A GRASP for PBX telephone migration scheduling

Talk given at
Eighth INFORMS Telecommunications Conference
Dallas, Texas
March 31, 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 Diogo Andrade

Summary of talk

- Batch scheduling of multi-grouped units
- GRASP for batch scheduling of multi-grouped units
- PBX telephone migration scheduling
- Concluding remarks

Summary of talk

- Batch scheduling of multi-grouped units
- GRASP for batch scheduling of multi-grouped units
- PBX telephone migration scheduling
- Concluding remarks

at&t
Your world. Delivered.

Summary of talk

- Batch scheduling of multi-grouped units
- GRASP for batch scheduling of multi-grouped units
- PBX telephone migration scheduling
- Concluding remarks

Summary of talk

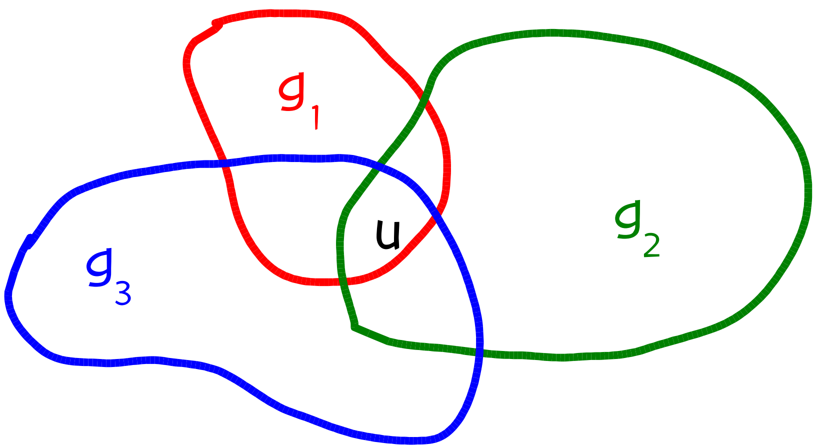
- Batch scheduling of multi-grouped units
- GRASP for batch scheduling of multi-grouped units
- PBX telephone migration scheduling
- Concluding remarks

Batch scheduling of multi-grouped units

- Consider a system with
  - A set U of N units
  - A set H of M groups of units

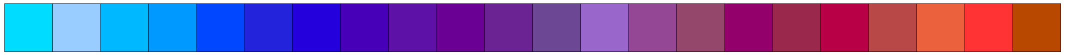
Batch scheduling of multi-grouped units

- 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

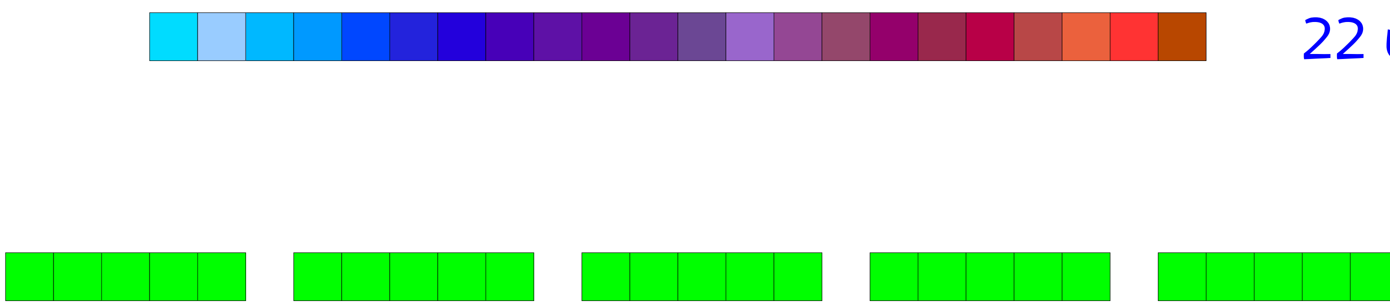
- Given T time periods on which to schedule units.
- No more than C units can be assigned to a single time period.



22 units

Batch scheduling of multi-grouped units

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



22 units

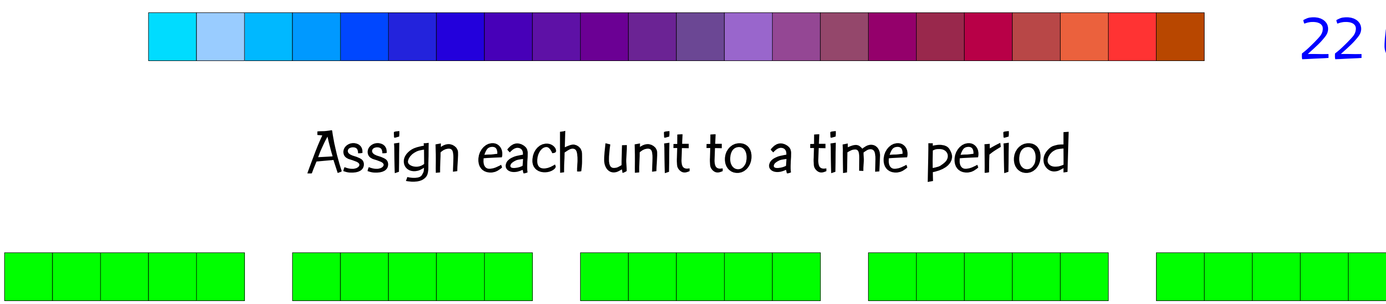
T=5 time periods

C=5 units/period

at&t
Your world. Delivered.

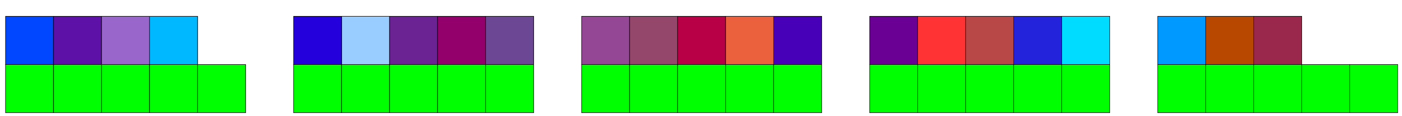
Batch scheduling of multi-grouped units

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



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.



T=5 time periods

C=5 units/period

Batch scheduling of multi-grouped units

- Objective: Schedule two units sharing same group as close together in time 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 in time 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 in time 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

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

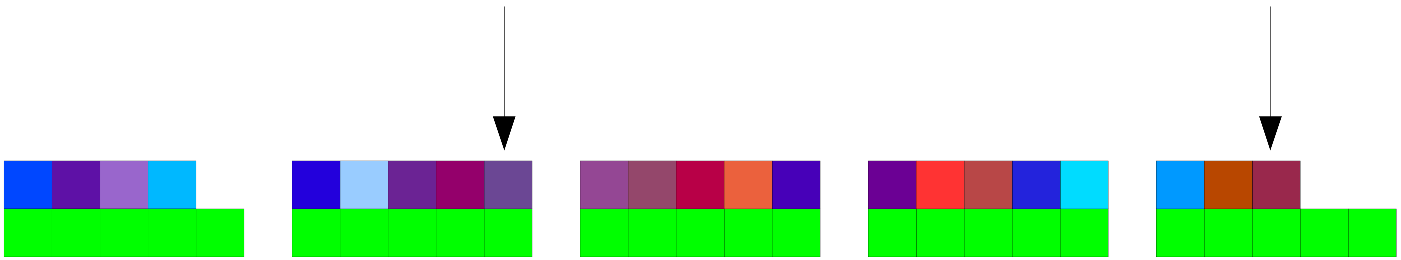
is incurred.

Batch scheduling of multi-grouped units

12

17

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



T=5 time periods

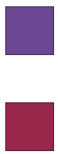
C=5 units/period

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

Batch scheduling of multi-grouped units

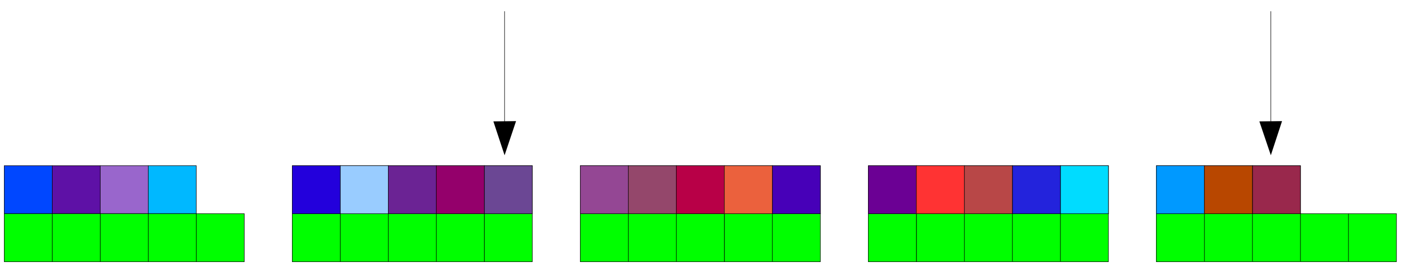
12

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.



Batch scheduling of multi-grouped units

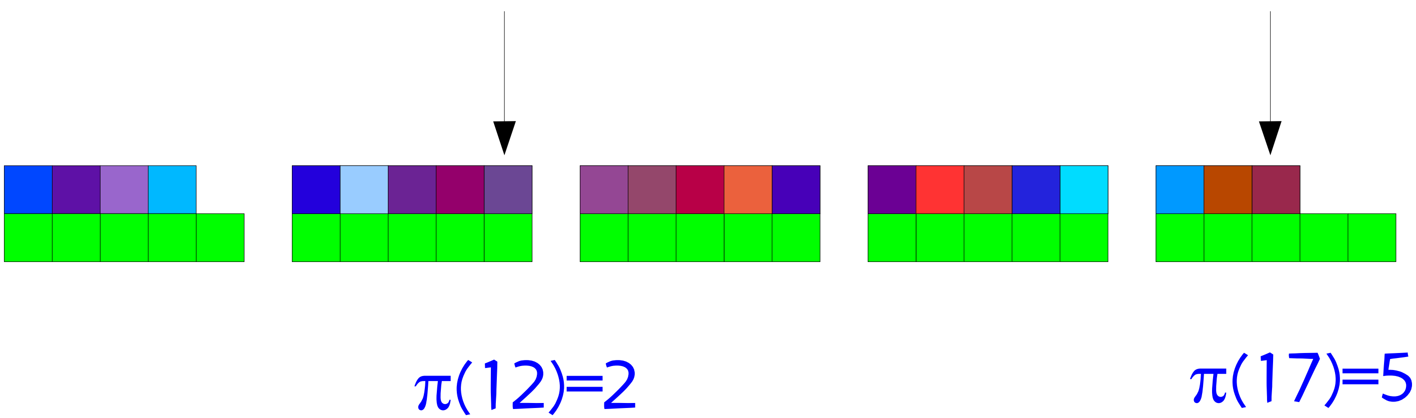
12

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



T=5 time periods

C=5 units/period

Batch scheduling of multi-grouped units

12

17

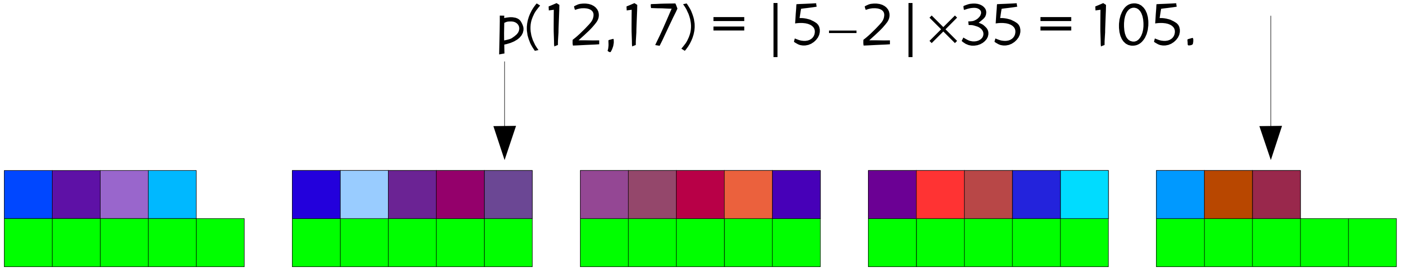
Units 12 and 17 share groups 2, 4, and 8.

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

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



T=5 time periods

C=5 units/period

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

Batch scheduling of multi-grouped units

- Given:
  - T time periods,
  - a limit C on the number of units that can be assigned to a single period,
  - a set of units U to be assigned to periods,
  - a set of groups H that units can share,
  - for each pair of units $u,v \in U$, a subset of groups $G(u,v) \subseteq H$ shared by the pair,
  - a per period penalty $w(u,v,g)$ associated with assigning group g pair u and v to different periods.

at&t
Your world. Delivered.

Batch scheduling of multi-grouped units

- Given:
  - T time periods,
  - a limit C on the number of units that can be assigned to a single period,
  - a set of units U to be assigned to periods,
  - a set of groups H that units can share,
  - for each pair of units $u,v \in U$, a subset of groups $G(u,v) \subseteq H$ shared by the pair,
  - a per period penalty $w(u,v,g)$ associated with assigning group g pair u and v to different periods.

Batch scheduling of multi-grouped units

- Given:
  - T time periods,
  - a limit C on the number of units that can be assigned to a single period,
  - a set of units U to be assigned to periods,
  - a set of groups H that units can share,
  - for each pair of units $u,v \in U$, a subset of groups $G(u,v) \subseteq H$ shared by the pair,
  - a per period penalty $w(u,v,g)$ associated with assigning group g pair u and v to different periods.

Batch scheduling of multi-grouped units

- Given:
  - T time periods,
  - a limit C on the number of units that can be assigned to a single period,
  - a set of units U to be assigned to periods,
  - a set of groups H that units can share,
  - for each pair of units $u,v \in U$, a subset of groups $G(u,v) \subseteq H$ shared by the pair,
  - a per period penalty $w(u,v,g)$ associated with assigning group g pair u and v to different periods.

Batch scheduling of multi-grouped units

- Given:
  - T time periods,
  - a limit C on the number of units that can be assigned to a single period,
  - a set of units U to be assigned to periods,
  - a set of groups H that units can share,
  - for each pair of units $u,v \in U$, a subset of groups $G(u,v) \subseteq H$ shared by the pair,
  - a per period penalty w(u,v,g) associated with assigning group g pair u and v to different periods.

at&t

Your world. Delivered.

Batch scheduling of multi-grouped units

- Given:
  - T time periods,
  - a limit C on the number of units that can be assigned to a single period,
  - a set of units U to be assigned to periods,
  - a set of groups H that units can share,
  - for each pair of units $u,v \in U$, a subset of groups $G(u,v) \subseteq H$ shared by the pair,
  - a per period penalty $w(u,v,g)$ associated with assigning group g pair u and v to different periods.

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.

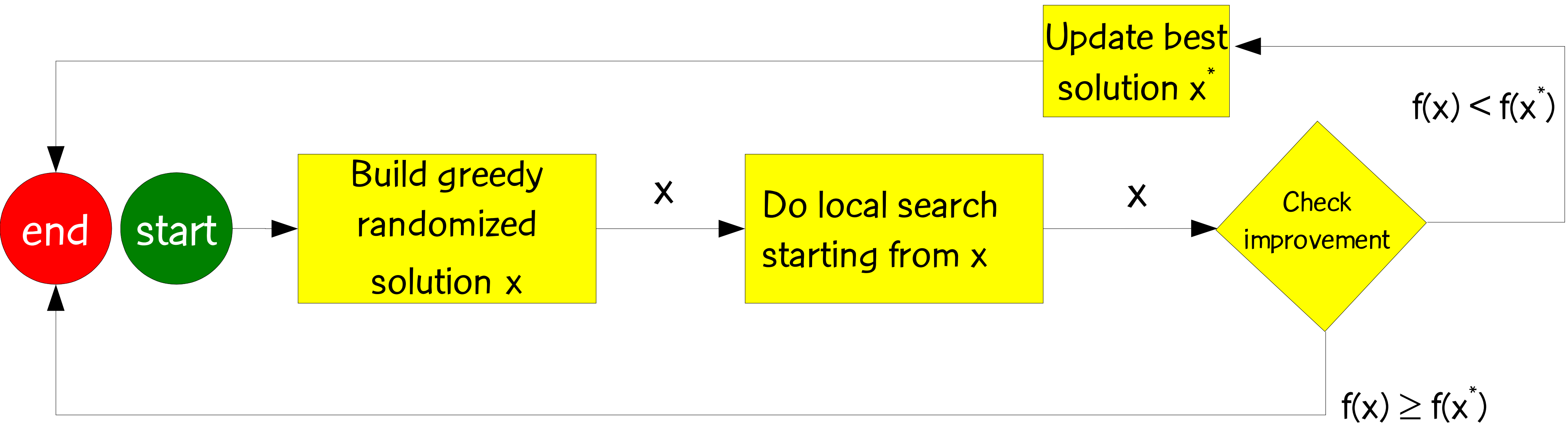
- Problem is NP-hard. It generalizes the minimum linear arrangement problem: Given a graph G(V,E), find $\pi: V \rightarrow \{1,\ldots,|V|\}$ that

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

at&t
Your world. Delivered.

GRASP for batch scheduling of multi-grouped units

- GRASP [Feo & Resende, 1989, 1995] is a multi-start metaheuristic for combinatorial optimization.
- Each GRASP iteration consists of:



- The best solution found, over all iterations, is returned.

