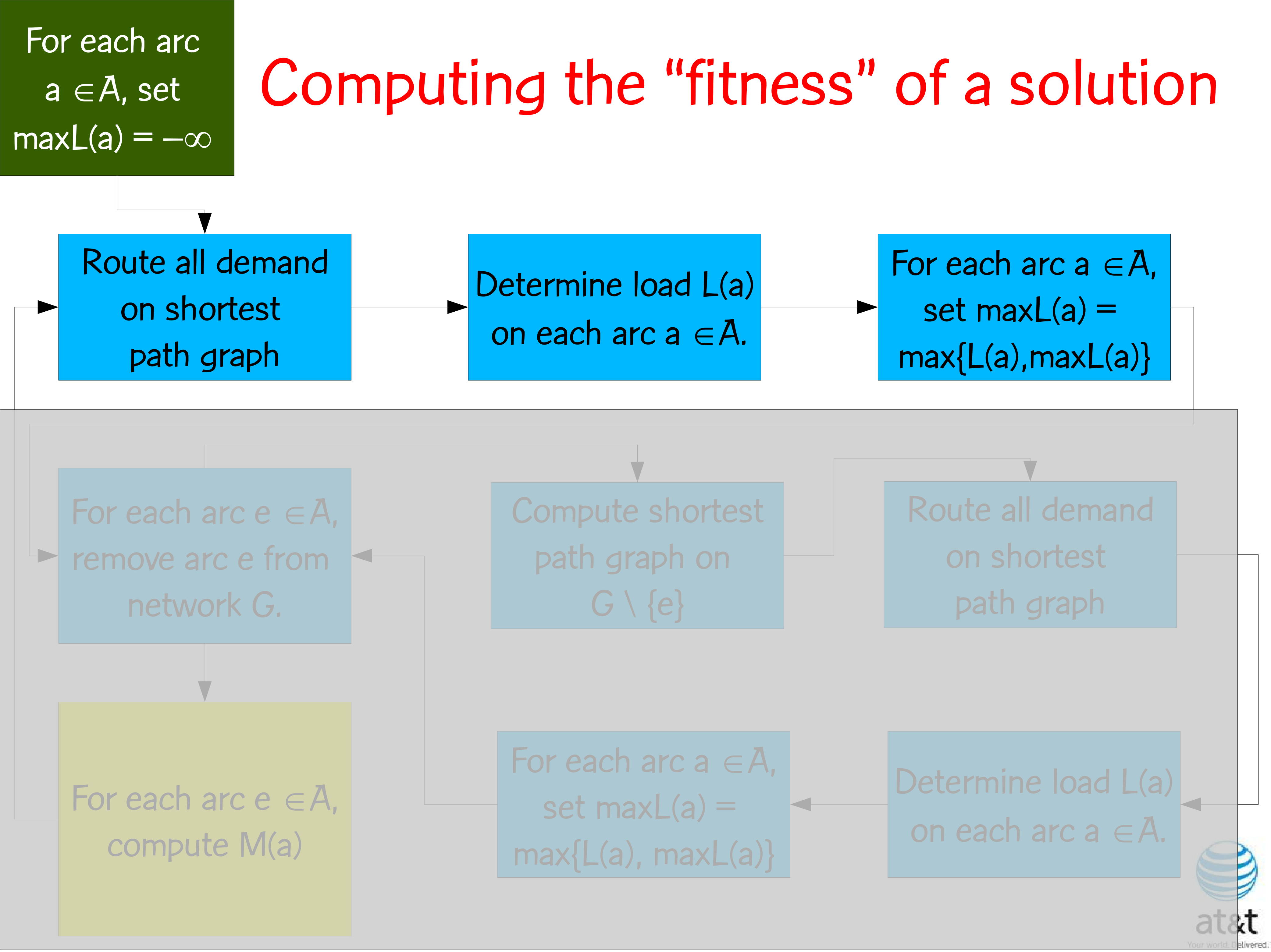
- Buriol, Resende, and Thorup (Networks, 2006) use weight setting to design survivable IP networks.
- Determine
  - OSPF weight w(a) to assign to each arc $a \in A$,
  - which arcs should be used to deploy fiber and how many units (multiplicities) M(a) of capacity C should be installed on each arc $a \in A$,
- such that all the demand can be routed on the network even when any single arc fails.
- Minimize total design cost = $\sum_{a \in A} M(a) \times K(a)$.

at&t
Your world. Delivered.

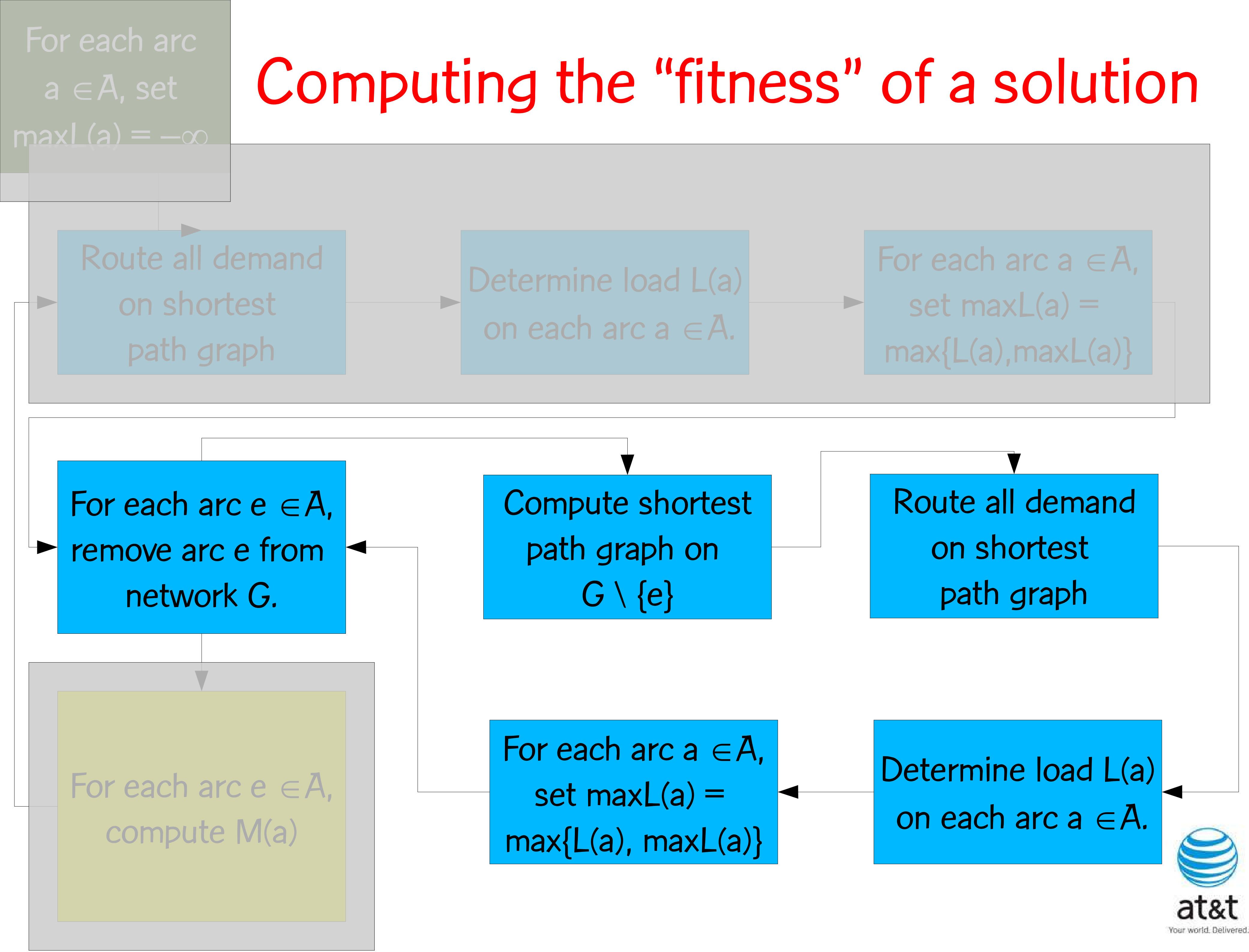
Survivable IP network design

- Buriol, Resende, and Thorup (Networks, 2006) use weight setting to design survivable IP networks.
- Use genetic algorithm (GA) to determine weights.
- GA needs to compute "fitness" of each solution it produces.