GRASP construction

- Construction sequences units and assigns them evenly to time periods.
- Let $\sigma(u)$ be the position of unit u in the sequence.
- Construction seeks to optimize approximate objective function $\sum_{\substack{u,v \in U \times U \\ (u>v)}} |\sigma(u) - \sigma(v)| \times \sum_{g \in G(u,v)} w(u,v,g)$
- Greedy algorithm: next unit k in sequence minimizes $\sum_{v \in \text{unselected}} \sum_{g \in G(k,v)} w(k,v,g) - \sum_{u \in \text{selected}} \sum_{g \in G(u,k)} w(u,k,g)$
- Greedy algorithm is randomized.

Your world. Delivered.

GRASP construction

- Construction sequences units and assigns them evenly to time periods.
- Let $\sigma(u)$ be the position of unit u in the sequence.
- Construction seeks to optimize approximate objective function $\sum_{\substack{u,v \in U \times U \\ (u>v)}} |\sigma(u) - \sigma(v)| \times \sum_{g \in G(u,v)} w(u,v,g)$
- Greedy algorithm: next unit k in sequence minimizes $\sum_{v \in \text{unselected}} \sum_{g \in G(k,v)} w(k,v,g) - \sum_{u \in \text{selected}} \sum_{g \in G(u,k)} w(u,k,g)$
- Greedy algorithm is randomized.

GRASP construction

- Construction sequences units and assigns them evenly to time periods.
- Let $\sigma(u)$ be the position of unit u in the sequence.
- Construction seeks to optimize approximate objective function $\sum_{\substack{u,v \in U \times U \\ (u>v)}} |\sigma(u) - \sigma(v)| \times \sum_{g \in G(u,v)} w(u,v,g)$
- Greedy algorithm: next unit k in sequence minimizes $\sum_{v \in \text{unselected}} \sum_{g \in G(k,v)} w(k,v,g) - \sum_{u \in \text{selected}} \sum_{g \in G(u,k)} w(u,k,g)$
- Greedy algorithm is randomized.

GRASP construction

- Construction sequences units and assigns them evenly to time periods.
- Let $\sigma(u)$ be the position of unit u in the sequence.
- Construction seeks to optimize approximate objective function $\sum_{\substack{u,v \in U \times U \\ (u>v)}} |\sigma(u) - \sigma(v)| \times \sum_{g \in G(u,v)} w(u,v,g)$
- Greedy algorithm: next unit k in sequence minimizes $\sum_{v \in \text{unselected}} \sum_{g \in G(k,v)} w(k,v,g) - \sum_{u \in \text{selected}} \sum_{g \in G(u,k)} w(u,k,g)$
- Greedy algorithm is randomized.

GRASP construction

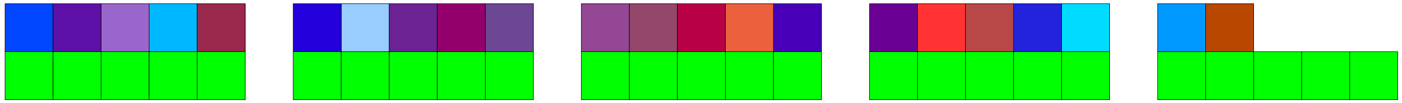
- Construction sequences units and assigns them evenly to time periods.
- Let $\sigma(u)$ be the position of unit u in the sequence.
- Construction seeks to optimize approximate objective function $\sum_{\substack{u,v \in U \times U \\ (u>v)}} |\sigma(u) - \sigma(v)| \times \sum_{g \in G(u,v)} w(u,v,g)$
- Greedy algorithm: next unit k in sequence minimizes $\sum_{v \in \text{unselected}} \sum_{g \in G(k,v)} w(k,v,g) - \sum_{u \in \text{selected}} \sum_{g \in G(u,k)} w(u,k,g)$
- Greedy algorithm is randomized.

at&t

Your world. Delivered.

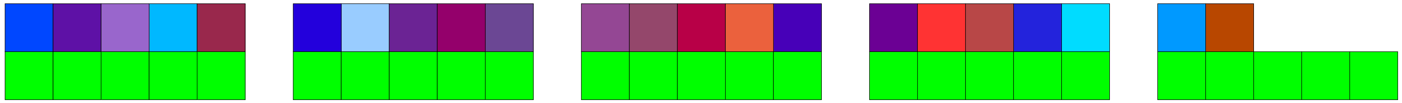
GRASP local search phase

- Input: Assignment of units to periods:



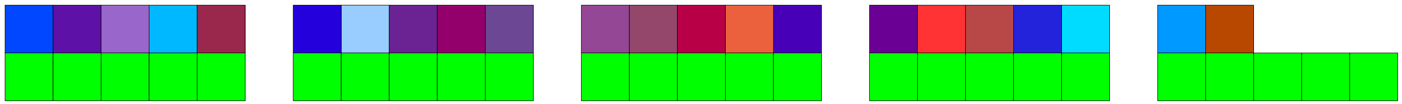
GRASP local search phase

- Local search: Examine neighborhood of current solution. If better solution found, make it current solution.



GRASP local search phase

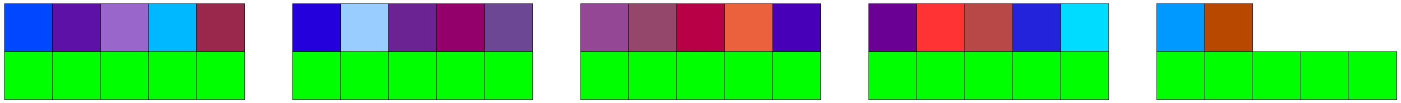
- Three neighborhoods: Swap units, move unit, swap periods.



GRASP local search phase

- Swap units neighborhood: Swaps places of two units assigned to distinct periods.

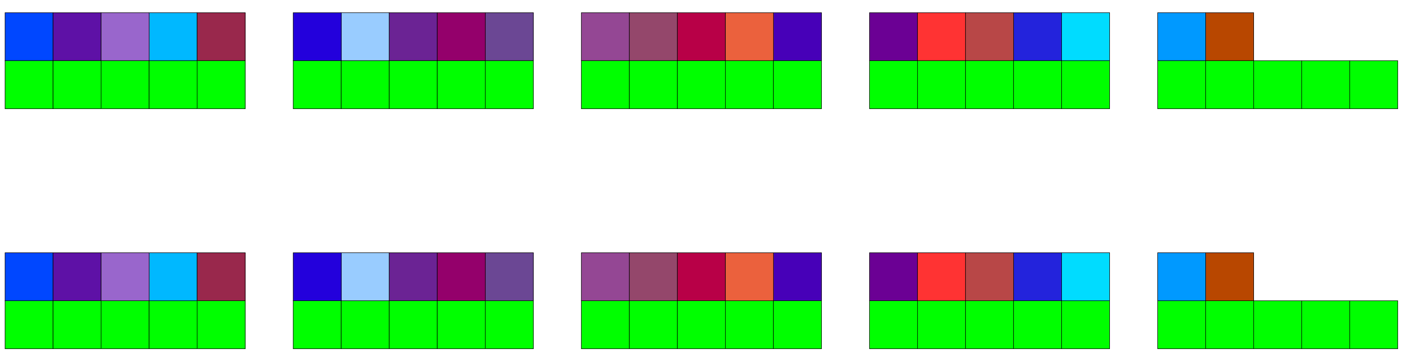
solution



GRASP local search phase

- Swap units neighborhood: Swaps places of two units assigned to distinct periods.

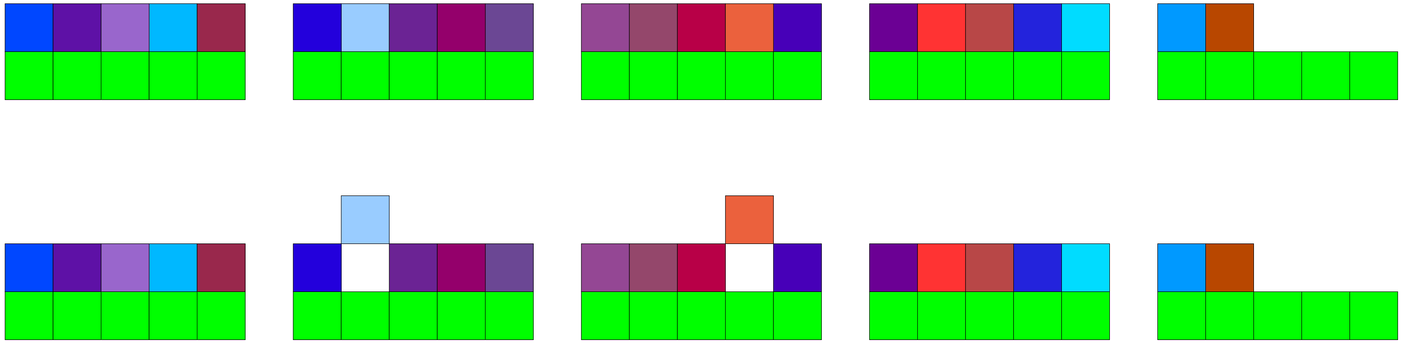
solution



GRASP local search phase

- Swap units neighborhood: Swaps places of two units assigned to distinct periods.

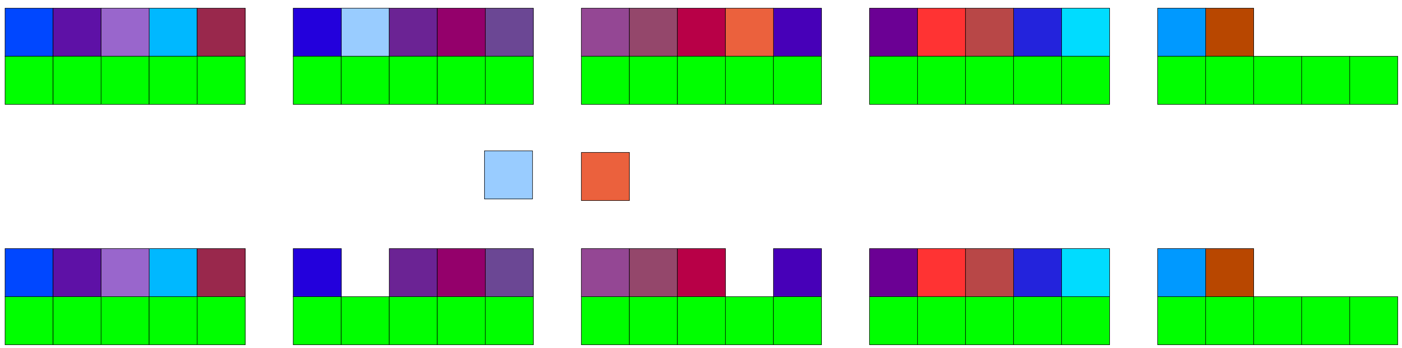
solution



GRASP local search phase

- Swap units neighborhood: Swaps places of two units assigned to distinct periods.

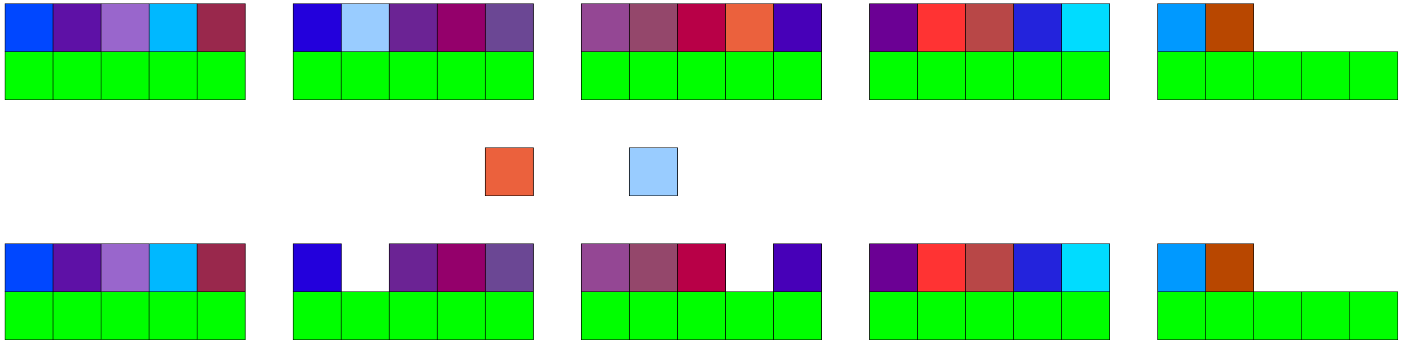
solution



GRASP local search phase

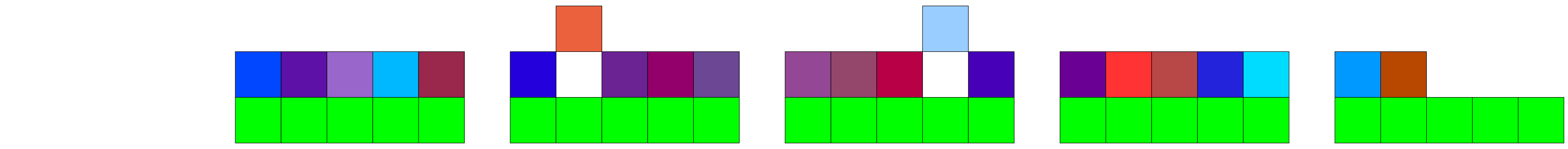
- Swap units neighborhood: Swaps places of two units assigned to distinct periods.

solution



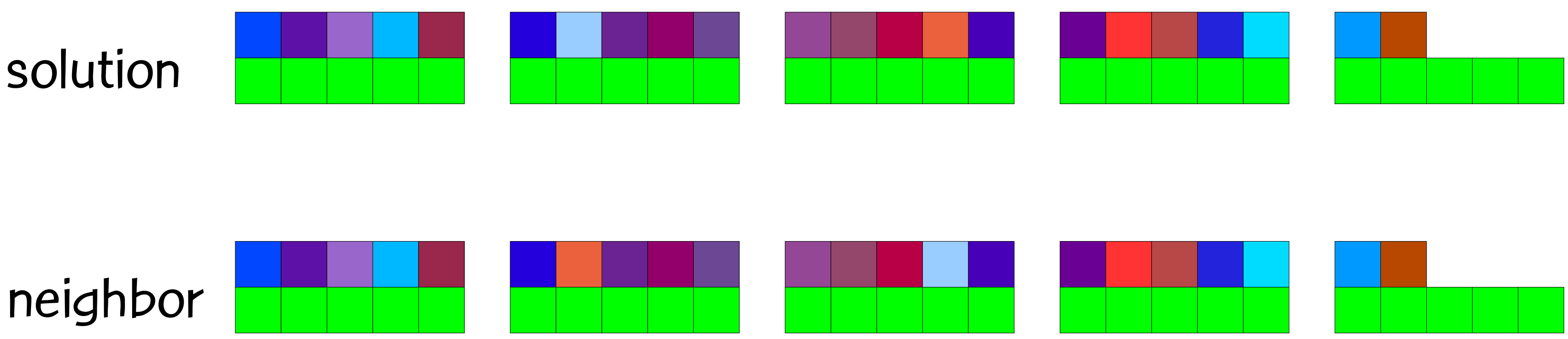
GRASP local search phase

- Swap units neighborhood: Swaps places of two units assigned to distinct periods.



GRASP local search phase

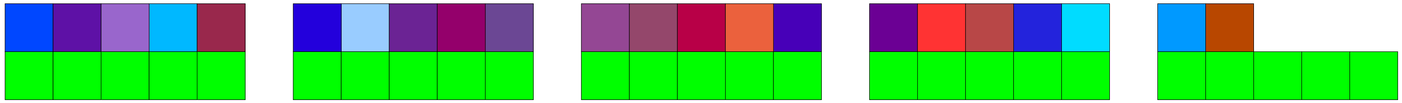
- Swap units neighborhood: Swaps places of two units assigned to distinct periods.



GRASP local search phase

- Move unit neighborhood: Moves unit from current period to available period.

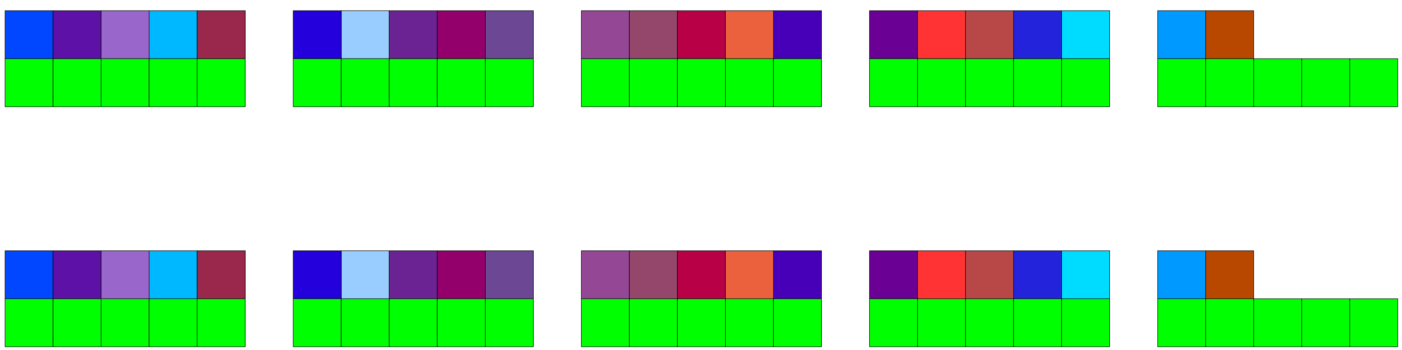
solution



GRASP local search phase

- Move unit neighborhood: Moves unit from current period to available period.

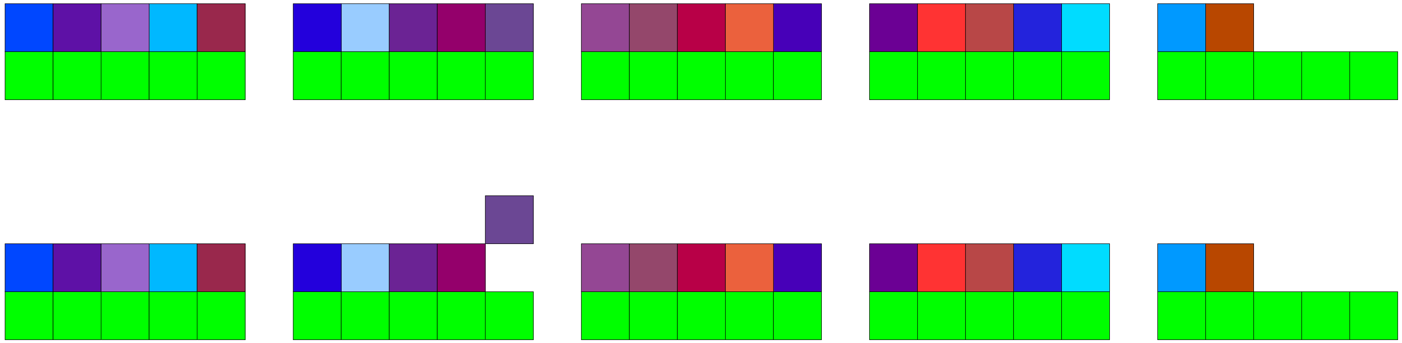
solution



GRASP local search phase

- Move unit neighborhood: Moves unit from current period to available period.

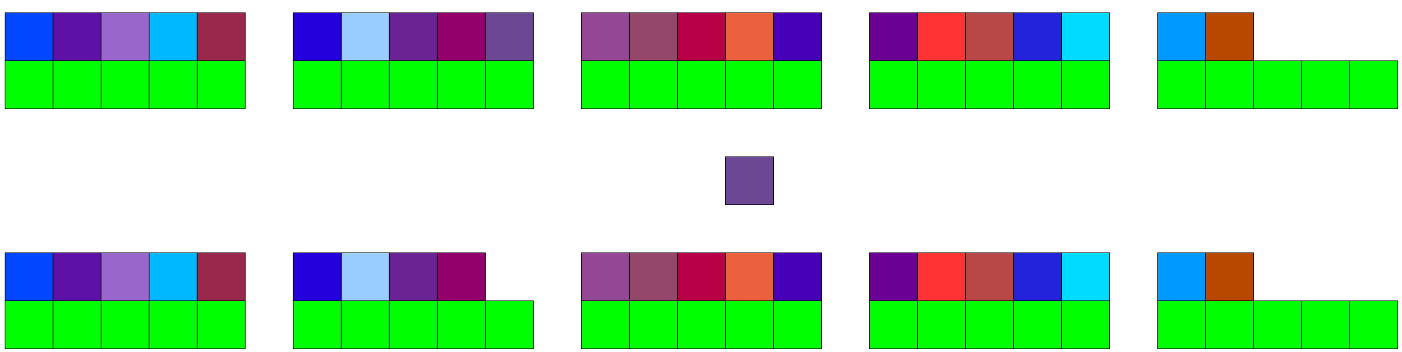
solution



GRASP local search phase

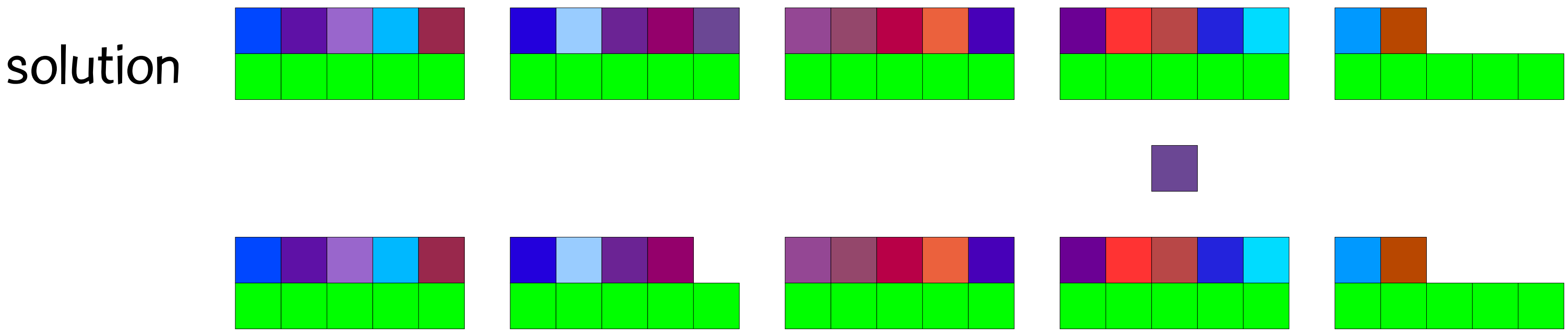
- Move unit neighborhood: Moves unit from current period to available period.

solution



GRASP local search phase

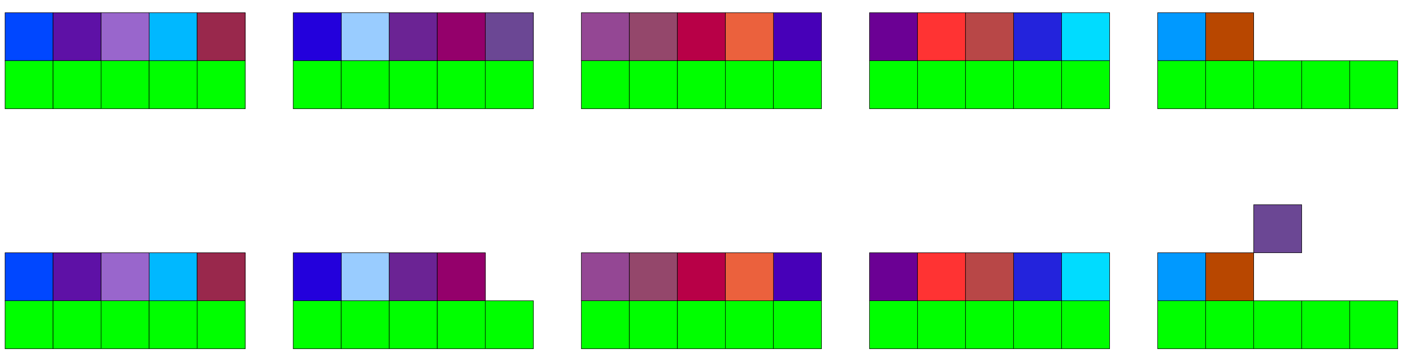
- Move unit neighborhood: Moves unit from current period to available period.



GRASP local search phase

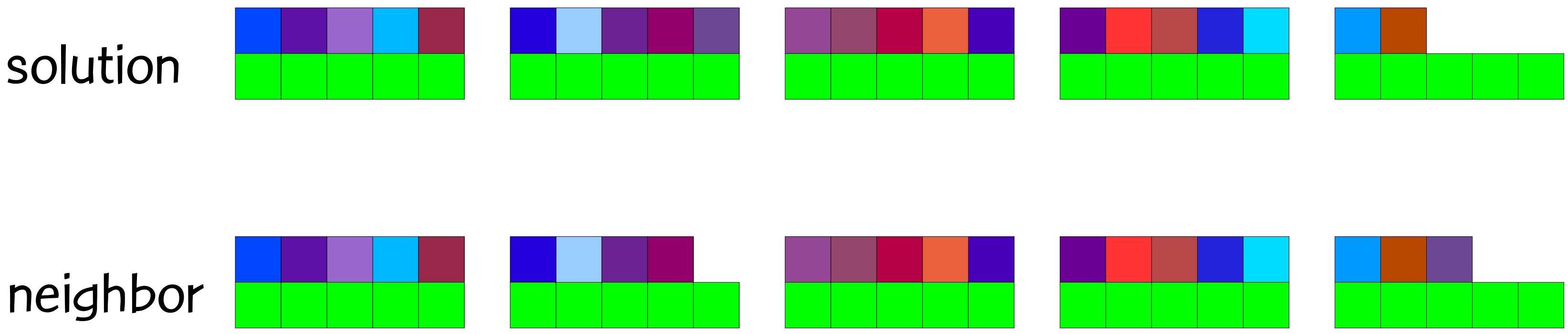
- Move unit neighborhood: Moves unit from current period to available period.

solution



GRASP local search phase

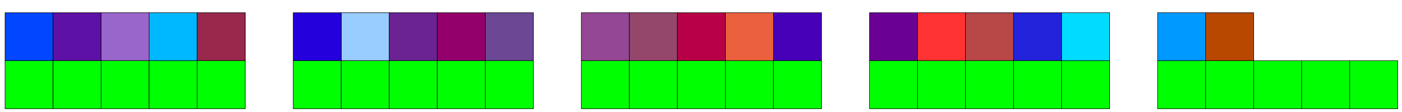
- Move unit neighborhood: Moves unit from current period to available period.



GRASP local search phase

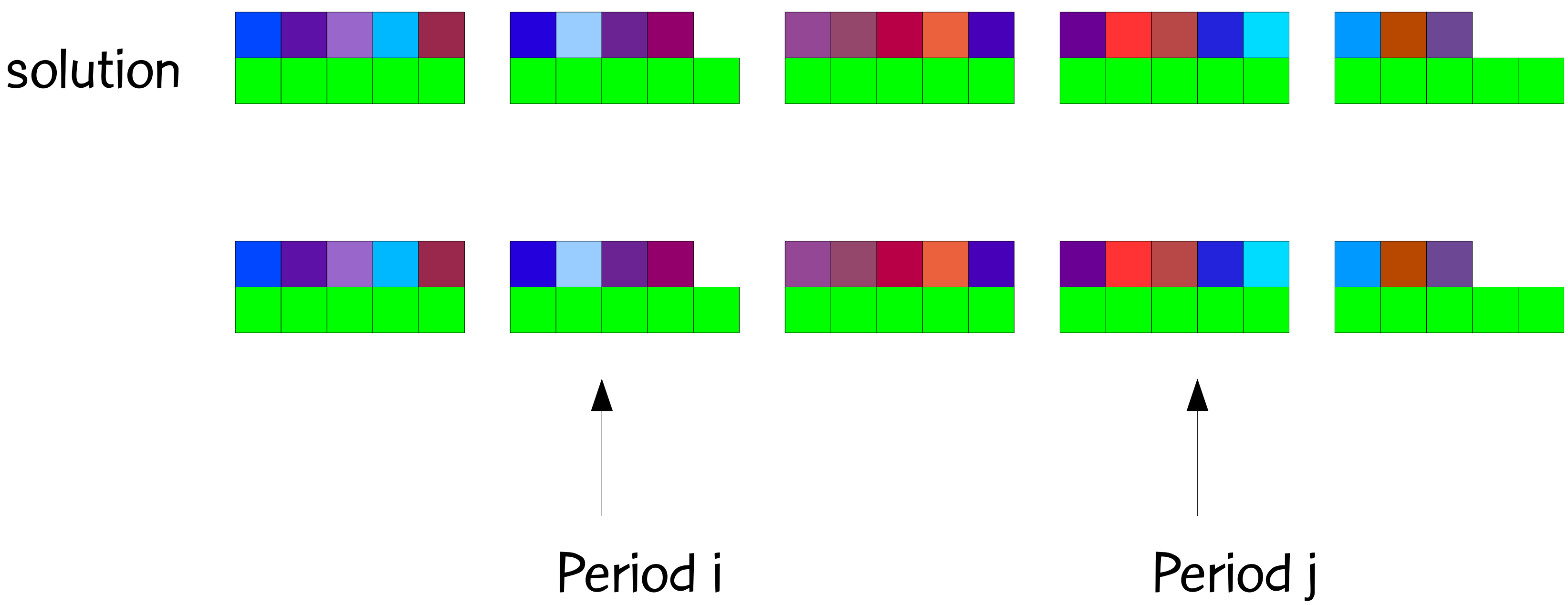
- Swap periods neighborhood: Swap all units in period i with all units in period j.

solution



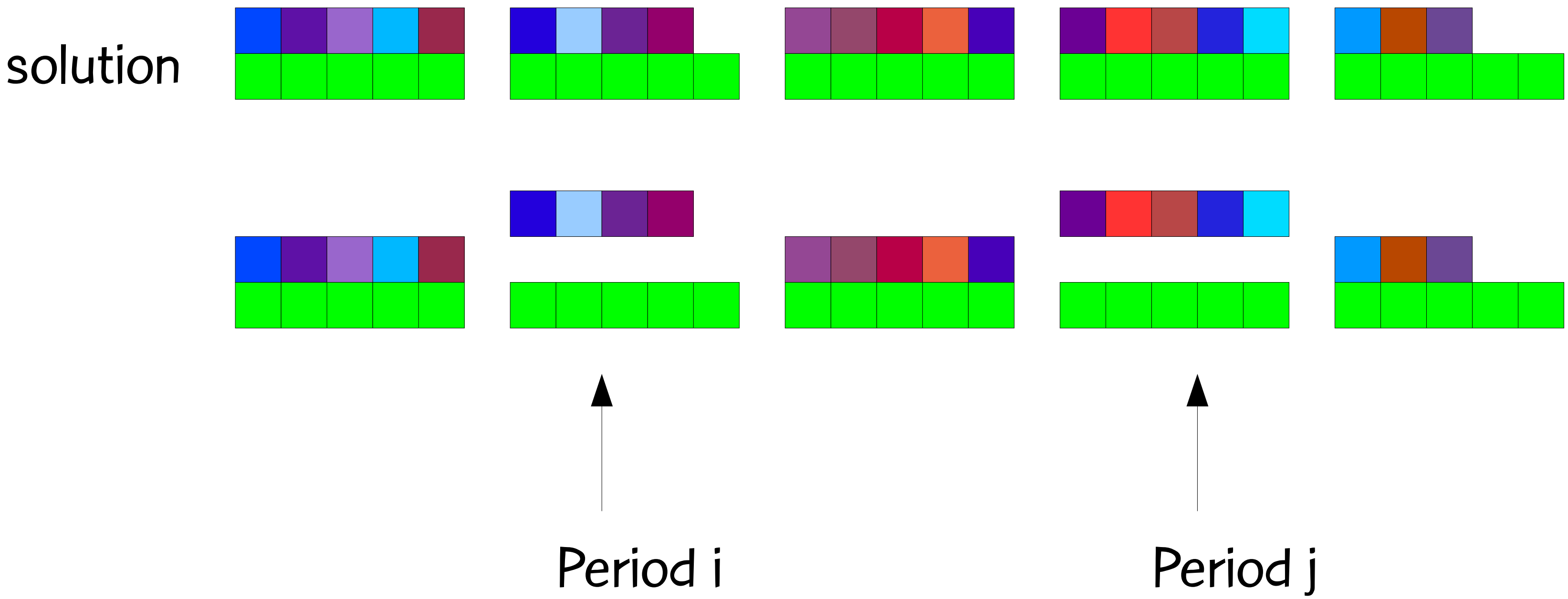
GRASP local search phase

- Swap periods neighborhood: Swap all units in period i with all units in period j.



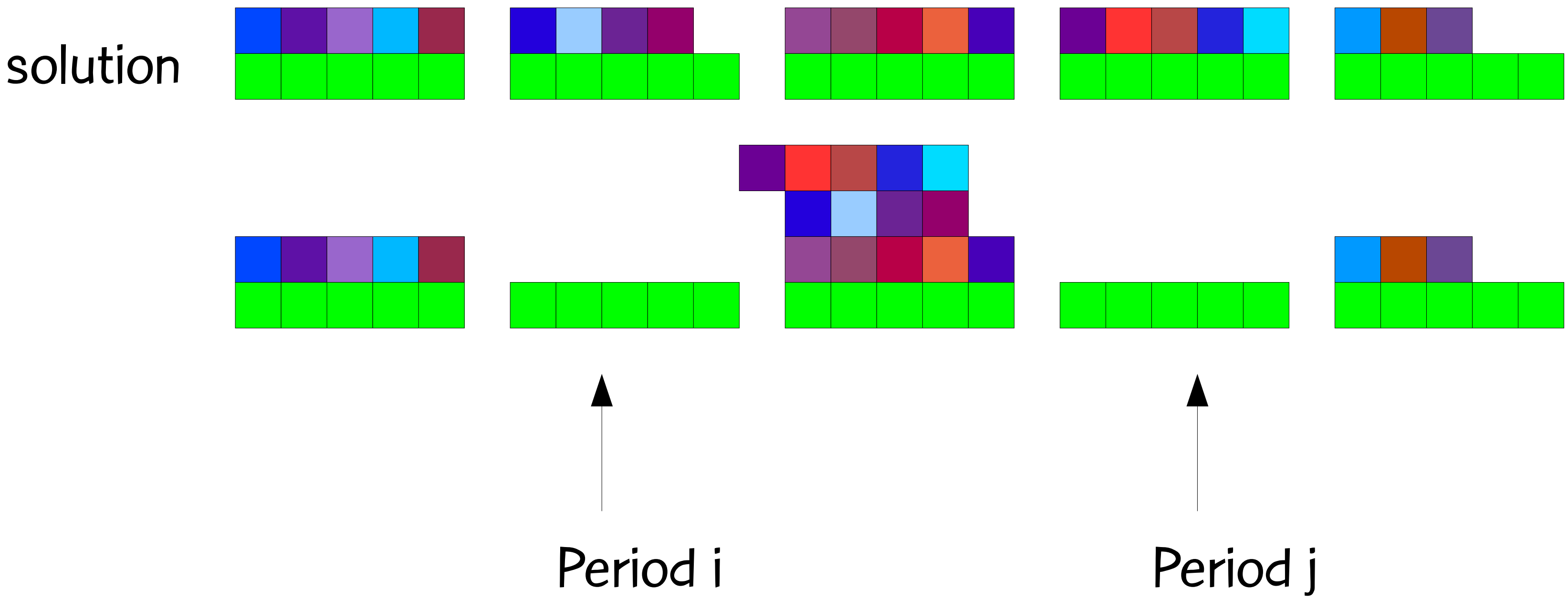
GRASP local search phase

- Swap periods neighborhood: Swap all units in period i with all units in period j.