Computing the "fitness" of a solution

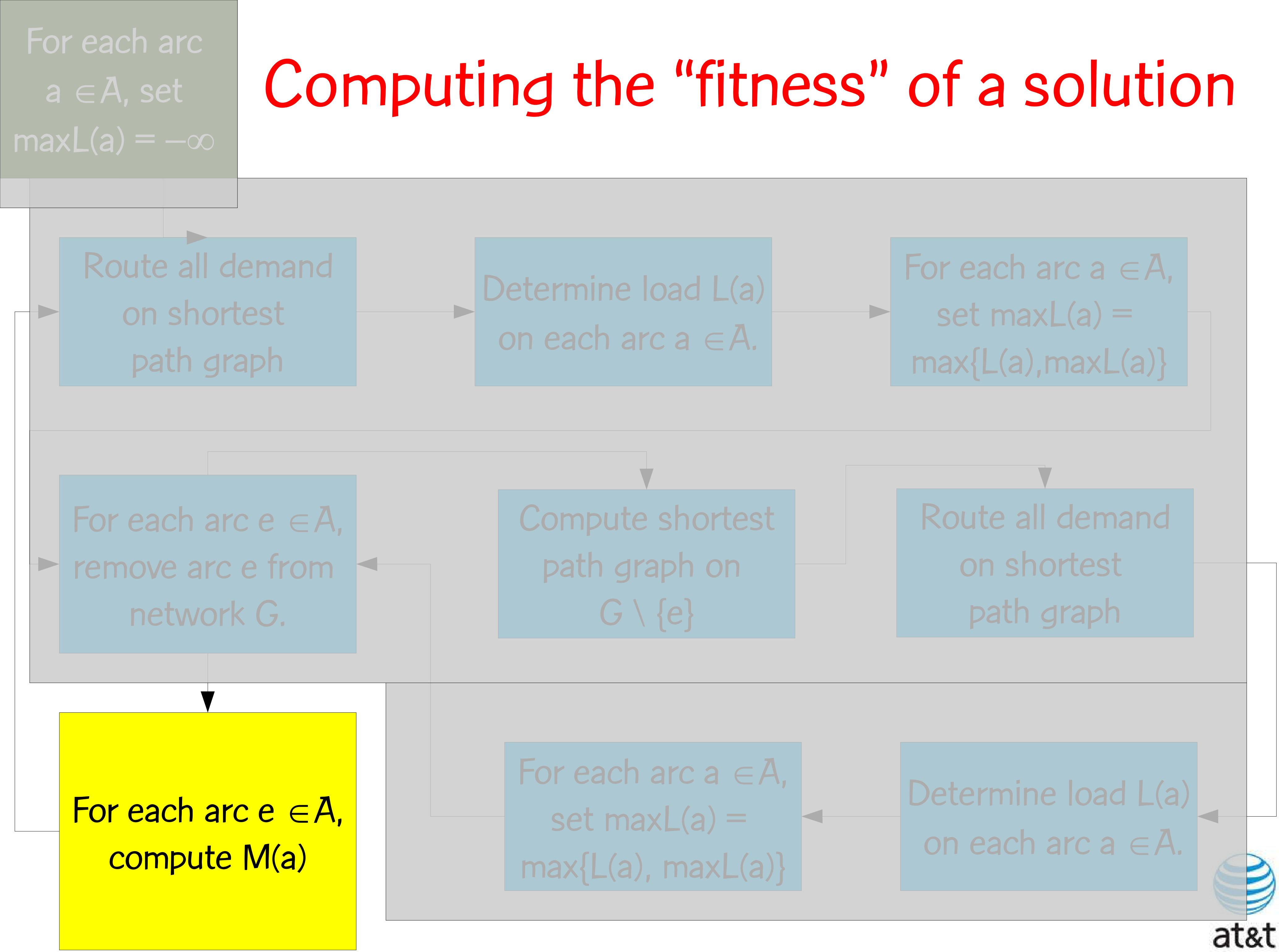


Computing the "fitness" of a solution

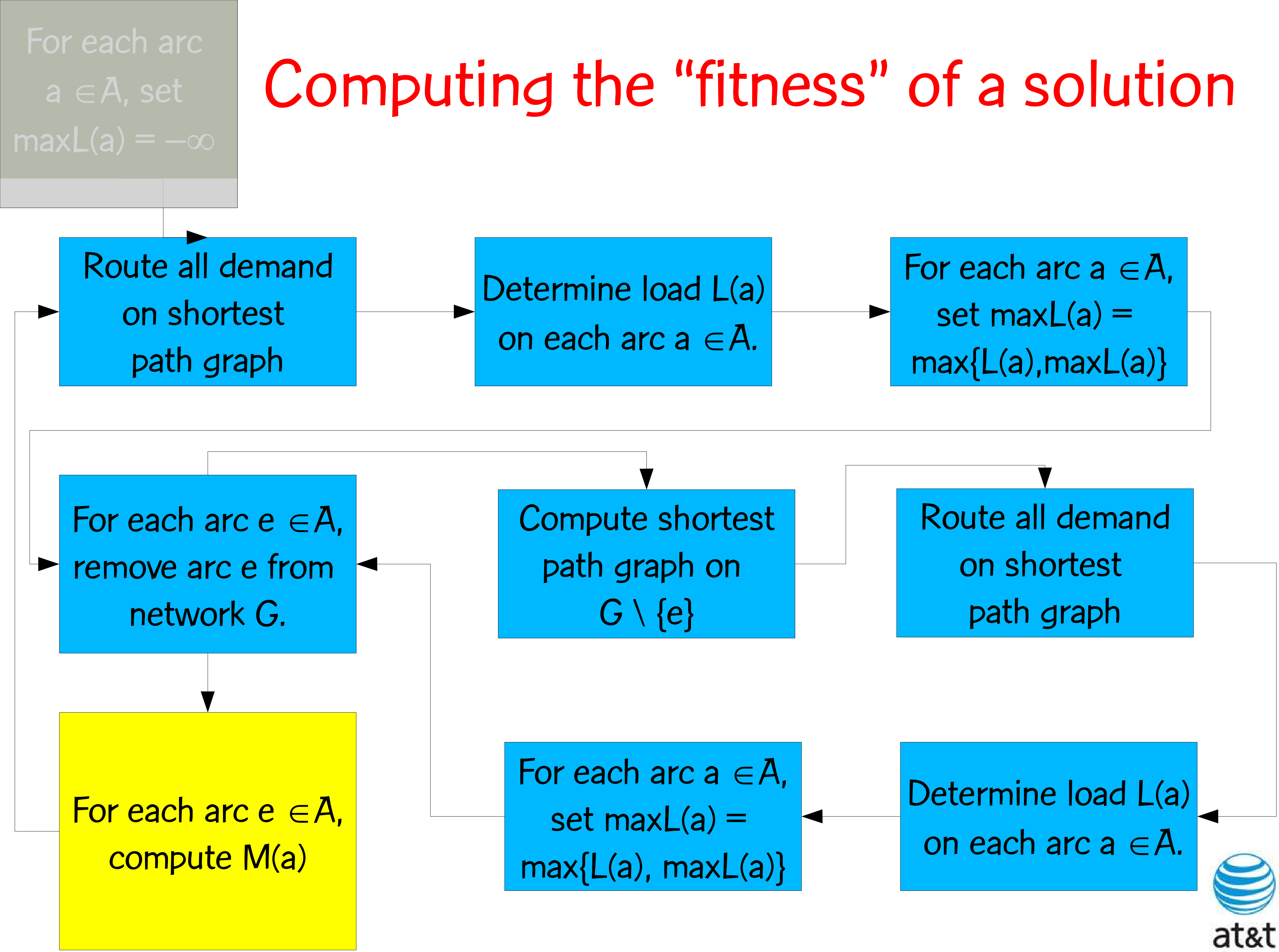


at&t
Your world. Delivered.