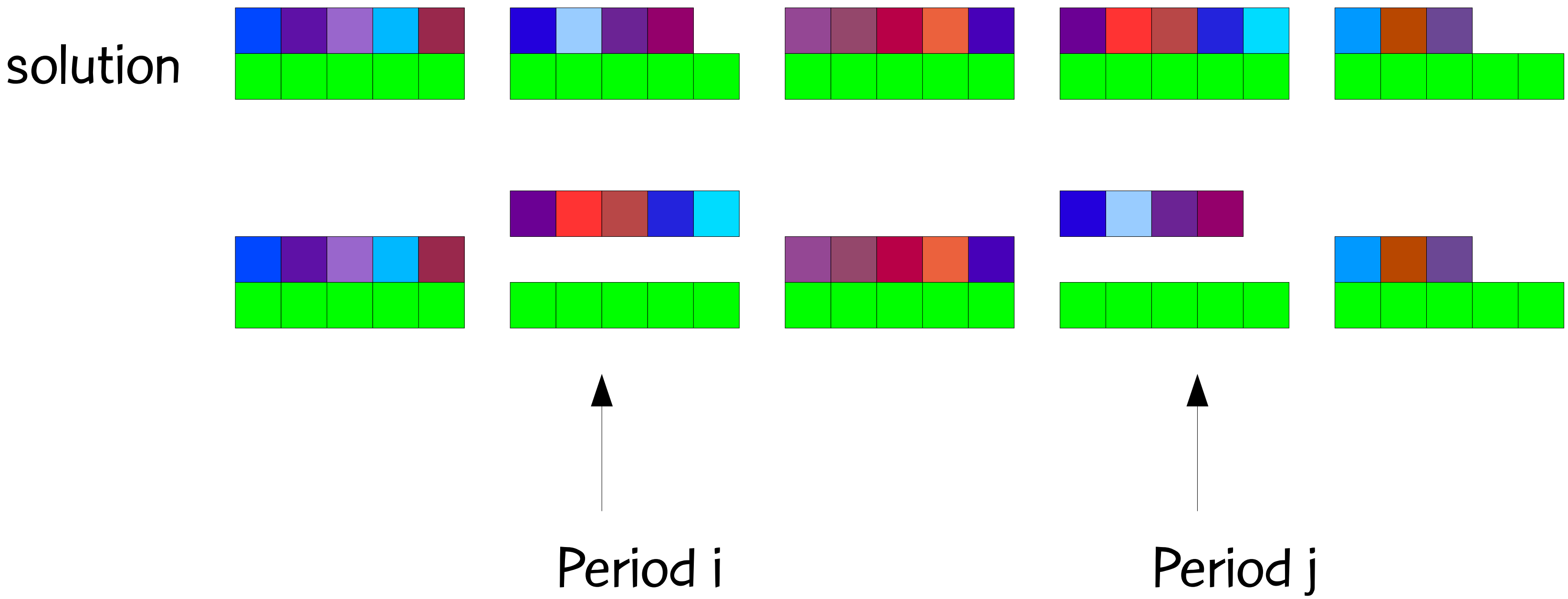
GRASP local search phase

- Swap periods neighborhood: Swap all units in period i with all units in period j.



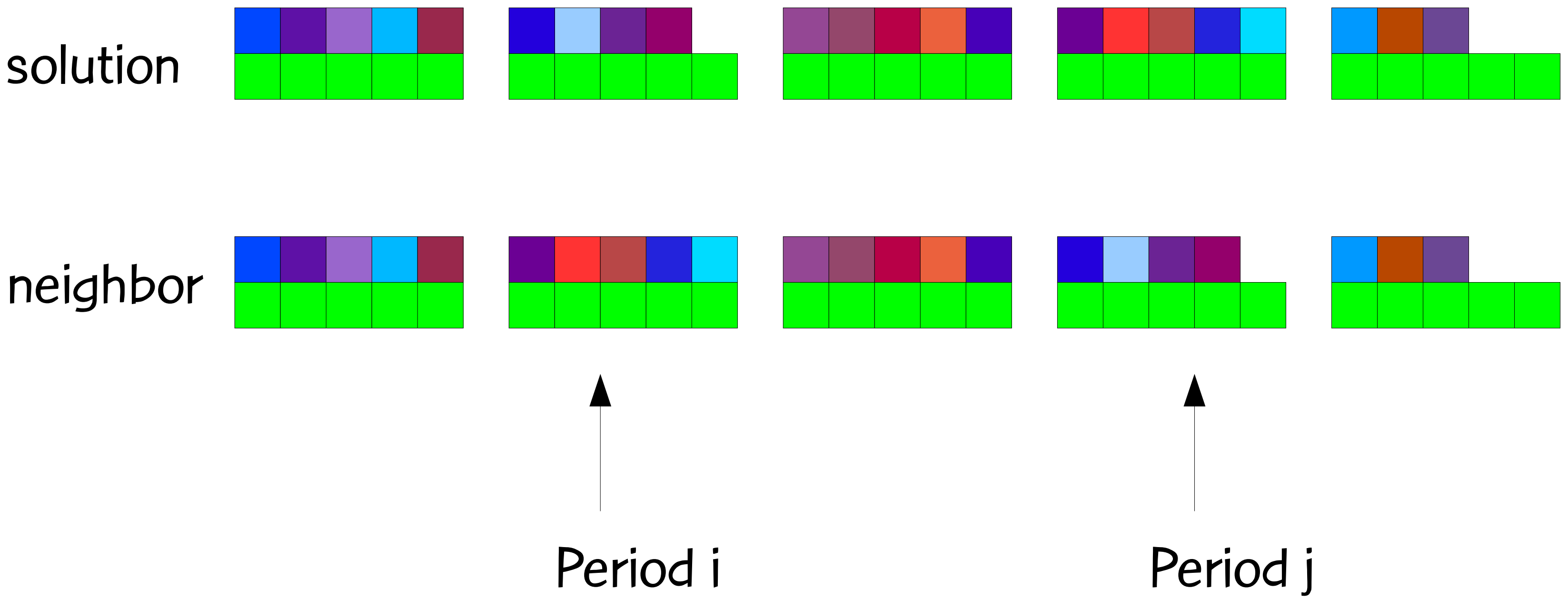
GRASP local search phase

- Swap periods neighborhood: Swap all units in period i with all units in period j.

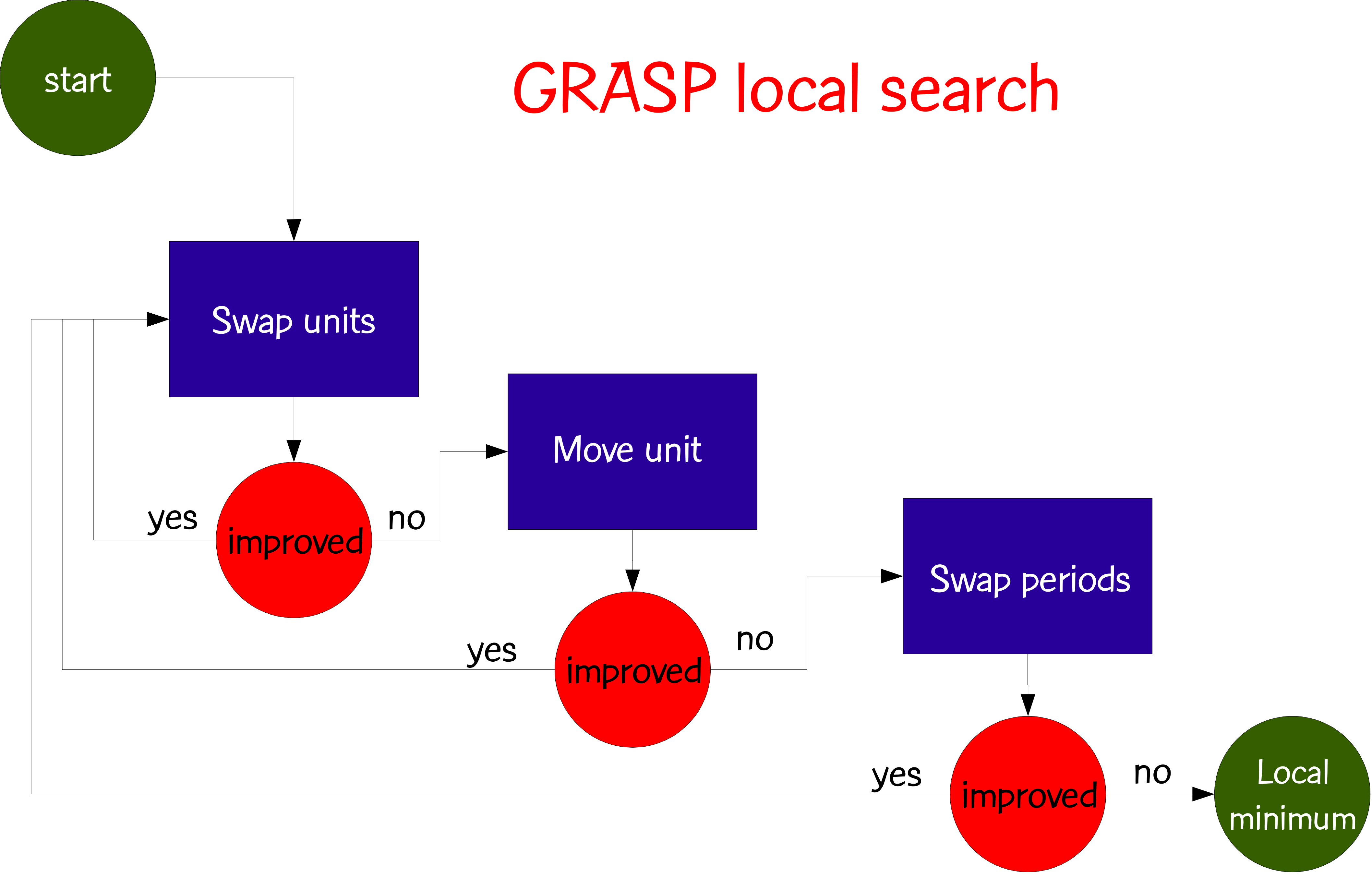


GRASP local search phase

- Swap periods neighborhood: Swap all units in period i with all units in period j.



GRASP local search



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 of more groups of phones that 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 of more groups of phones that 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

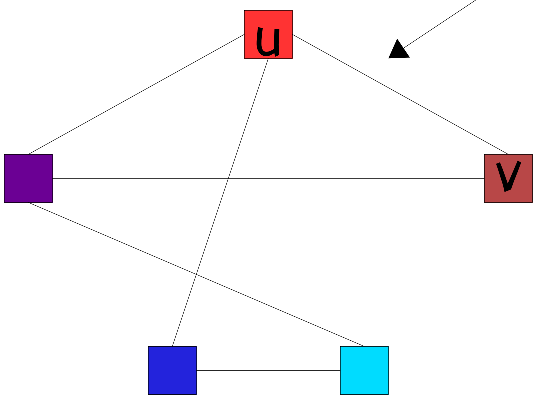
- Phone migration occurs when an organization upgrades to a newer phone switch (PBX).
- All phones using the old PBX must be moved to the new PBX.
- Each phone belong to one of more groups of phones that should to be moved together in same time period.
- Given penalties for not moving a pair of phones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

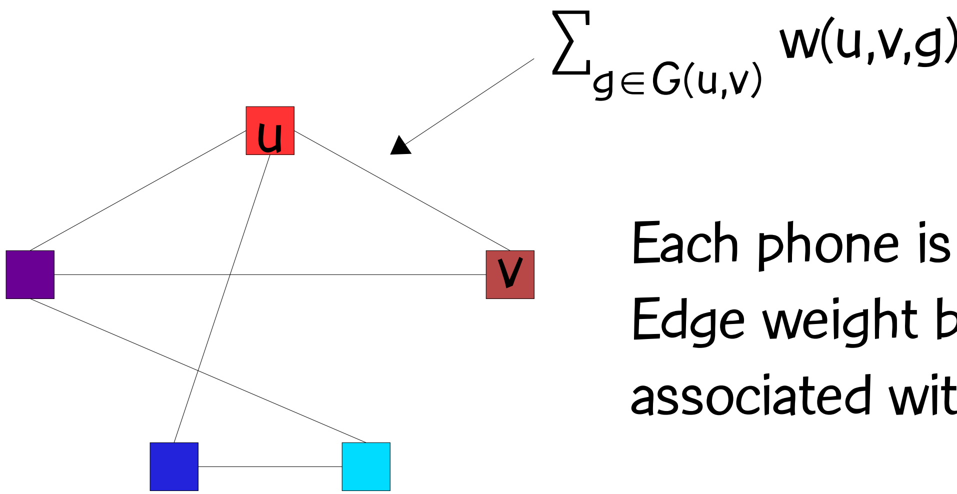
PBX telephone migration scheduling

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

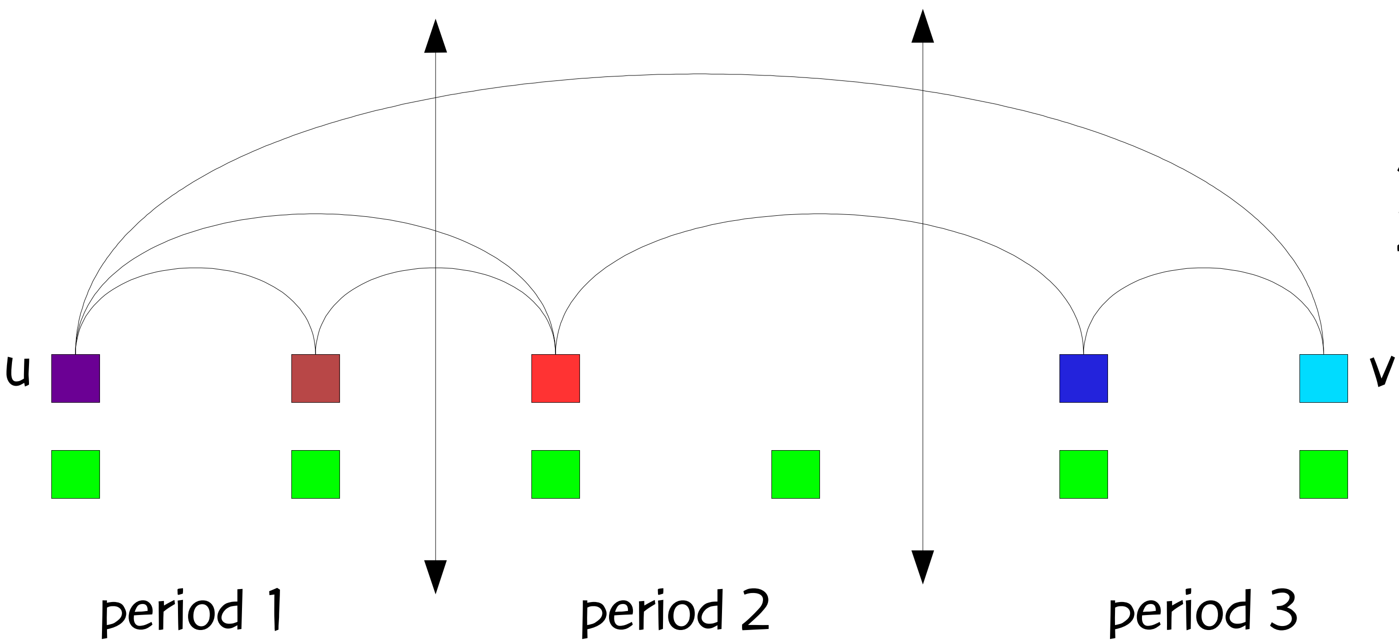


Each phone is a unit.
Edge weight between phones is penalty associated with phone pair.

PBX telephone migration scheduling

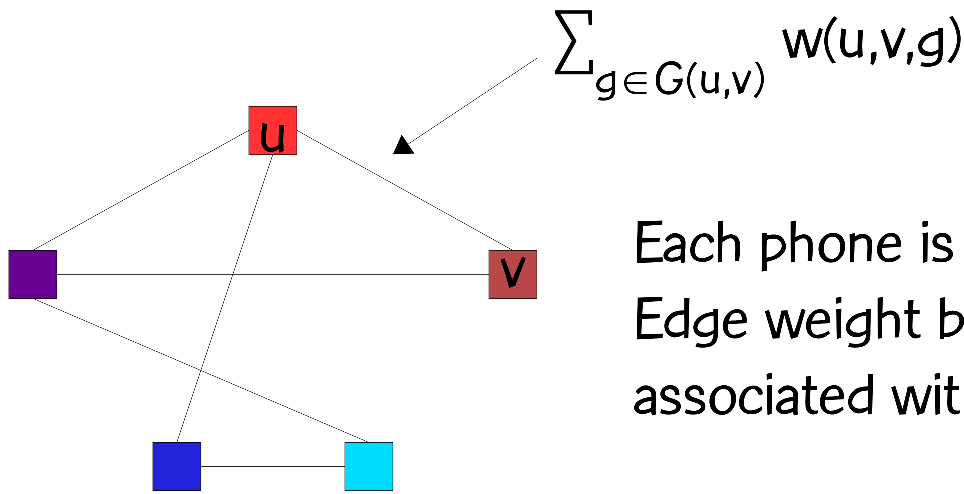


Each phone is a unit.
Edge weight between phones is penalty associated with phone pair.