Computing the “fitness” of a solution



Computing the "fitness" of a solution



at&t
Your world. Delivered.

Composite-link design

- In Buriol, Resende, and Thorup (2006)
  - links were all of the same type,
  - only the link multiplicity had to be determined.
- Now consider composite links. Given a load $L(a)$ on arc a, we can compose several different link types that sum up to the needed capacity $c(a) \geq L(a)$:
  - $c(a) = \sum_{t \text{ used in arc } a} M(t) \times \gamma(t)$, where
  - $M(t)$ is the multiplicity of link type t
  - $\gamma(t)$ is the capacity of link type t

Composite-link design

- In Buriol, Resende, and Thorup (2006)
  - links were all of the same type,
  - only the link multiplicity had to be determined.
- Now consider composite links. Given a load L(a) on arc a, we can compose several different link types that sum up to the needed capacity $c(a) \geq L(a)$:
  - $c(a) = \sum_{t \text{ used in arc } a} M(t) \times \gamma(t)$, where
  - $M(t)$ is the multiplicity of link type t
  - $\gamma(t)$ is the capacity of link type t

Composite-link design

- Link types = { 1, 2, ..., T }
- Capacities = { c(1), c(2), ..., c(T) } : c(i) < c(i+1)
- Prices / unit length = { p(1), p(2), ..., p(T) }: p(i) < p(i+1)
- Assumptions:
  - $[p(T)/c(T)] < [p(T-1)/c(T-1)] < \cdots < [p(1)/c(1)]$, i.e. price per unit of capacity is smaller for links with greater capacity
  - $c(i) = \alpha \times c(i-1)$, for $\alpha \in N$, $\alpha > 1$, i.e. capacities are multiples of each other by powers of α

Composite-link design

- Link types = { 1, 2, ..., T }
- Capacities = { c(1), c(2), ..., c(T) } : $c(i) < c(i+1)$
- Prices / unit length = { p(1), p(2), ..., p(T) }: $p(i) < p(i+1)$
- Assumptions:
  - $[p(T)/c(T)] < [p(T-1)/c(T-1)] < \cdots < [p(1)/c(1)]$, i.e. price per unit of capacity is smaller for links with greater capacity
  - $c(i) = \alpha \times c(i-1)$, for $\alpha \in N$, $\alpha > 1$, i.e. capacities are multiples of each other by powers of α

Composite-link design

- Link types = { 1, 2, ..., T }
- Capacities = { c(1), c(2), ..., c(T) } : c(i) < c(i+1)
- Prices / unit length = { p(1), p(2), ..., p(T) }: p(i) < p(i+1)
- Assumptions:
  - [p(T)/c(T)] < [p(T−1)/c(T−1)] < ··· < [p(1)/c(1)]: economies of scale
  - $c(i) = \alpha \times c(i-1)$, for $\alpha \in N, \alpha > 1$,
    e.g. c(OC192) = 4 × c(OC48); c(OC48) = 4 × c(OC12); c(OC12) = 4 × c(OC3);

OC3	OC12	OC48	OC192	
155 Mb/s	622 Mb/s	2.5 Gb/s	10 Gb/s	$\alpha = 4$

at&t
Your world. Delivered.

Heuristics for composite-link design

- Designed to minimize overall network cost.
- Heuristics are:
  - Min capacity
  - Min cost
  - Min cost k types
  - Min multiplicities

Heuristics for composite-link design

- Designed to minimize overall network cost.
- Heuristics are:
  - Min capacity
  - Min cost
  - Min cost k types
  - Min multiplicities

Heuristics for composite-link design

- Designed to minimize overall network cost.
- Heuristics are:
  - Min capacity
  - Min cost
  - Min cost k types
  - Min multiplicities

Heuristics for composite-link design

- Designed to minimize overall network cost.
- Heuristics are:
  - Min capacity
  - Min cost
  - Min cost k types
  - Min multiplicities

Heuristics for composite-link design

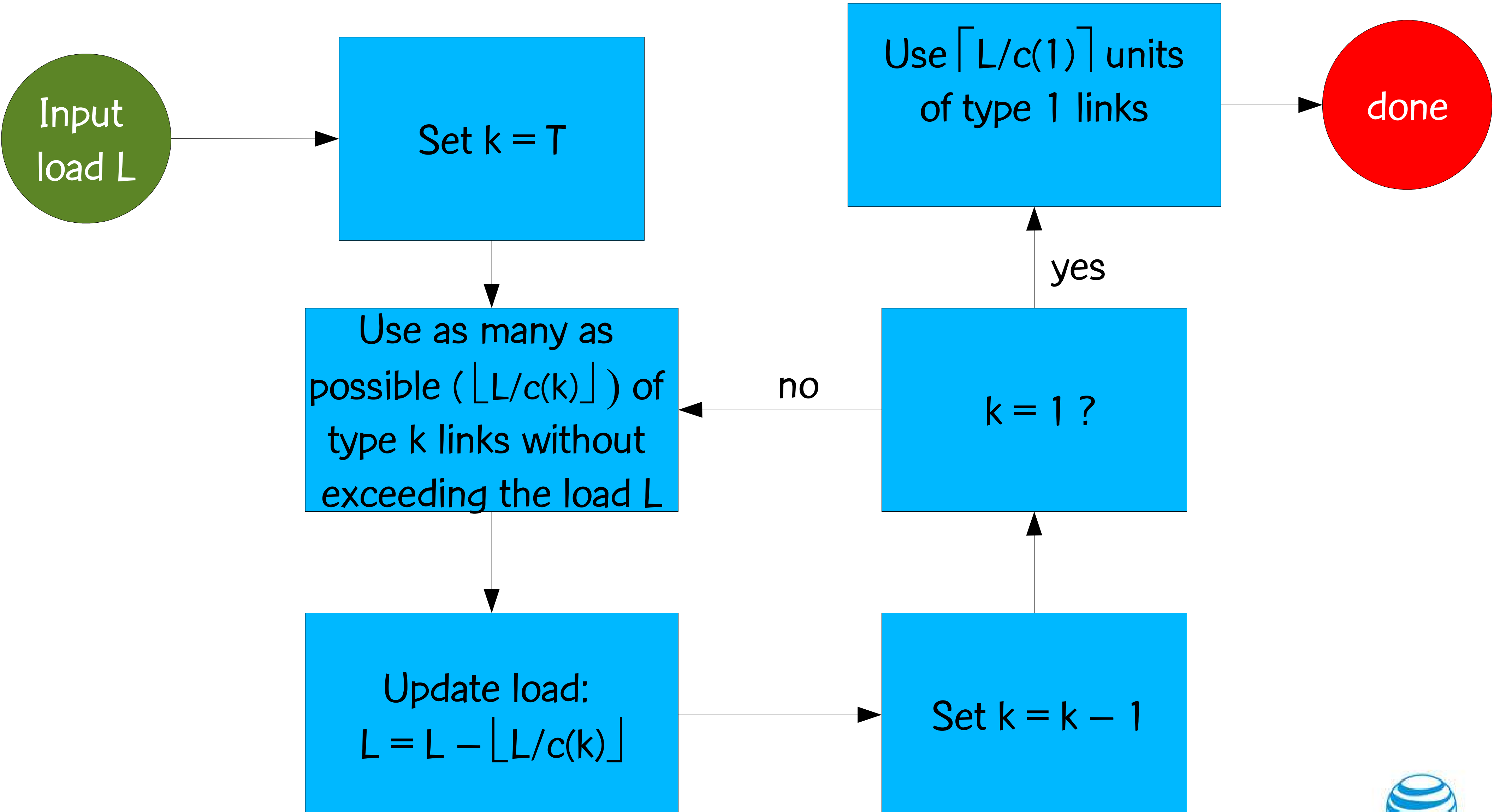
- Designed to minimize overall network cost.
- Heuristics are:
  - Min capacity
  - Min cost
  - Min cost k types
  - Min multiplicities

at&t
Your world. Delivered.

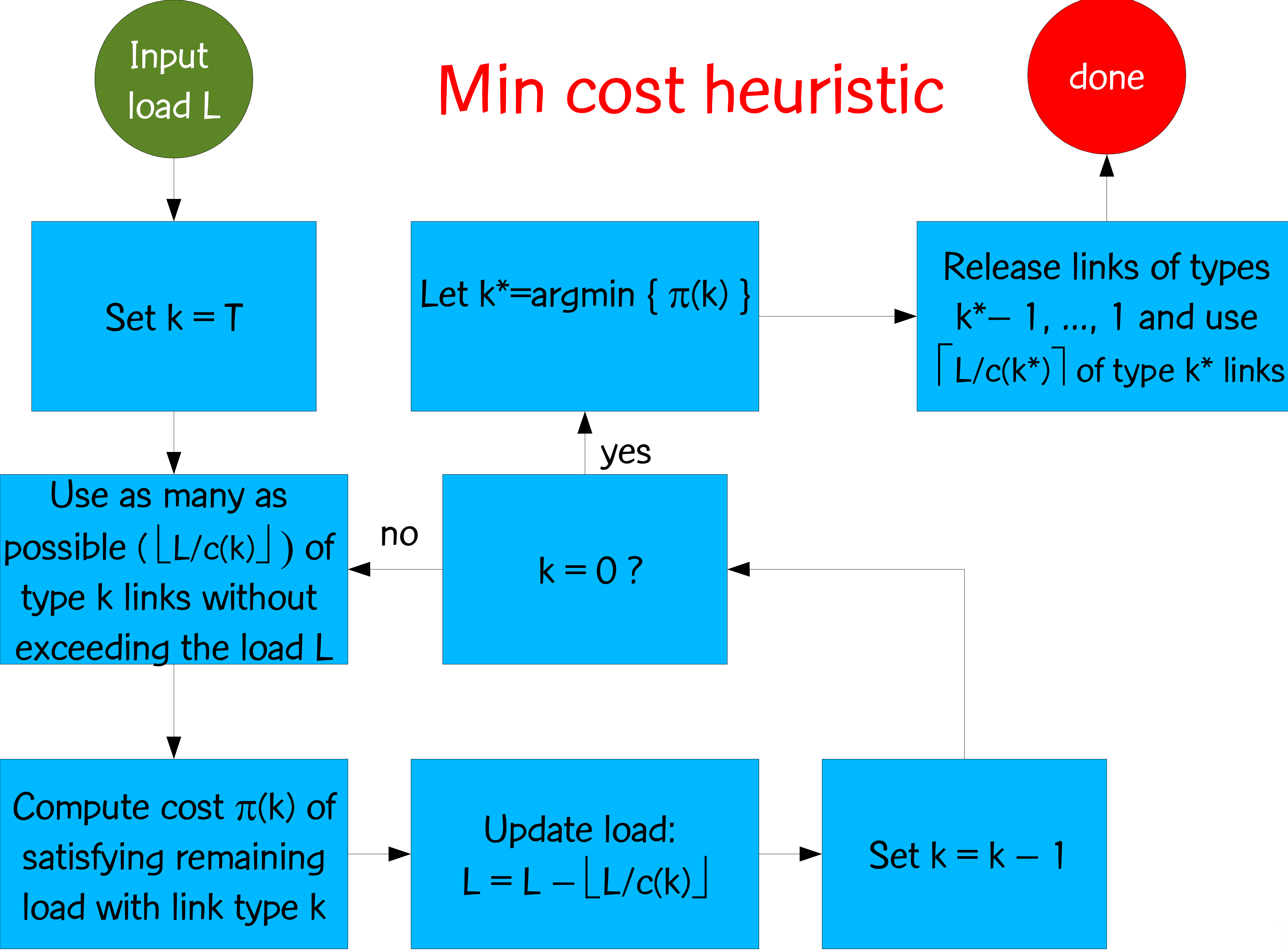
Heuristics for composite-link design

- Designed to minimize overall network cost.
- Heuristics are:
  - Min capacity
  - Min cost
  - Min cost k types
  - Min multiplicities

Min capacity heuristic



Min cost heuristic



$\pi(k)$ is total cost of using links of types T, T– 1, ..., k.

Min cost k types

- Follows same idea as "Min cost" heuristic.
- Can use at most k different link types.
- In some applications, this additional constraint can be imposed.
- Use small values of k, e.g. k=1, k=2

Min cost k types

- Follows same idea as "Min cost" heuristic.
- Can use at most k different link types.
- In some applications, this additional constraint can be imposed.
- Use small values of k, e.g. k=1, k=2

Min cost k types

- Follows same idea as "Min cost" heuristic.
- Can use at most k different link types.
- In some applications, this additional constraint can be imposed.
- Use small values of k, e.g. k=1, k=2

Min cost k types

- Follows same idea as "Min cost" heuristic.
- Can use at most k different link types.
- In some applications, this additional constraint can be imposed.
- Use small values of k, e.g. k=1, k=2

Min cost k=1 type

- Let L be the load on the arc.
- For each link type $k \in \{ 1, 2, ..., T \}$ compute cost $\pi(k)$ of deploying $\lceil L/c(k) \rceil$ units of link type k.
- Let $k^*=\text{argmin} \{ \pi(k): k \in \{ 1, 2, ..., T \} \}$ be the least cost link type.
- Deploy $\lceil L/c(k^*) \rceil$ units of link type k^*.

Min cost k=1 type

- Let L be the load on the arc.
- For each link type $k \in \{ 1, 2, ..., T \}$ compute cost $\pi(k)$ of deploying $\lceil L/c(k) \rceil$ units of link type k.
- Let $k^*=\text{argmin} \{ \pi(k): k \in \{ 1, 2, ..., T \} \}$ be the least cost link type.
- Deploy $\lceil L/c(k^*) \rceil$ units of link type k^*.

Min cost k=1 type

- Let L be the load on the arc.
- For each link type $k \in \{ 1, 2, ..., T \}$ compute cost $\pi(k)$ of deploying $\lceil L/c(k) \rceil$ units of link type k.
- Let $k^* = \text{argmin} \{ \pi(k) : k \in \{ 1, 2, ..., T \} \}$ be the least cost link type.
- Deploy $\lceil L/c(k^*) \rceil$ units of link type k^*.

Min cost k=1 type

- Let L be the load on the arc.
- For each link type $k \in \{ 1, 2, ..., T \}$ compute cost $\pi(k)$ of deploying $\lceil L/c(k) \rceil$ units of link type k.
- Let $k^*=\text{argmin} \{ \pi(k): k \in \{ 1, 2, ..., T \} \}$ be the least cost link type.
- Deploy $\lceil L/c(k^*) \rceil$ units of link type k^*.

Min cost k=2 types

- Let L be the load on the arc.
- For each link type $k \in \{ 1, 2, \ldots, T \}$ compute cost $\pi(k)$ of deploying $\lceil L/c(k) \rceil$ units of link type k.
- For each pair of link types i and j ($j > i$) compute cost $\Pi(i,j)$ of deploying $\lfloor L/c(j) \rfloor$ units of link type j and $\lceil L/c(i) \rceil$ units of link type i.
- Let $k^* = \mathrm{argmin} \{ \pi(k) \}$ and $(i^*,j^*) = \mathrm{argmin} \{ \Pi(i,j) \}$.
- If $\Pi(i^*,j^*) > \pi(k^*)$ deploy $\lceil L/c(k^*) \rceil$ units of link type k^*;
- Else deploy $\lfloor L/c(j^*) \rfloor$ units of link type j^* and $\lceil L/c(i^*) \rceil$ units of link type i^*.

at&t
Your world. Delivered.

Min cost k=2 types

- Let L be the load on the arc.
- For each link type $k \in \{ 1, 2, ..., T \}$ compute cost $\pi(k)$ of deploying $\lceil L/c(k) \rceil$ units of link type k.
- For each pair of link types i and j (j > i) compute cost $\Pi(i,j)$ of deploying $\lfloor L/c(j) \rfloor$ units of link type j and $\lceil L/c(i) \rceil$ units of link type i.
- Let $k^*=\text{argmin} \{ \pi(k) \}$ and $(i^*,j^*) = \text{argmin} \{ \Pi(i,j) \}$.
- If $\Pi(i^*,j^*) > \pi(k^*)$ deploy $\lceil L/c(k^*) \rceil$ units of link type k^*;
- Else deploy $\lfloor L/c(j^*) \rfloor$ units of link type j^* and $\lceil L/c(i^*) \rceil$ units of link type i^*.

at&t
Your world. Delivered.