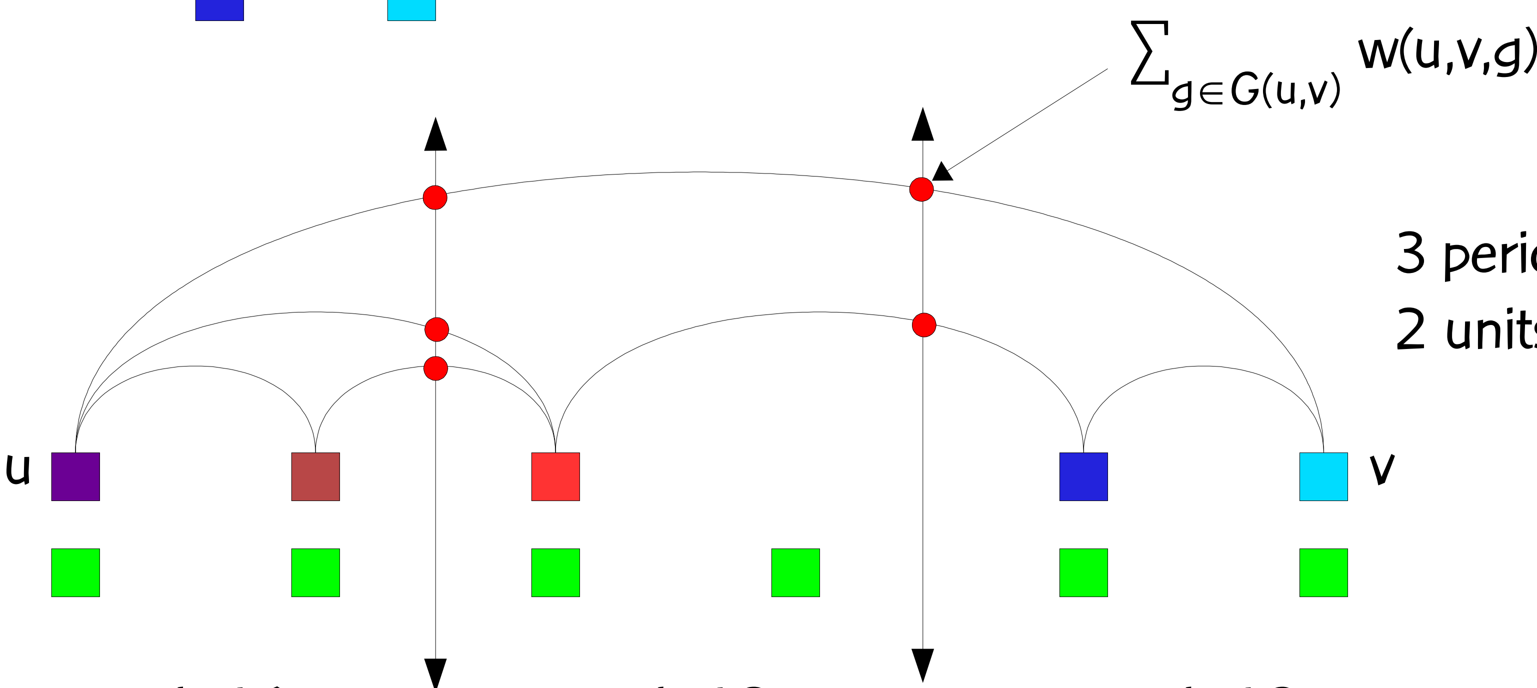


3 periods
2 units/period max

PBX telephone migration scheduling



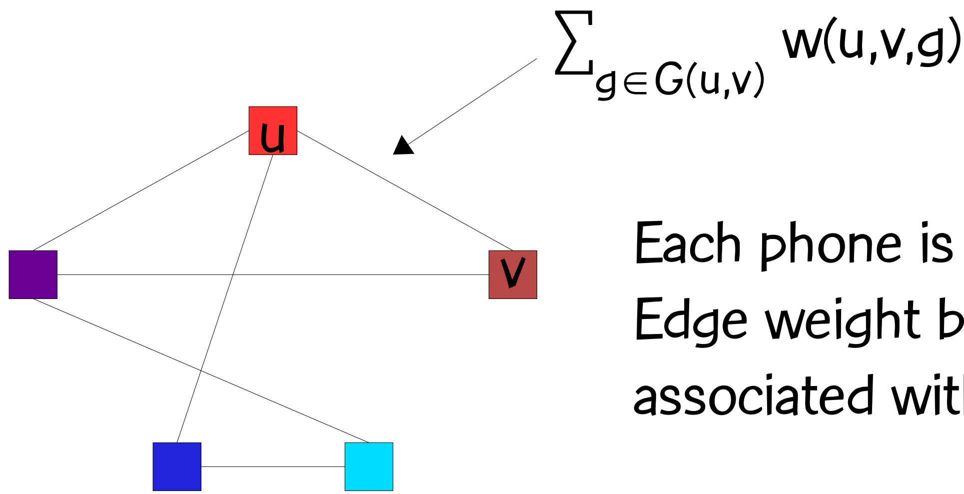
Each phone is a unit.
Edge weight between phones is penalty associated with phone pair.



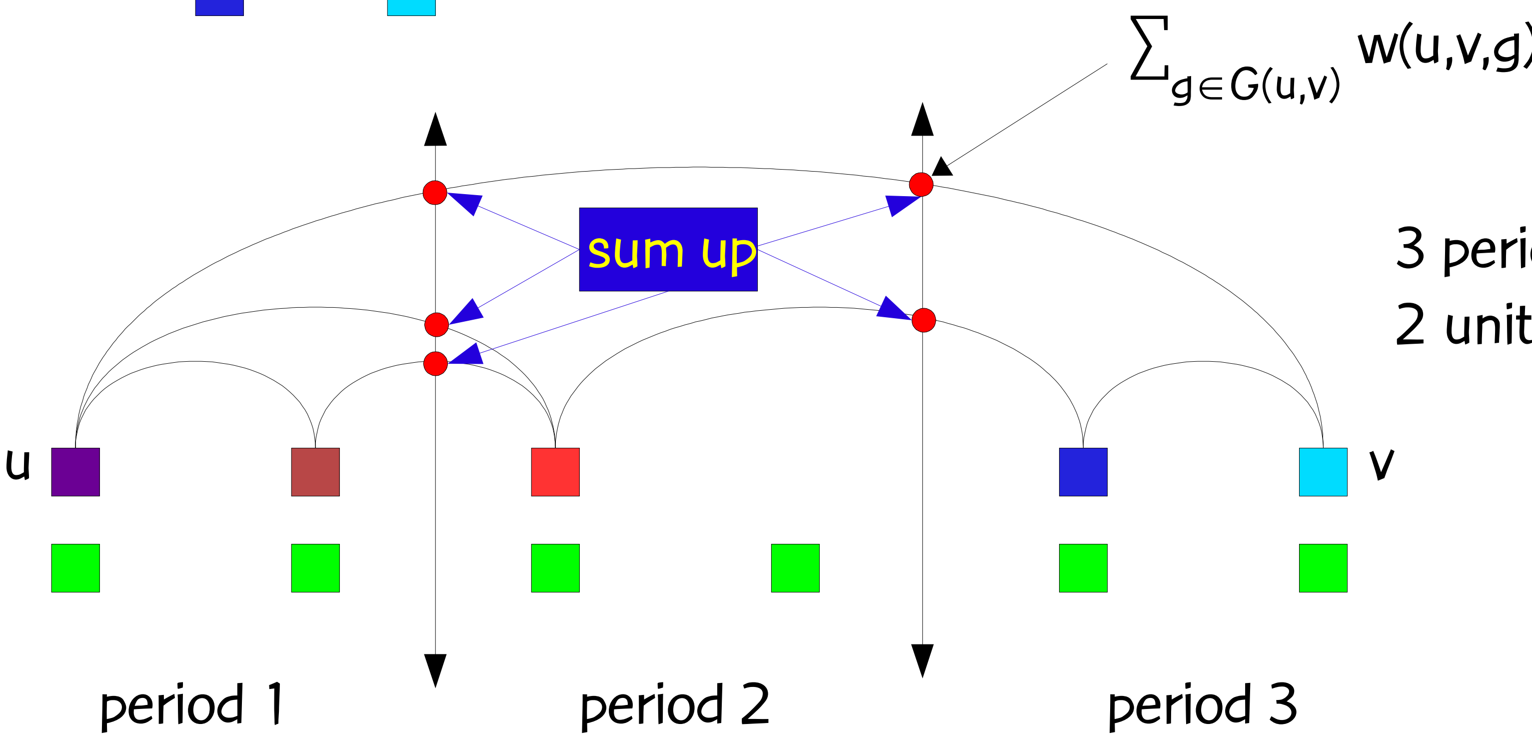
3 periods
2 units/period max

at&t
Your world. Delivered.

PBX telephone migration scheduling

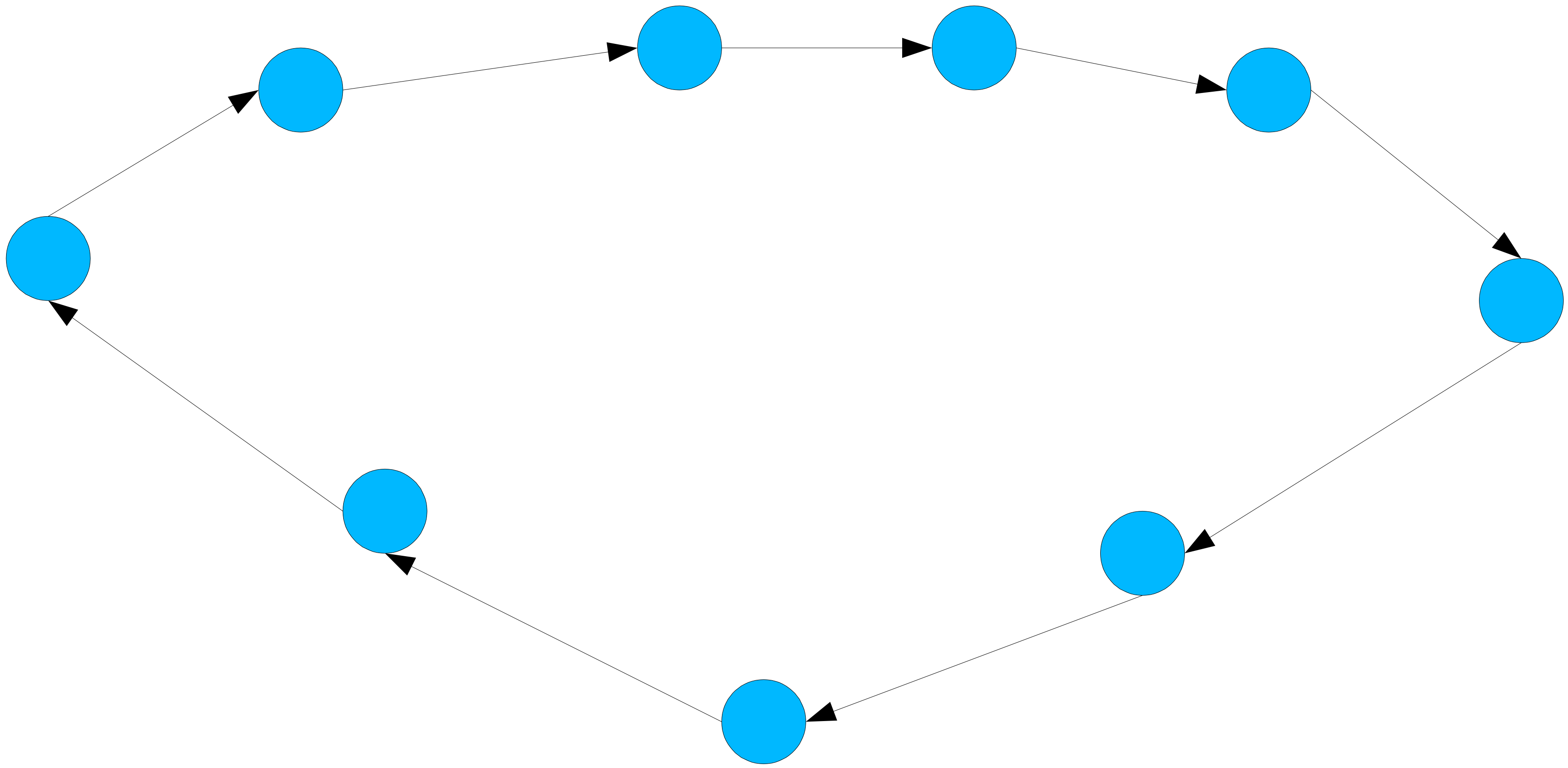


Each phone is a unit.
Edge weight between phones is penalty associated with phone pair.