Min cost k=2 types

- Let L be the load on the arc.
- For each link type $k \in \{ 1, 2, ..., T \}$ compute cost $\pi(k)$ of deploying $\lceil L/c(k) \rceil$ units of link type k.
- For each pair of link types i and j ($j > i$) compute cost $\Pi(i,j)$ of deploying $\lfloor L/c(j) \rfloor$ units of link type j and $\lceil L/c(i) \rceil$ units of link type i.
- Let $k^* = \text{argmin} \{ \pi(k) \}$ and $(i^*,j^*) = \text{argmin} \{ \Pi(i,j) \}$.
- If $\Pi(i^*,j^*) > \pi(k^*)$ deploy $\lceil L/c(k^*) \rceil$ units of link type k^*;
- Else deploy $\lfloor L/c(j^*) \rfloor$ units of link type j^* and $\lceil L/c(i^*) \rceil$ units of link type i^*.

at&t
Your world. Delivered.

Min cost k=2 types

- Let L be the load on the arc.
- For each link type $k \in \{ 1, 2, ..., T \}$ compute cost $\pi(k)$ of deploying $\lceil L/c(k) \rceil$ units of link type k.
- For each pair of link types i and j ($j > i$) compute cost $\Pi(i,j)$ of deploying $\lfloor L/c(j) \rfloor$ units of link type j and $\lceil L/c(i) \rceil$ units of link type i.
- Let $k^* = \text{argmin} \{ \pi(k) \}$ and $(i^*,j^*) = \text{argmin} \{ \Pi(i,j) \}$.
- If $\Pi(i^*,j^*) > \pi(k^*)$ deploy $\lceil L/c(k^*) \rceil$ units of link type k*;
- Else deploy $\lfloor L/c(j^*) \rfloor$ units of link type j* and $\lceil L/c(i^*) \rceil$ units of link type i*.

at&t
Your world. Delivered.

Min cost k=2 types

- Let L be the load on the arc.
- For each link type $k \in \{ 1, 2, ..., T \}$ compute cost $\pi(k)$ of deploying $\lceil L/c(k) \rceil$ units of link type k.
- For each pair of link types i and j ($j > i$) compute cost $\Pi(i,j)$ of deploying $\lfloor L/c(j) \rfloor$ units of link type j and $\lceil L/c(i) \rceil$ units of link type i.
- Let $k^* = \text{argmin} \{ \pi(k) \}$ and $(i^*, j^*) = \text{argmin} \{ \Pi(i,j) \}$.
- If $\Pi(i^*, j^*) > \pi(k^*)$ deploy $\lceil L/c(k^*) \rceil$ units of link type k^*;
- Else deploy $\lfloor L/c(j^*) \rfloor$ units of link type j^* and $\lceil L/c(i^*) \rceil$ units of link type i^*.

Min cost k=2 types

- Let L be the load on the arc.
- For each link type $k \in \{ 1, 2, ..., T \}$ compute cost $\pi(k)$ of deploying $\lceil L/c(k) \rceil$ units of link type k.
- For each pair of link types i and j (j > i) compute cost $\Pi(i,j)$ of deploying $\lfloor L/c(j) \rfloor$ units of link type j and $\lceil L/c(i) \rceil$ units of link type i.
- Let $k^* = \text{argmin} \{ \pi(k) \}$ and $(i^*,j^*) = \text{argmin} \{ \Pi(i,j) \}$.
- If $\Pi(i^*,j^*) > \pi(k^*)$ deploy $\lceil L/c(k^*) \rceil$ units of link type k*;
- Else deploy $\lfloor L/c(j^*) \rfloor$ units of link type j* and $\lceil L/c(i^*) \rceil$ units of link type i*.

Your world. Delivered.

Min multiplicities

- Minimizes number of copies of links used to satisfy the load.
- Multiplicity of link type k is $\lceil L/c(k) \rceil$;
- If $\lceil L/c(T) \rceil > 1$, then deploy $\lceil L/c(T) \rceil$ units of link type T and stop;
- For $k = T - 1, \ldots, 1$ do:
  - If $\lceil L/c(k) \rceil > 1$, then deploy $\lceil L/c(k+1) \rceil$ units of link type k+1 and stop;

Min multiplicities

- Minimizes number of copies of links used to satisfy the load.
- Multiplicity of link type k is $\lceil L/c(k) \rceil$;
- If $\lceil L/c(T) \rceil > 1$, then deploy $\lceil L/c(T) \rceil$ units of link type T and stop;
- For $k = T - 1, \ldots, 1$ do:
  - If $\lceil L/c(k) \rceil > 1$, then deploy $\lceil L/c(k+1) \rceil$ units of link type k+1 and stop;

Min multiplicities

- Minimizes number of copies of links used to satisfy the load.
- Multiplicity of link type k is $\lceil L/c(k) \rceil$;
- If $\lceil L/c(T) \rceil > 1$, then deploy $\lceil L/c(T) \rceil$ units of link type T and stop;
- For k = T – 1, ..., 1 do:
  - If $\lceil L/c(k) \rceil > 1$, then deploy $\lceil L/c(k+1) \rceil$ units of link type k+1 and stop;

Min multiplicities

- Minimizes number of copies of links used to satisfy the load.
- Multiplicity of link type k is $\lceil L/c(k) \rceil$;
- If $\lceil L/c(T) \rceil > 1$, then deploy $\lceil L/c(T) \rceil$ units of link type T and stop;
- For $k = T - 1, \ldots, 1$ do:
  - If $\lceil L/c(k) \rceil > 1$, then deploy $\lceil L/c(k+1) \rceil$ units of link type k+1 and stop;

Observations

- Subject to the assumptions listed earlier, all heuristics (except min cost k > 1 types) can be implemented to take O(T) time to execute per arc.
- Min capacity gives optimal solution for the minimum capacity objective function.
- Min cost gives the optimal solution for the minimum cost objective function.
- Without the assumptions, a knapsack problem must be solved to find min cap and min cost solutions.

Observations

- Subject to the assumptions listed earlier, all heuristics (except min cost k > 1 types) can be implemented to take O(T) time to execute per arc.
- Min capacity gives optimal solution for the minimum capacity objective function.
- Min cost gives the optimal solution for the minimum cost objective function.
- Without the assumptions, a knapsack problem must be solved to find min cap and min cost solutions.

Observations

- Subject to the assumptions listed earlier, all heuristics (except min cost $k > 1$ types) can be implemented to take $O(T)$ time to execute per arc.
- Min capacity gives optimal solution for the minimum capacity objective function.
- Min cost gives the optimal solution for the minimum cost objective function.
- Without the assumptions, a knapsack problem must be solved to find min cap and min cost solutions.

Observations

- Subject to the assumptions listed earlier, all heuristics (except min cost $k > 1$ types) can be implemented to take $O(T)$ time to execute per arc.
- Min capacity gives optimal solution for the minimum capacity objective function.
- Min cost gives the optimal solution for the minimum cost objective function.
- Without the assumptions, a knapsack problem must be solved to find min cap and min cost solutions.

Experimental results

- Use a "real" network with 54 routers and 278 arcs.
- Three link types used: { 1, 2, 3 }
- $c(2) = 4\ c(1)$; $c(3) = 16\ c(1)$
- $p(2)/c(2) = 0.95\ p(1)/c(1)$; $p(3)/c(3) = 0.90\ p(1)/c(1)$
- All four heuristics tested. Min cost k types was tested for k=1 and k=2.
- GA was run 100, 200, 300, ..., 1000 generations and costs were recorded for each heuristic.

Experimental results

- Use a “real” network with 54 routers and 278 arcs.
- Three link types used: { 1, 2, 3 }
- $c(2) = 4\ c(1)$; $c(3) = 16\ c(1)$
- $p(2)/c(2) = 0.95\ p(1)/c(1)$; $p(3)/c(3) = 0.90\ p(1)/c(1)$
- All four heuristics tested. Min cost k types was tested for k=1 and k=2.
- GA was run 100, 200, 300, ..., 1000 generations and costs were recorded for each heuristic.

Experimental results

- Use a "real" network with 54 routers and 278 arcs.
- Three link types used: { 1, 2, 3 }
- $c(2) = 4\ c(1)$; $c(3) = 16\ c(1)$
- $p(2)/c(2) = 0.95\ p(1)/c(1)$; $p(3)/c(3) = 0.90\ p(1)/c(1)$
- All four heuristics tested. Min cost k types was tested for k=1 and k=2.
- GA was run 100, 200, 300, ..., 1000 generations and costs were recorded for each heuristic.

Experimental results

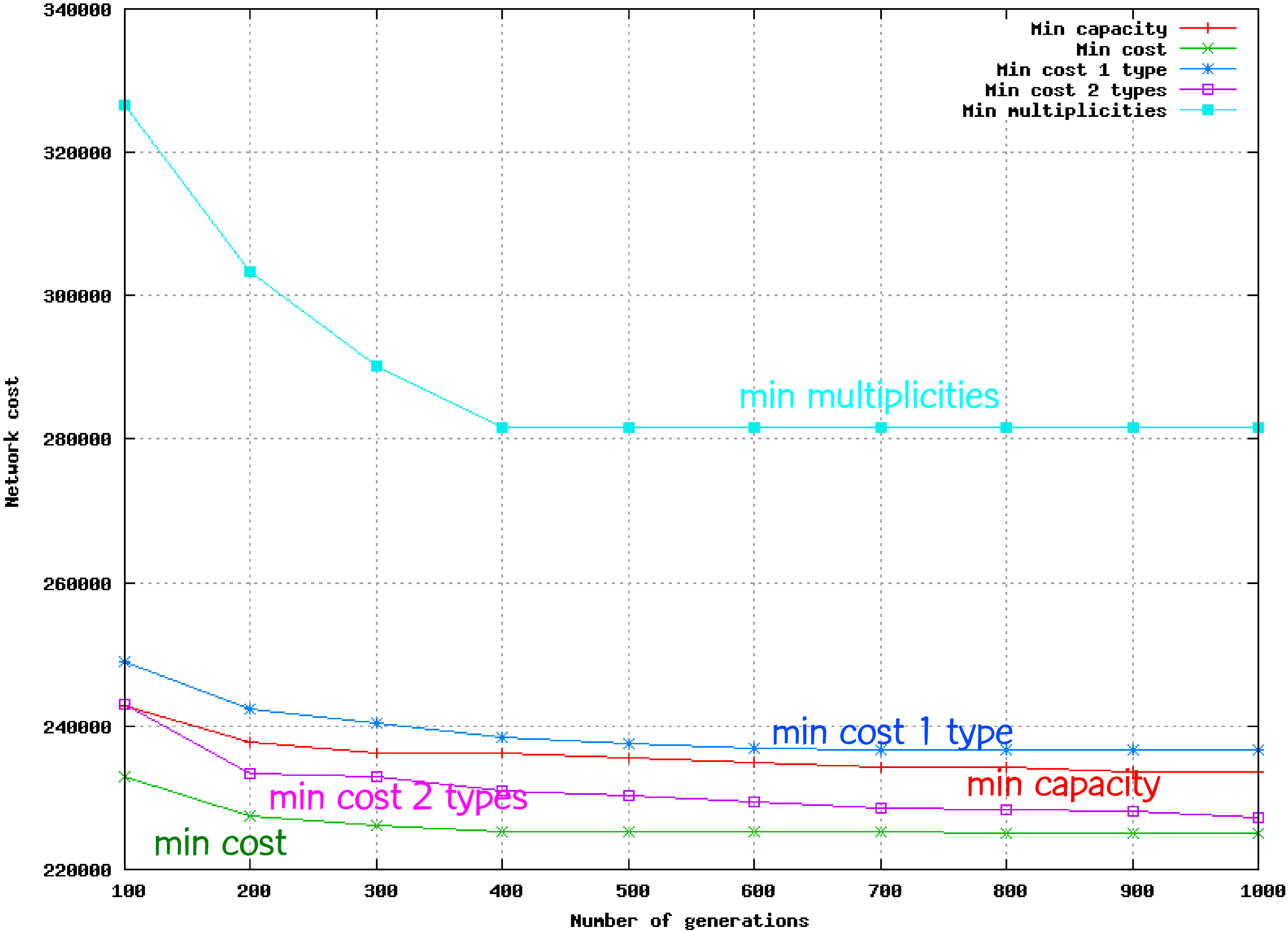
- Use a "real" network with 54 routers and 278 arcs.
- Three link types used: { 1, 2, 3 }
- $c(2) = 4\ c(1)$; $c(3) = 16\ c(1)$
- $p(2)/c(2) = 0.95\ p(1)/c(1)$; $p(3)/c(3) = 0.90\ p(1)/c(1)$
- All four heuristics tested. Min cost k types was tested for k=1 and k=2.
- GA was run 100, 200, 300, ..., 1000 generations and costs were recorded for each heuristic.