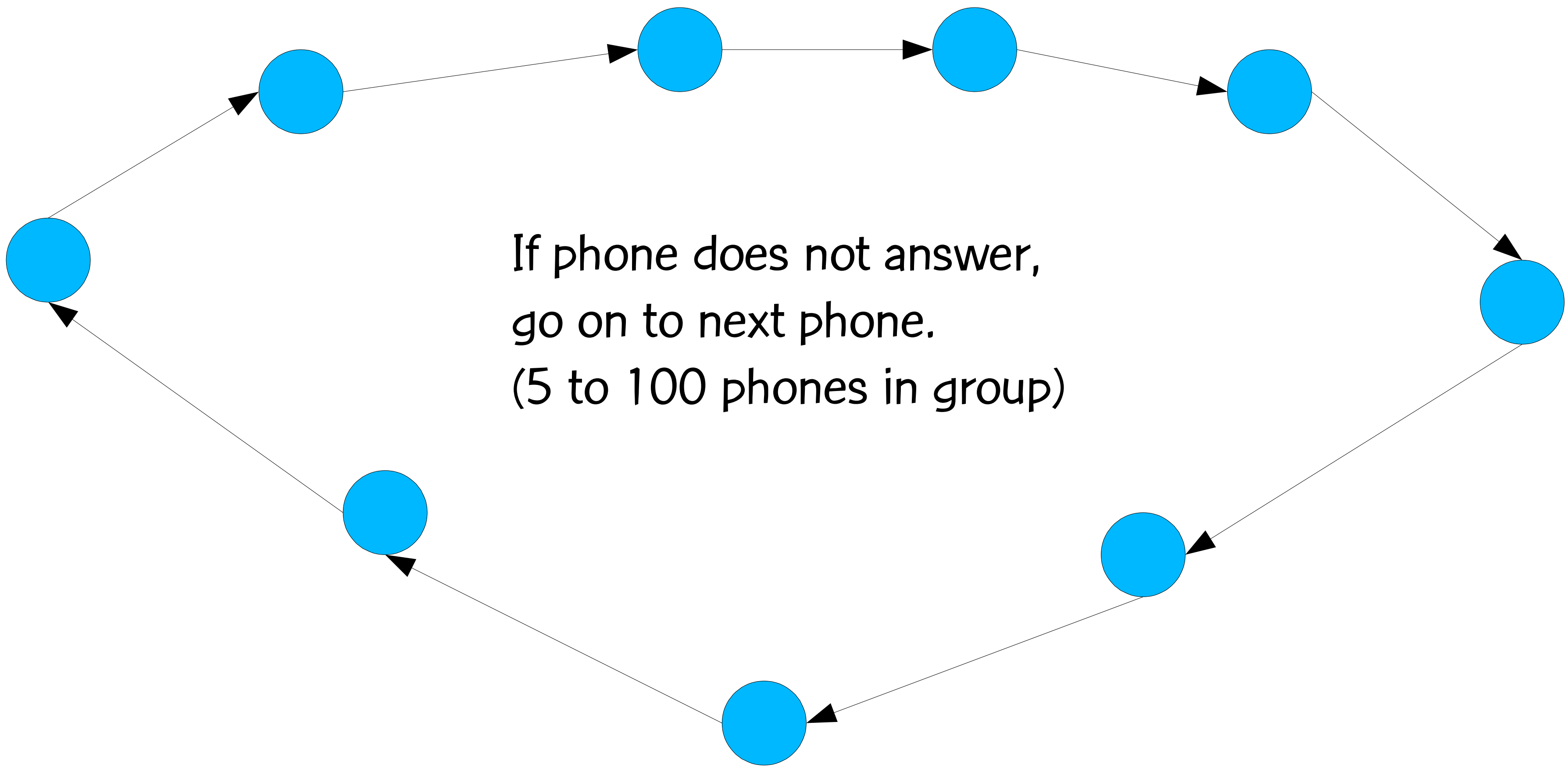
3 periods
2 units/period max

Multi-line hunt group

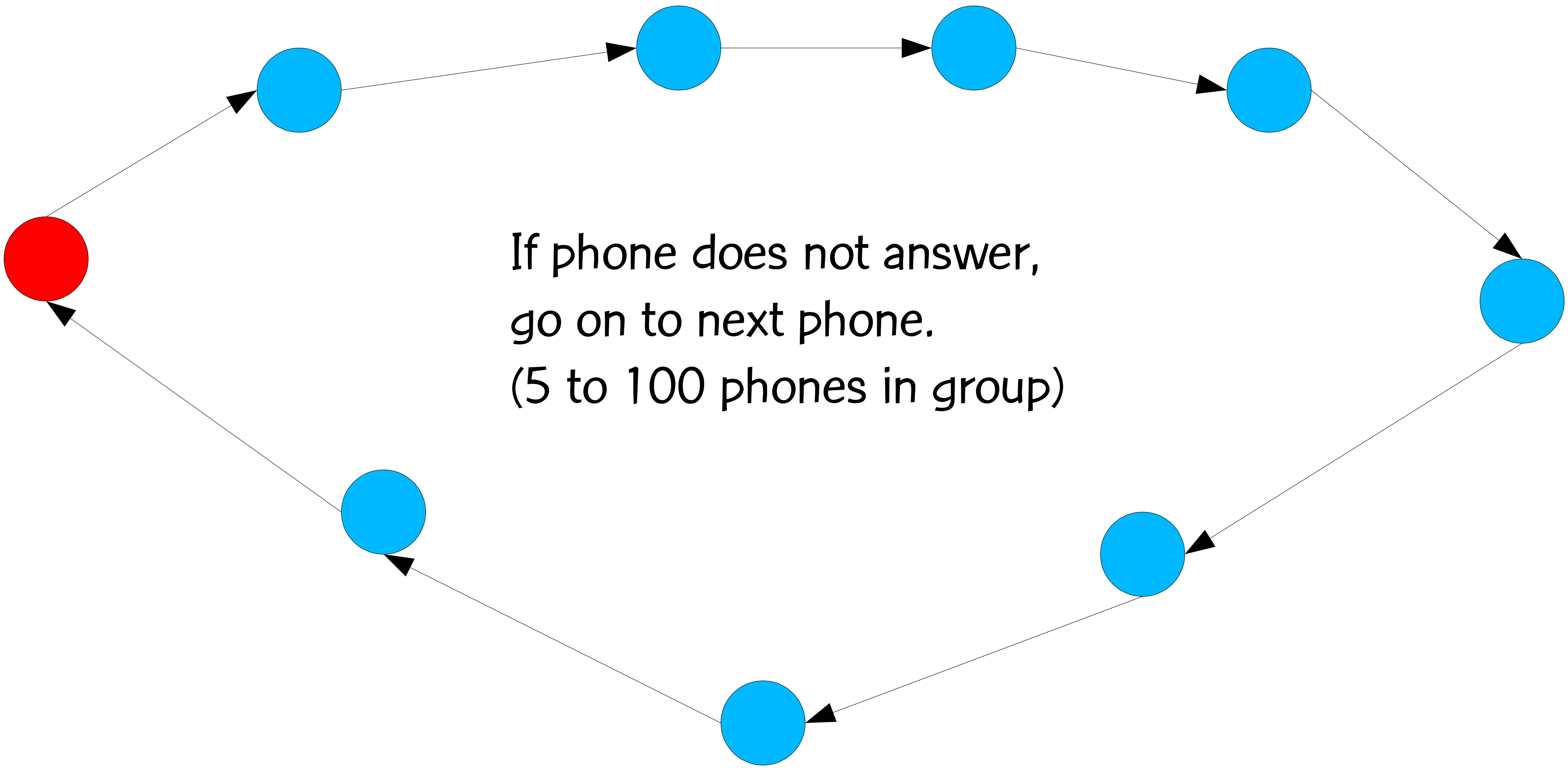


at&t
Your world. Delivered.

Multi-line hunt group

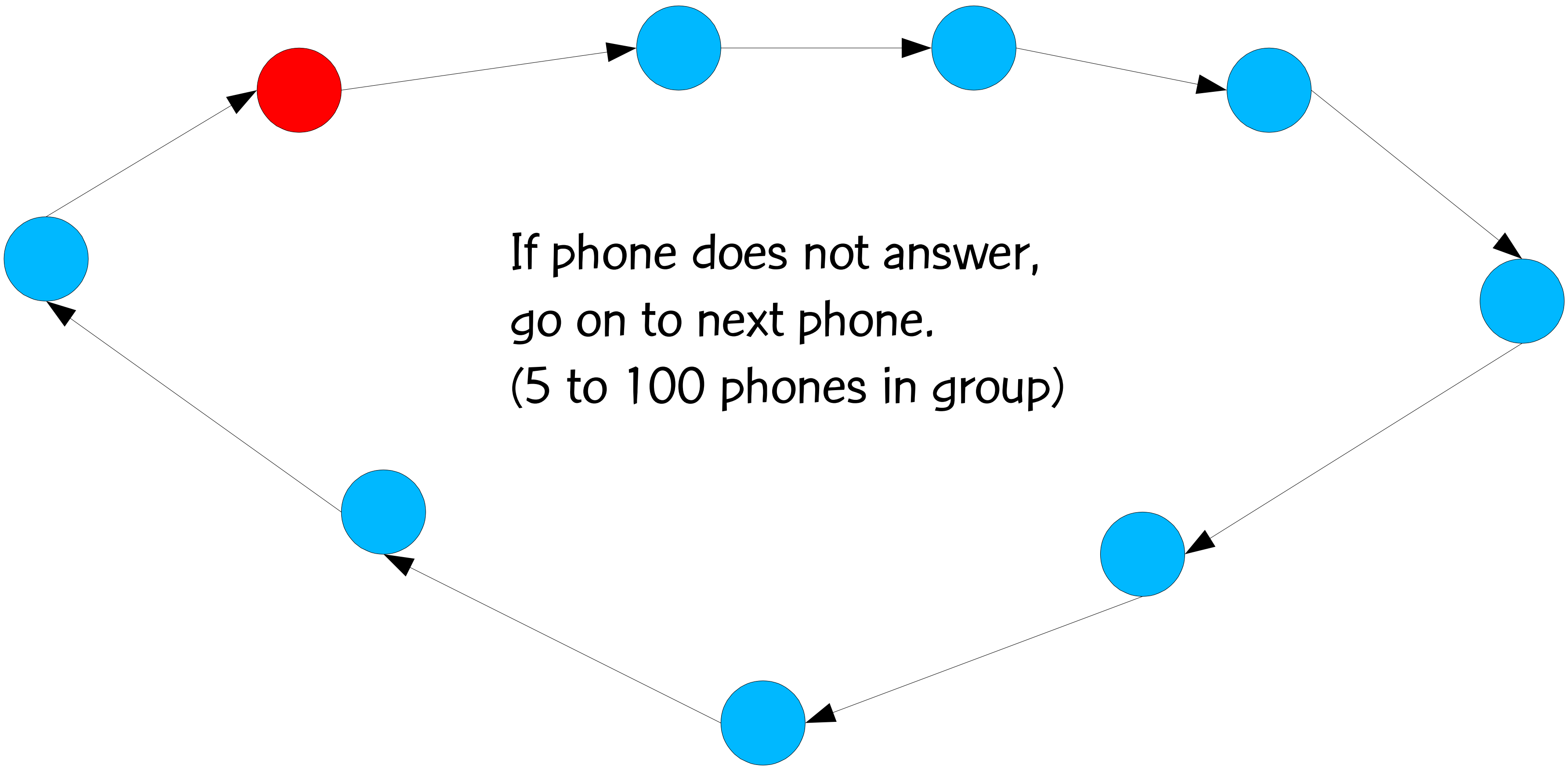


Multi-line hunt group



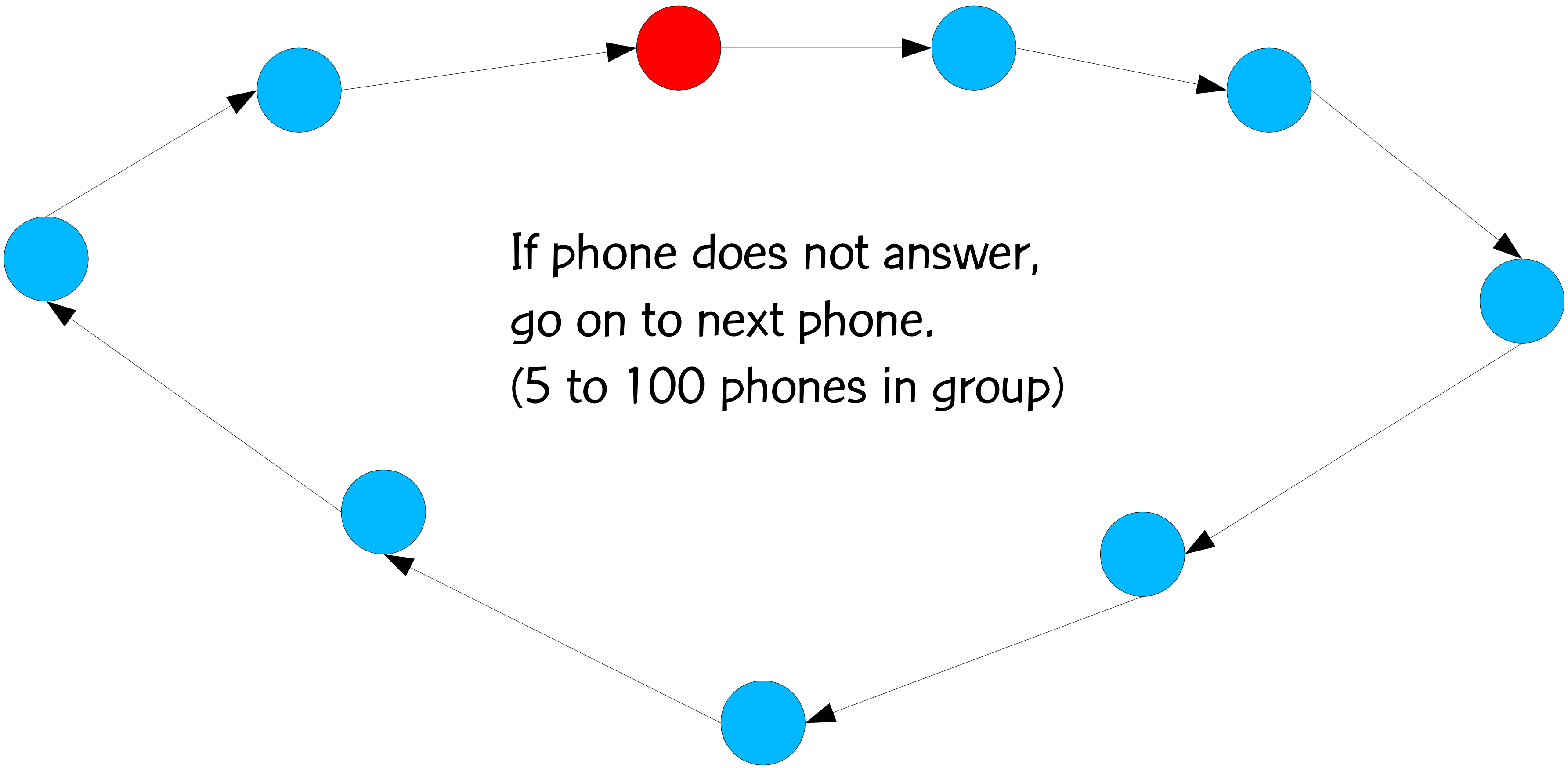
If phone does not answer,
go on to next phone.
(5 to 100 phones in group)

Multi-line hunt group

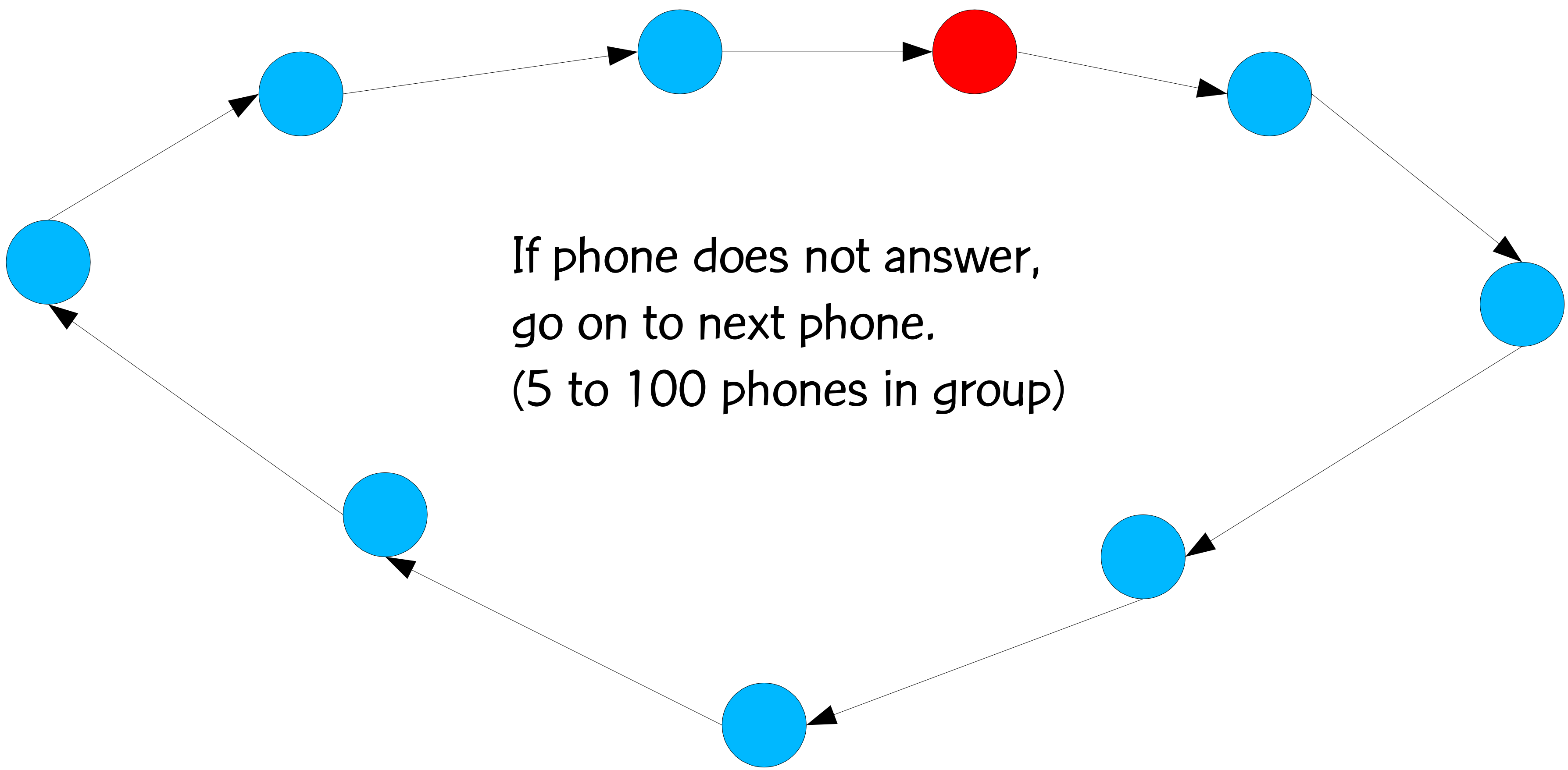


If phone does not answer,
go on to next phone.
(5 to 100 phones in group)