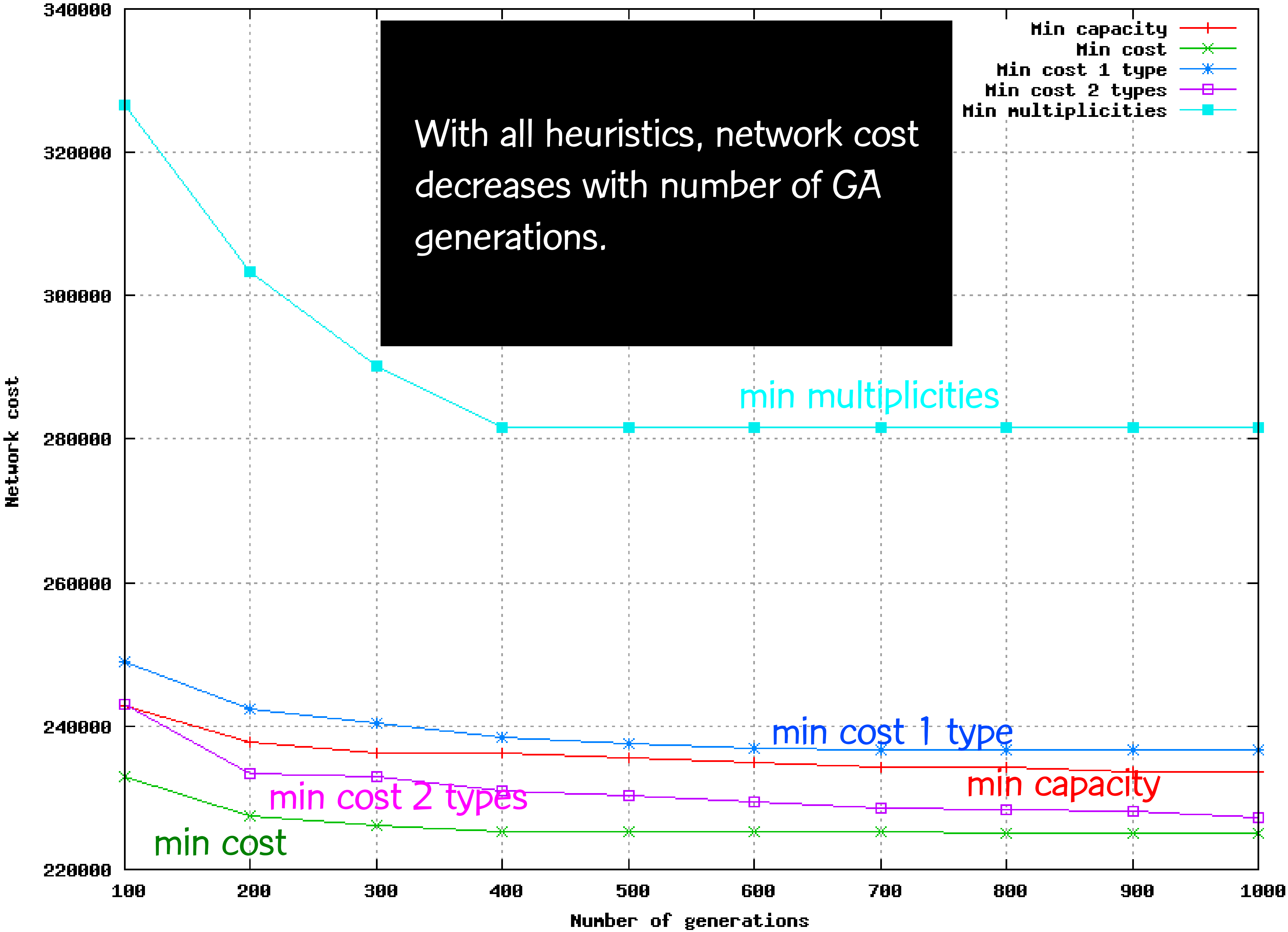
Experimental results

- Use a "real" network with 54 routers and 278 arcs.
- Three link types used: { 1, 2, 3 }
- $c(2) = 4\ c(1)$; $c(3) = 16\ c(1)$
- $p(2)/c(2) = 0.95\ p(1)/c(1)$; $p(3)/c(3) = 0.90\ p(1)/c(1)$
- All four heuristics tested. Min cost k types was tested for k=1 and k=2.
- GA was run 100, 200, 300, ..., 1000 generations and costs were recorded for each heuristic.

Experimental results

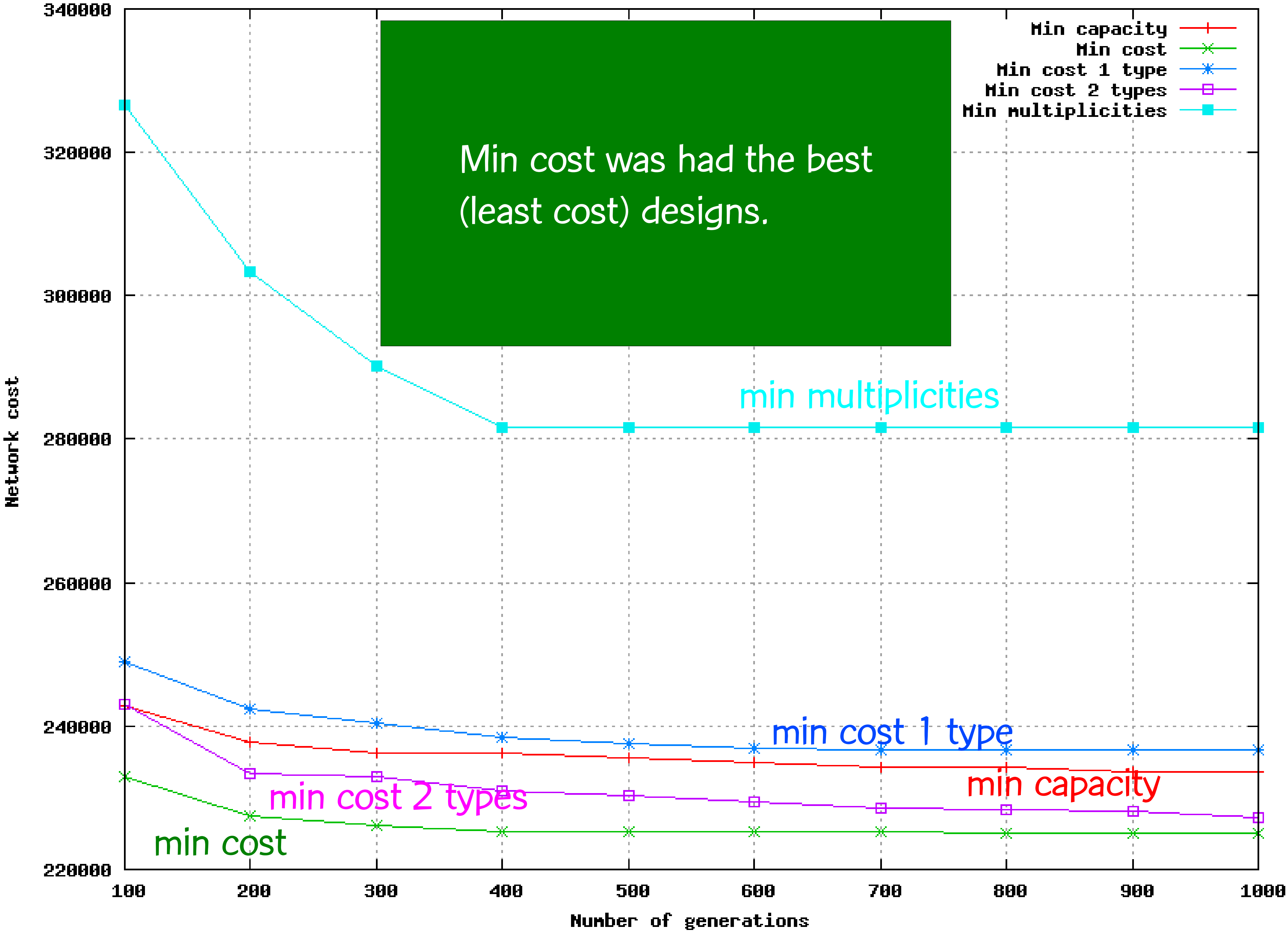
- Use a "real" network with 54 routers and 278 arcs.
- Three link types used: { 1, 2, 3 }
- $c(2) = 4\,c(1)$; $c(3) = 16\,c(1)$
- $p(2)/c(2) = 0.95\,p(1)/c(1)$; $p(3)/c(3) = 0.90\,p(1)/c(1)$
- All four heuristics tested. Min cost k types was tested for k=1 and k=2.
- GA was run 100, 200, 300, ..., 1000 generations and costs were recorded for each heuristic.