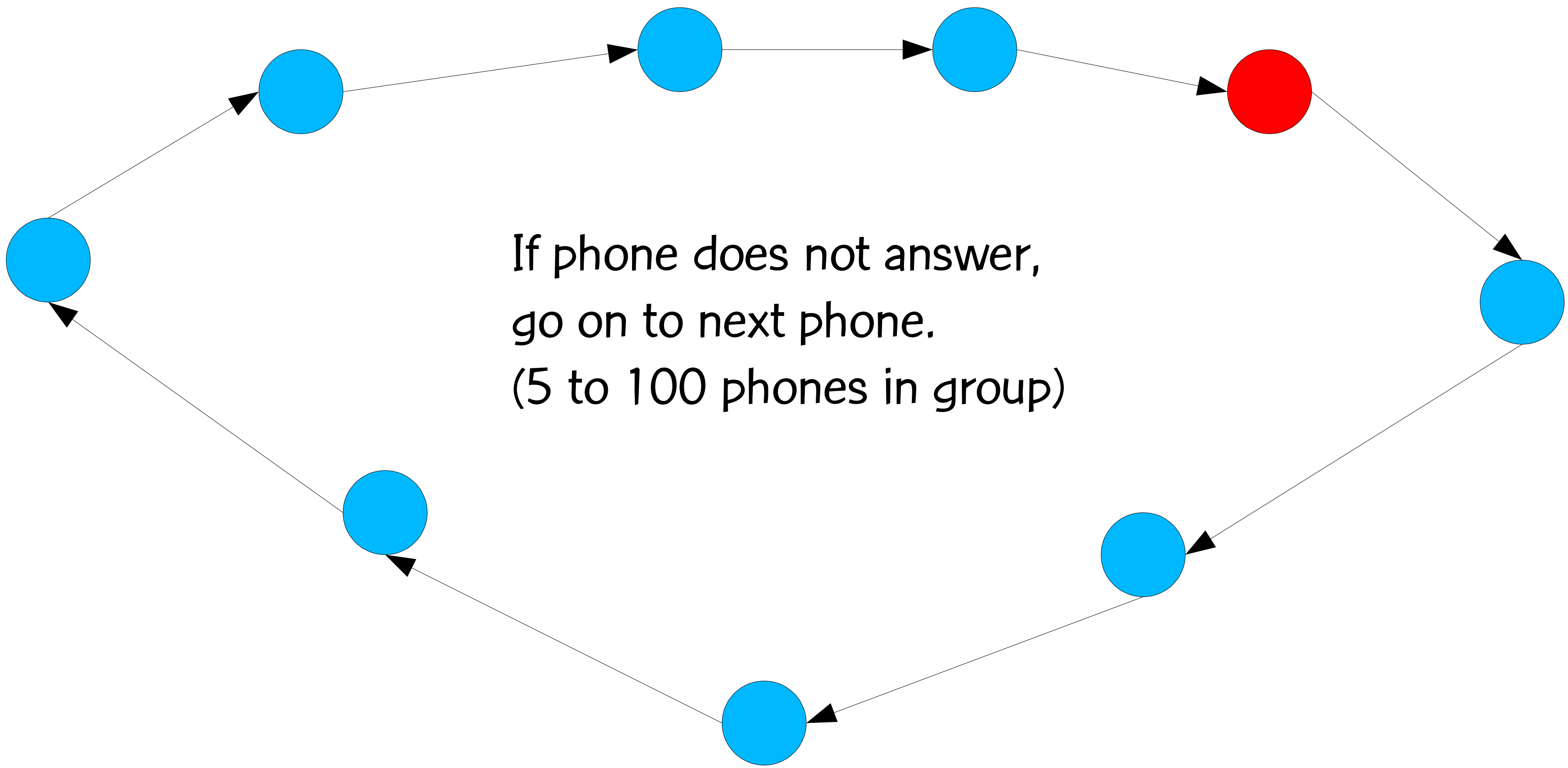
Multi-line hunt group



Multi-line hunt group



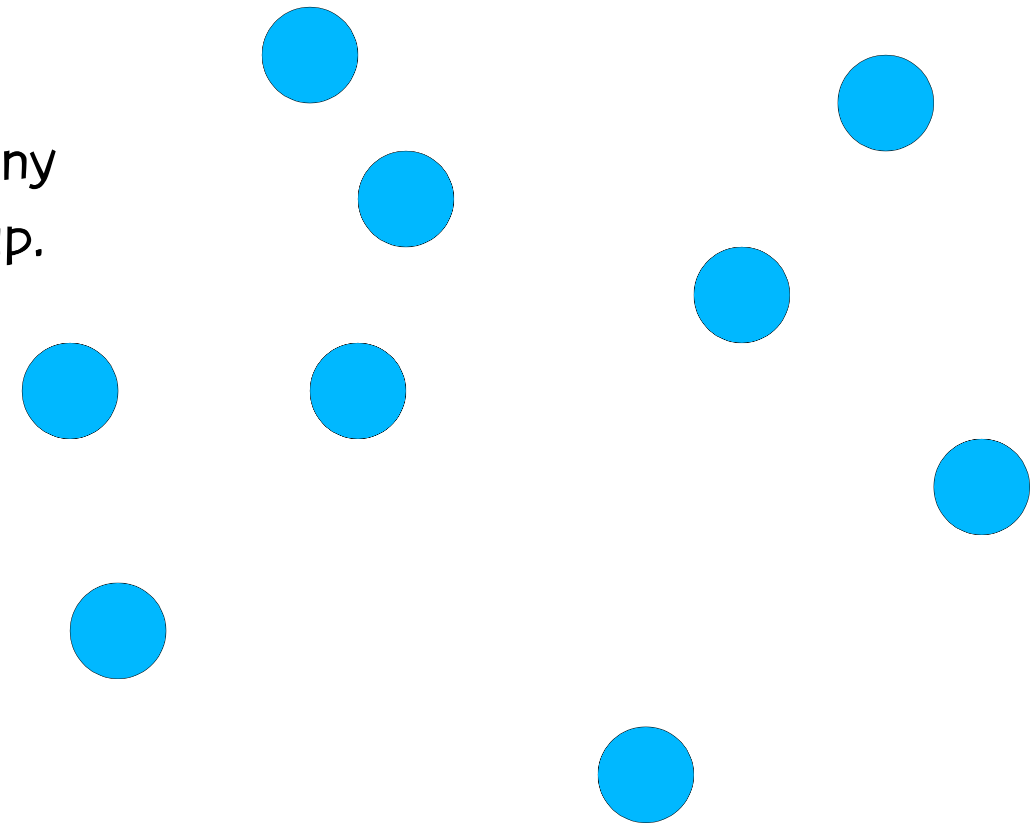
Multi-line hunt group



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

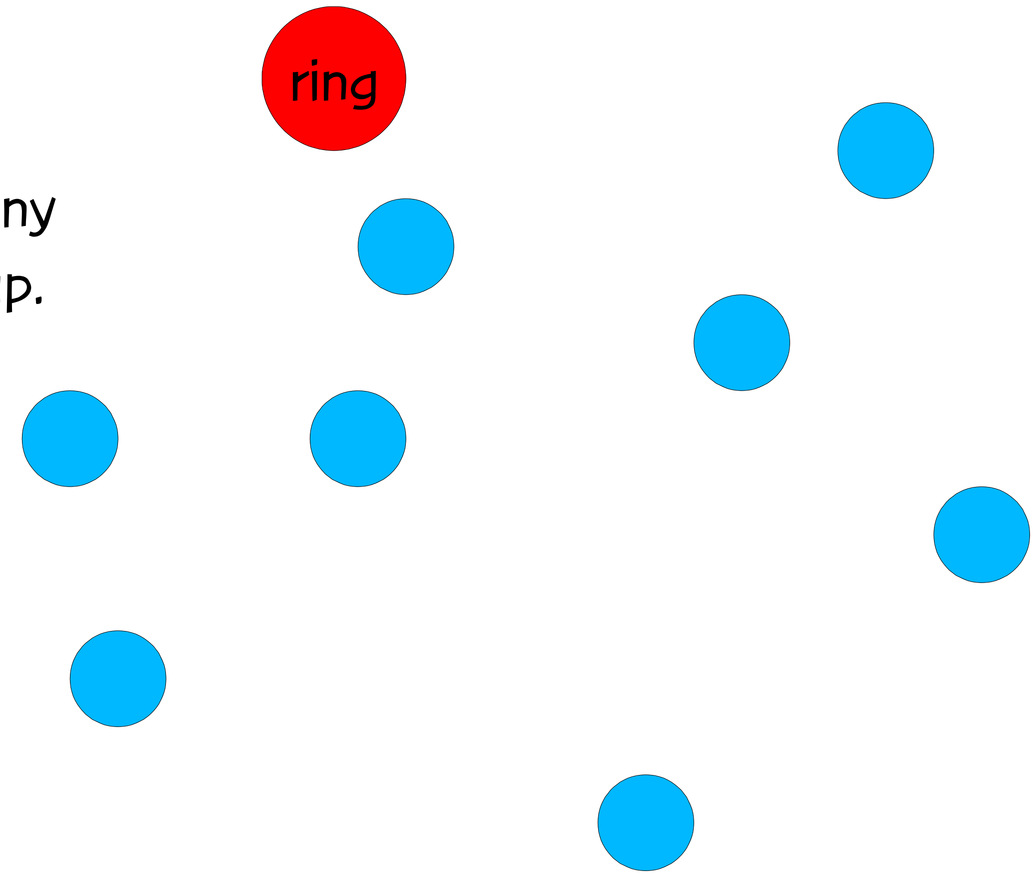
Call pickup (CPU)

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



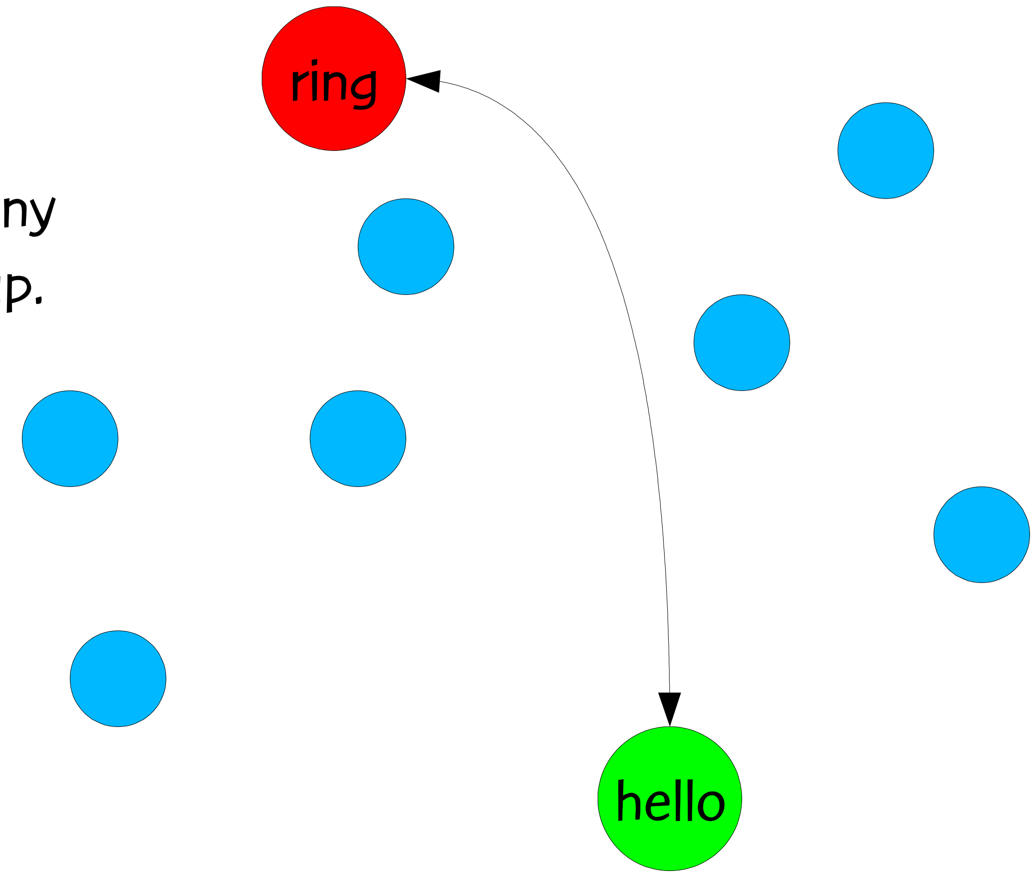
Call pickup (CPU)

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



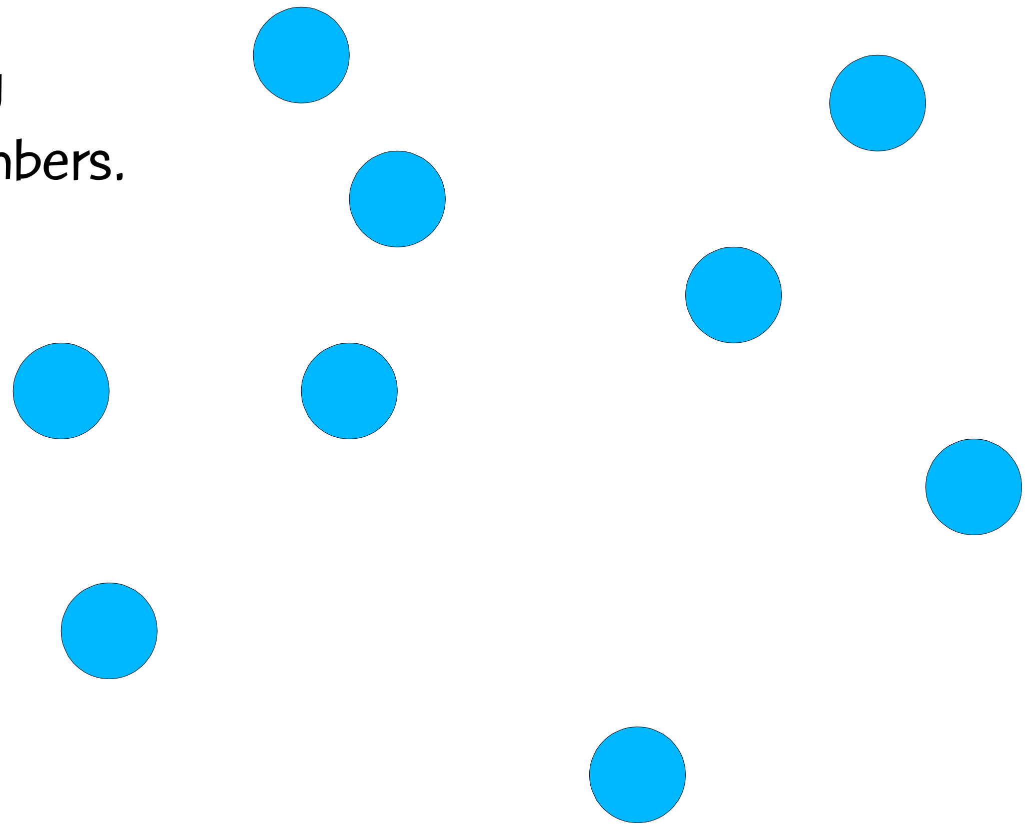
Call pickup (CPU)

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



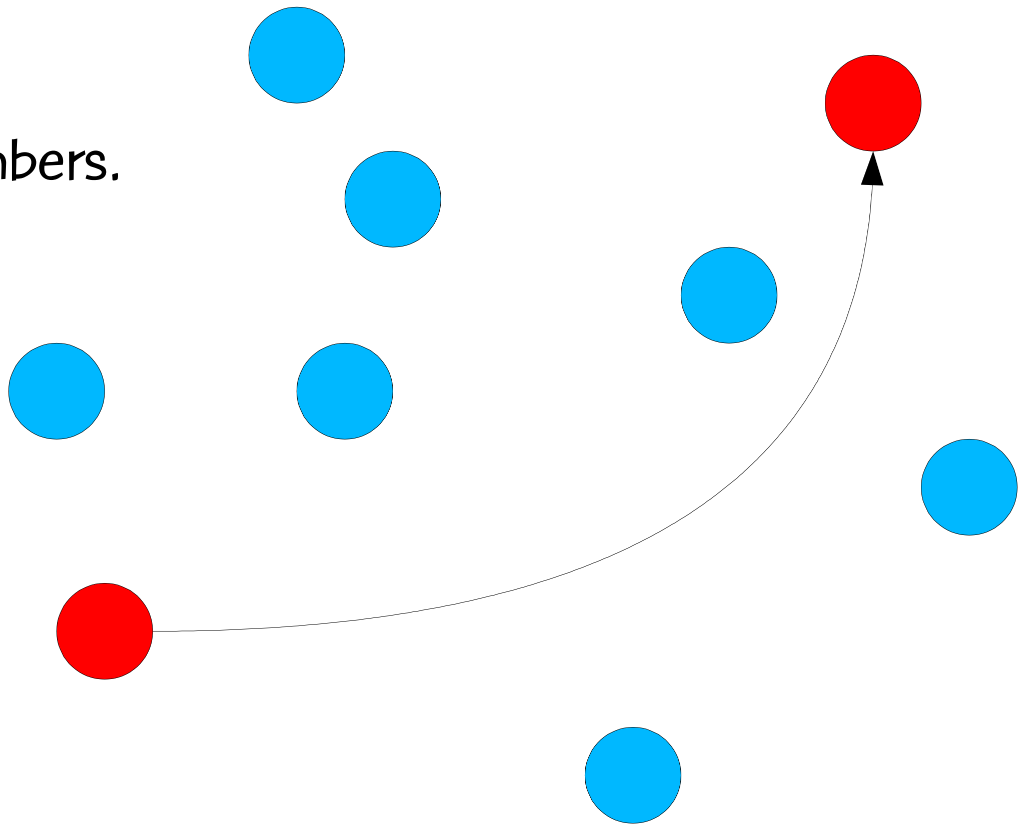
Intercomm (ICOM)

Allows speed dialing
between group members.



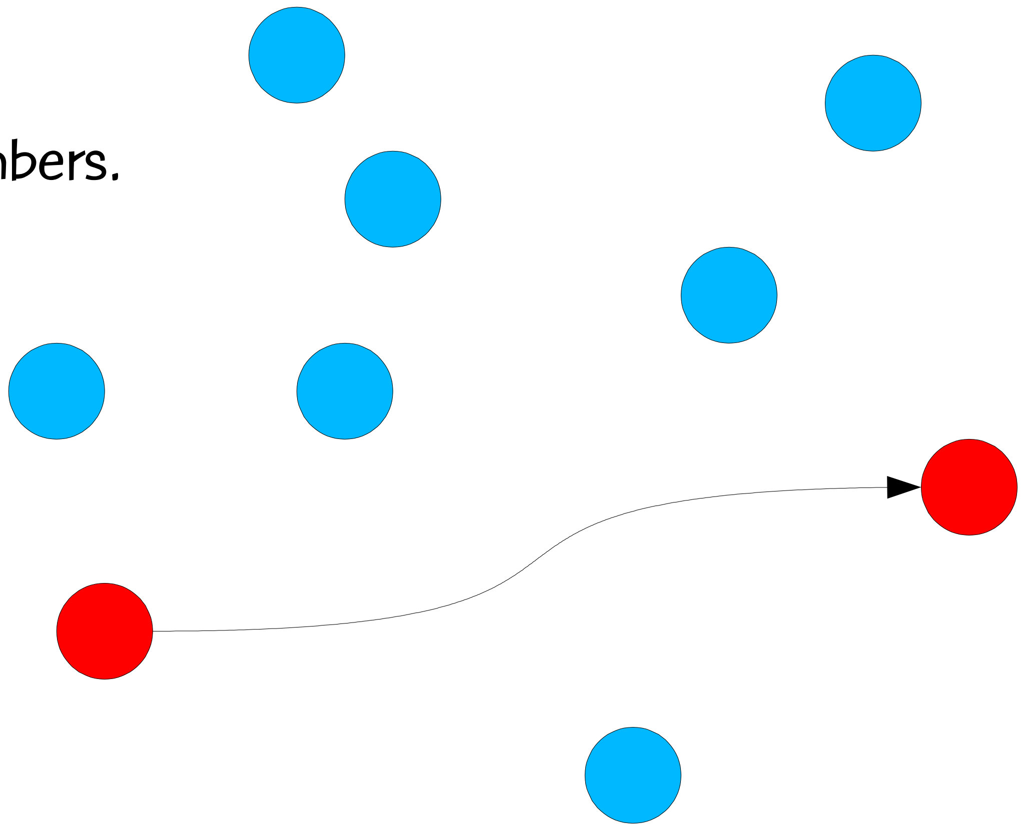
Intercomm (ICOM)

Allows speed dialing between group members.