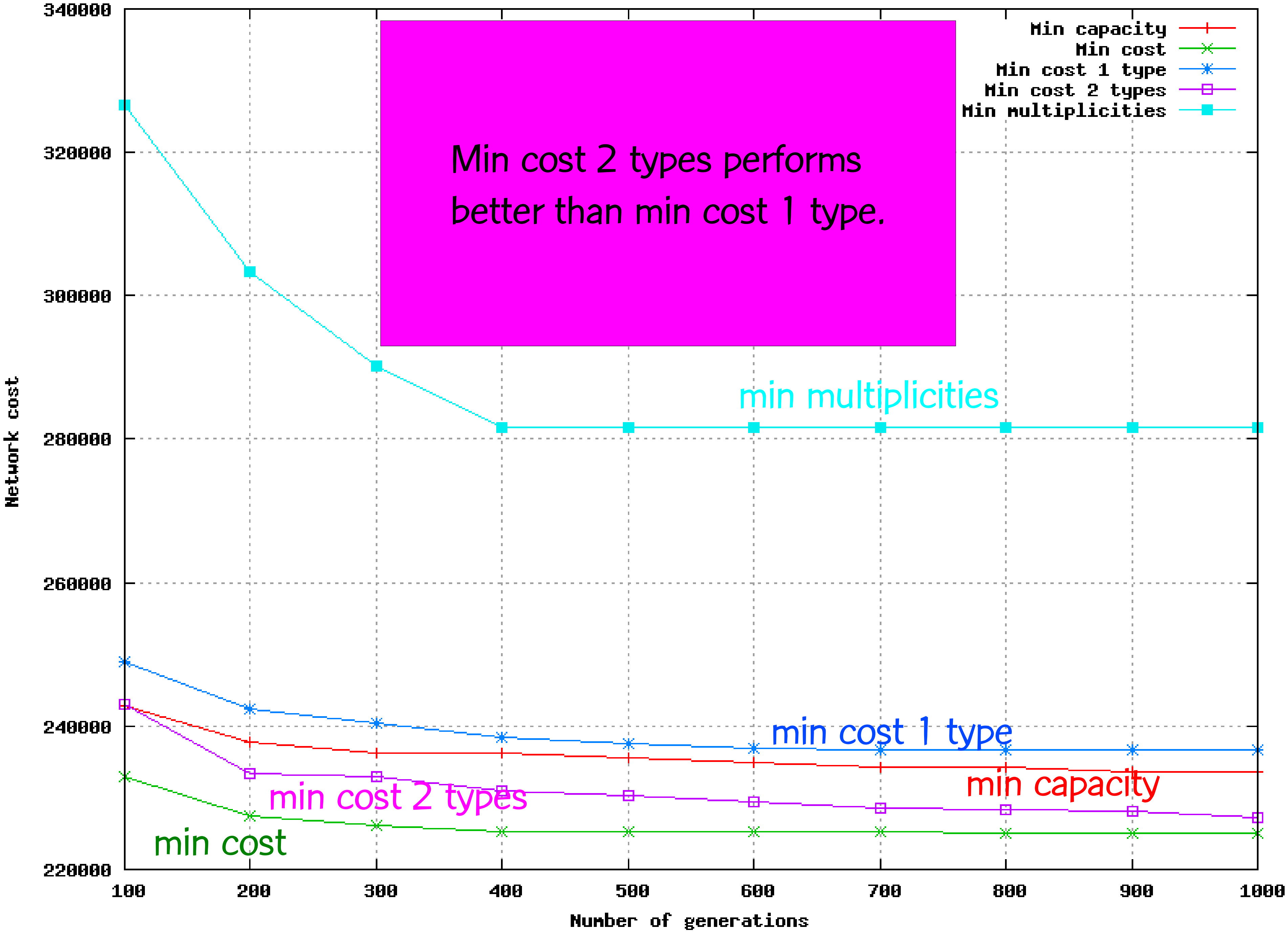

Min capacity
Min cost
Min cost 1 type
Min cost 2 types
Min multiplicities
min multiplicities
min cost 1 type
min capacity
min cost 2 types
min cost
Network cost
340000
320000
300000
280000
260000
240000
220000
100
200
300
400
500
600
700
800
900
1000
Number of generations

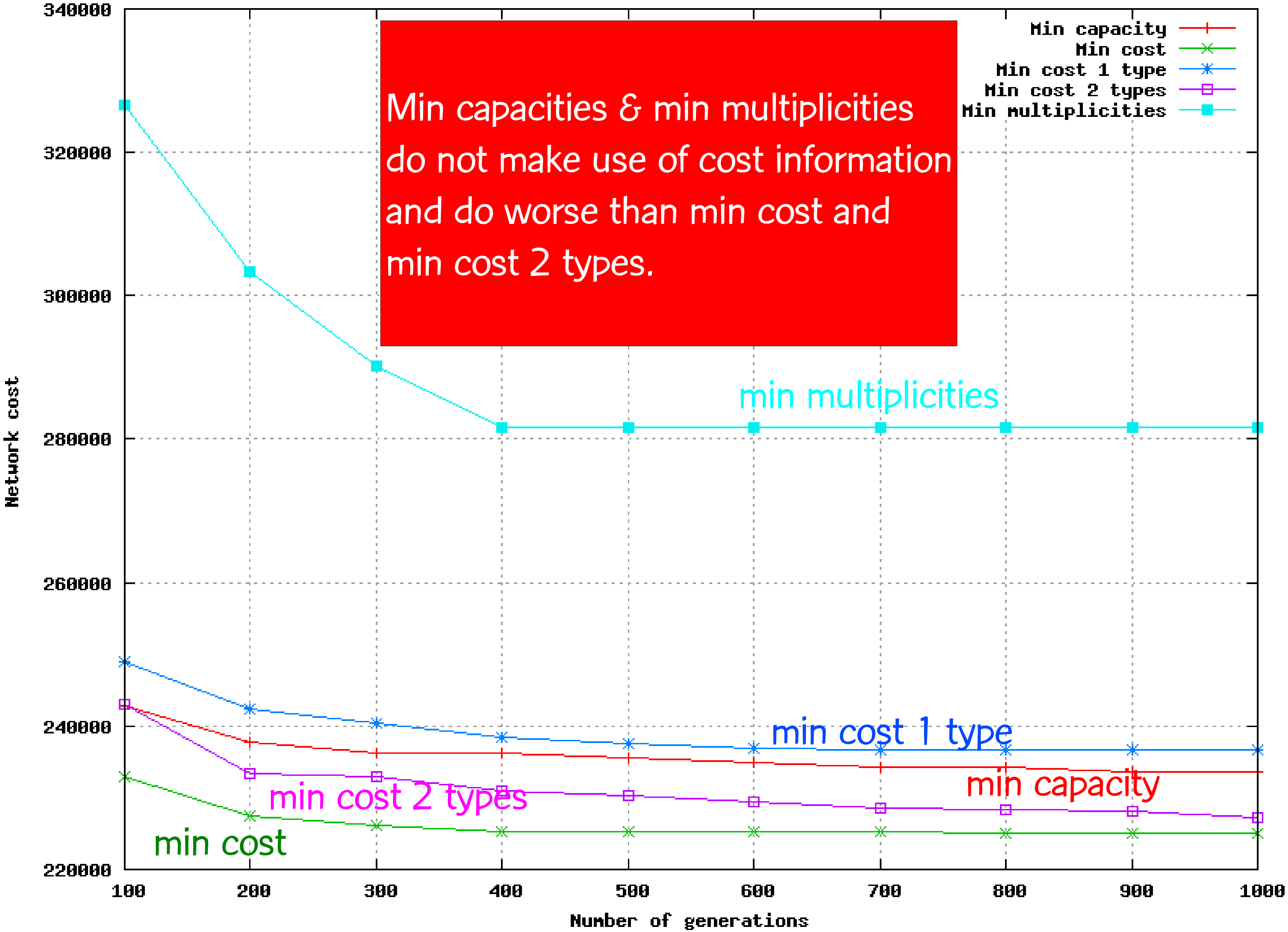


at&t
Your world. Delivered.

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




Min cost 2 types performs better than min cost 1 type.
Min capacity
Min cost
Min cost 1 type
Min cost 2 types
Min multiplicities
min multiplicities
min cost 1 type
min capacity
min cost 2 types
min cost
Network cost
340000
320000
300000
280000
260000
240000
220000
100
200
300
400
500
600
700
800
900
1000
Number of generations


Min capacities & min multiplicities do not make use of cost information and do worse than min cost and min cost 2 types.
Min capacity
Min cost
Min cost 1 type
Min cost 2 types
Min multiplicities
min multiplicities
min cost 1 type
min capacity
min cost 2 types
min cost
Network cost
340000
320000
300000
280000
260000
240000
220000
100
200
300
400
500
600
700
800
900
1000
Number of generations
at&t
Your world. Delivered.

Concluding remarks

- We have extended our survivable IP network design tool to handle composite links.
- Min cost heuristic runs fast and finds best-quality solutions.
- In this talk, traffic splitting was not implemented for the composite link case, as was done in Buriol, Resende, and Thorup (2006).
- We have recently added a traffic splitting option to our tool.

at&t
Your world. Delivered.

Concluding remarks

- We have extended our survivable IP network design tool to handle composite links.
- Min cost heuristic runs fast and finds best-quality solutions.
- In this talk, traffic splitting was not implemented for the composite link case, as was done in Buriol, Resende, and Thorup (2006).
- We have recently added a traffic splitting option to our tool.

Concluding remarks

- We have extended our survivable IP network design tool to handle composite links.
- Min cost heuristic runs fast and finds best-quality solutions.
- In this talk, traffic splitting was not implemented for the composite link case, as was done in Buriol, Resende, and Thorup (2006).
- We have recently added a traffic splitting option to our tool.

Concluding remarks

- We have extended our survivable IP network design tool to handle composite links.
- Min cost heuristic runs fast and finds best-quality solutions.
- In this talk, traffic splitting was not implemented for the composite link case, as was done in Buriol, Resende, and Thorup (2006).
- We have recently added a traffic splitting option to our tool.

My coauthors

Diogo V. Andrade
Rutgers University

Mikkel Thorup
AT&T Labs Research

Luciana S. Buriol
Federal University of
Rio Grande do Sul

at&t
Your world. Delivered.

The End