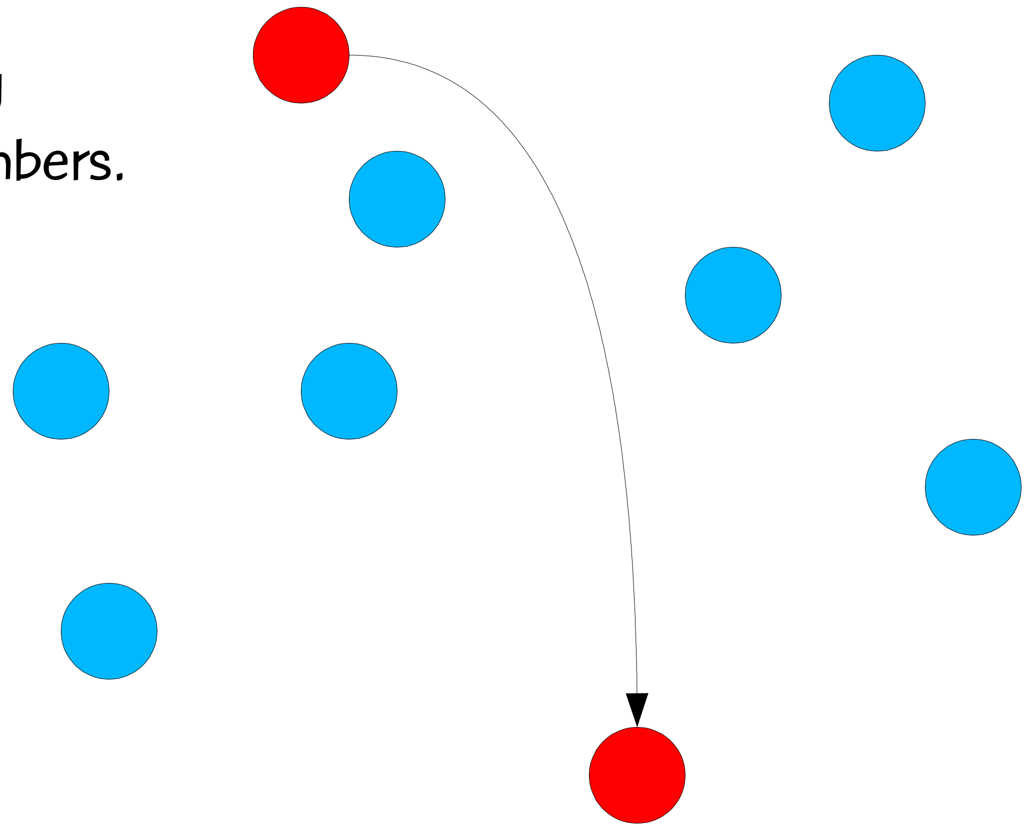
Intercomm (ICOM)

Allows speed dialing between group members.

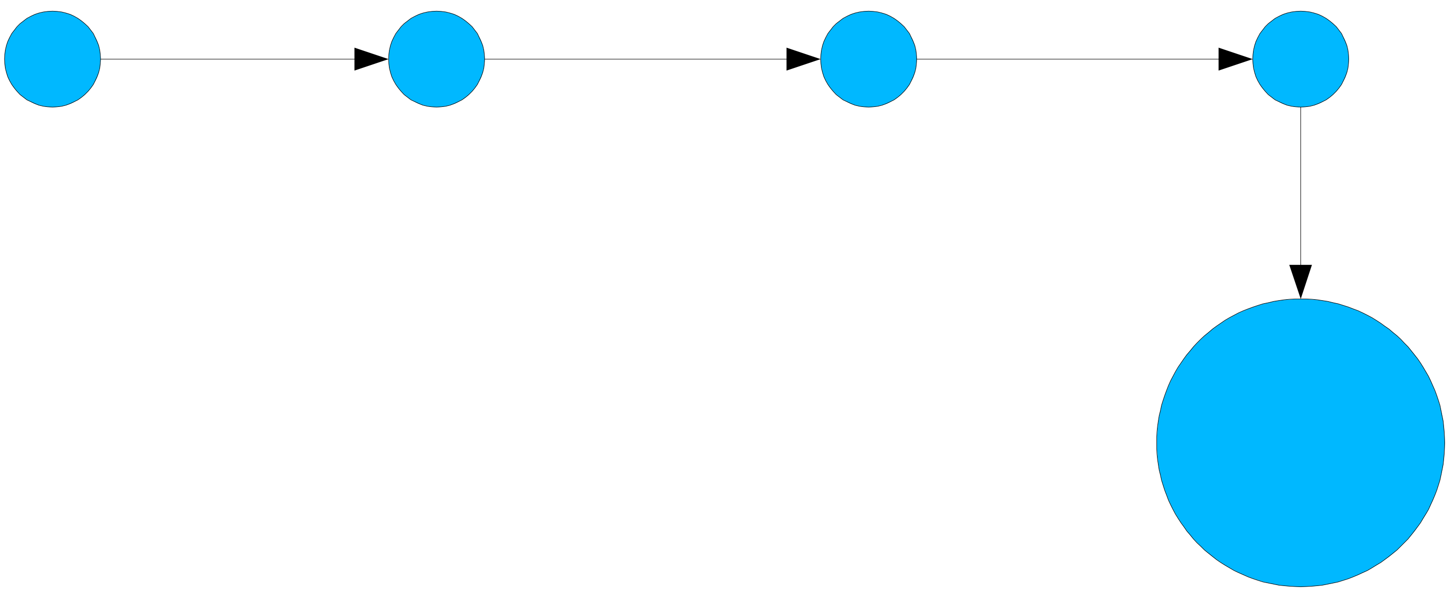


Intercomm (ICOM)

Allows speed dialing
between group members.

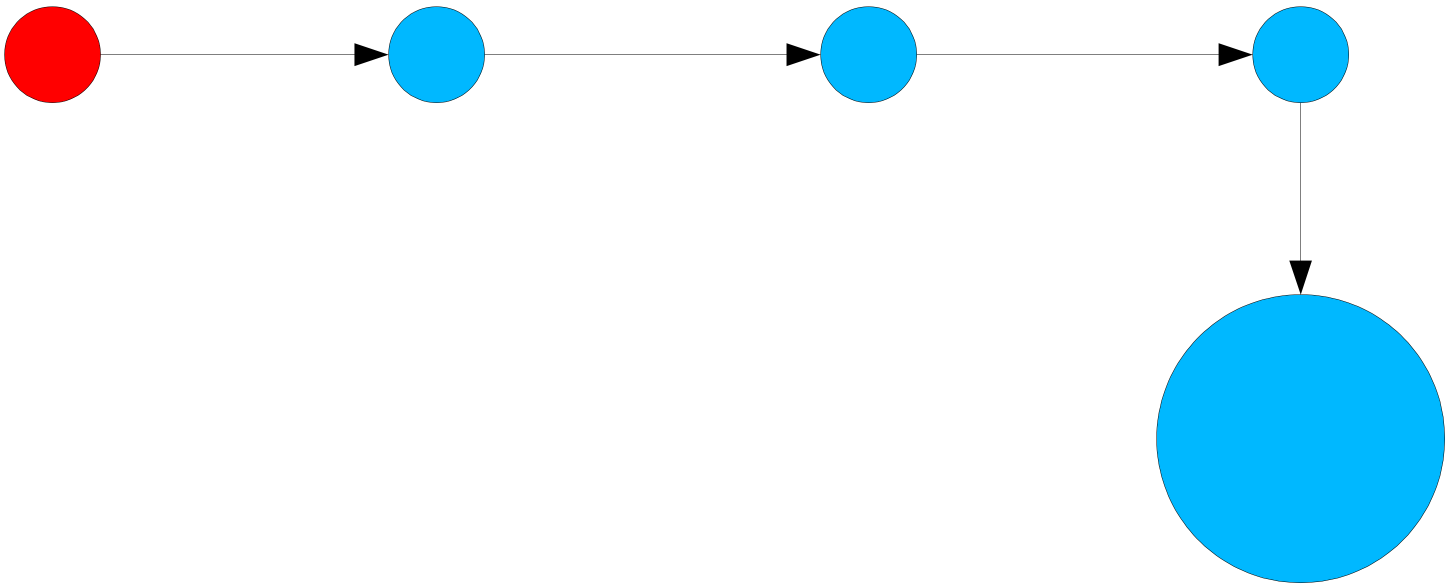


Series completion



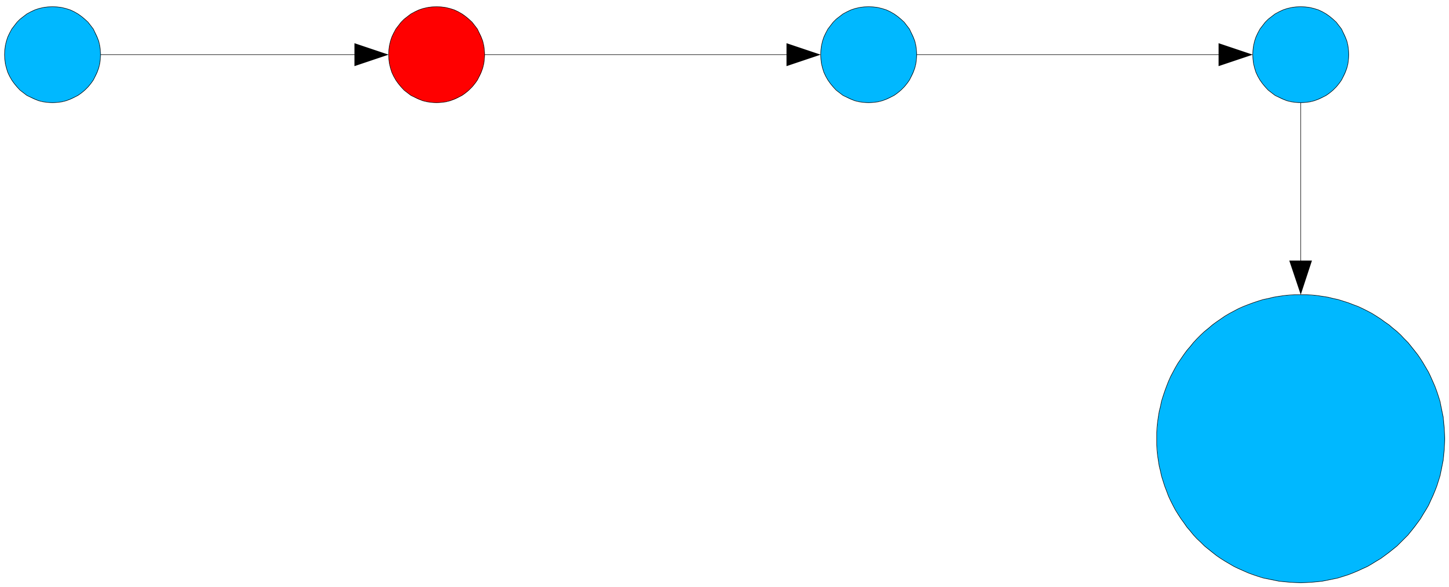
Series completion

If call not answered ...



Series completion

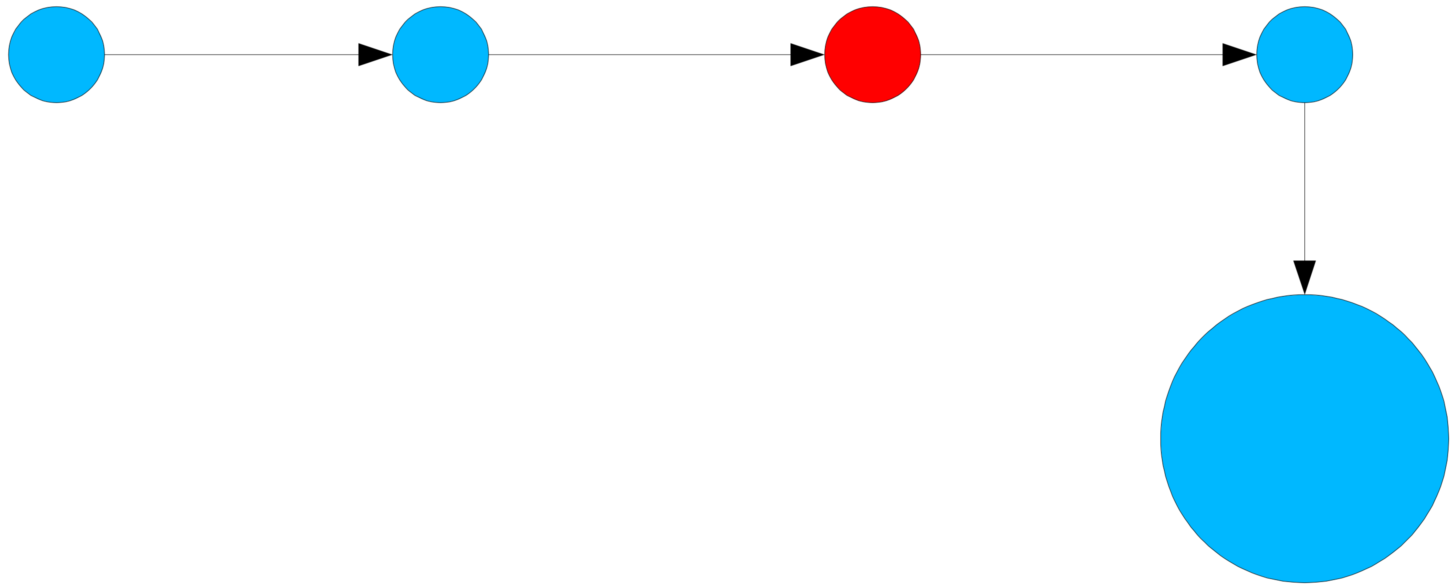
If call not answered, it moves
to ....next in series.



Series completion

If call not answered, it moves

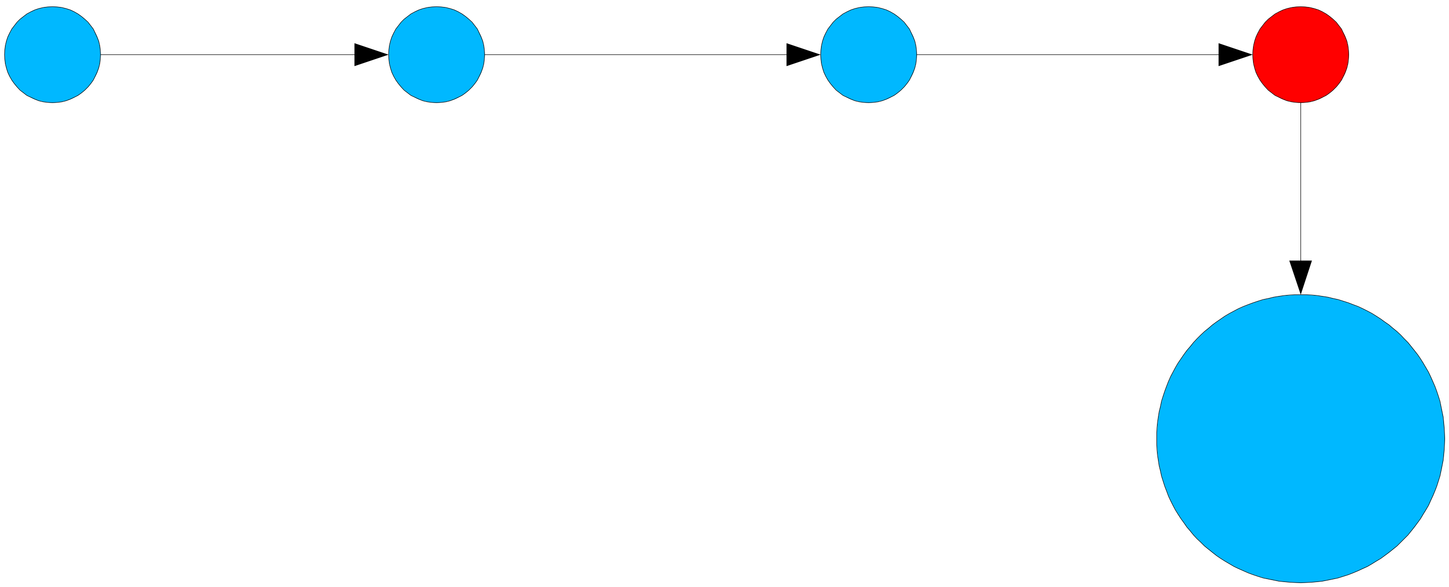
to .... next in series.



Series completion

If call not answered, it moves

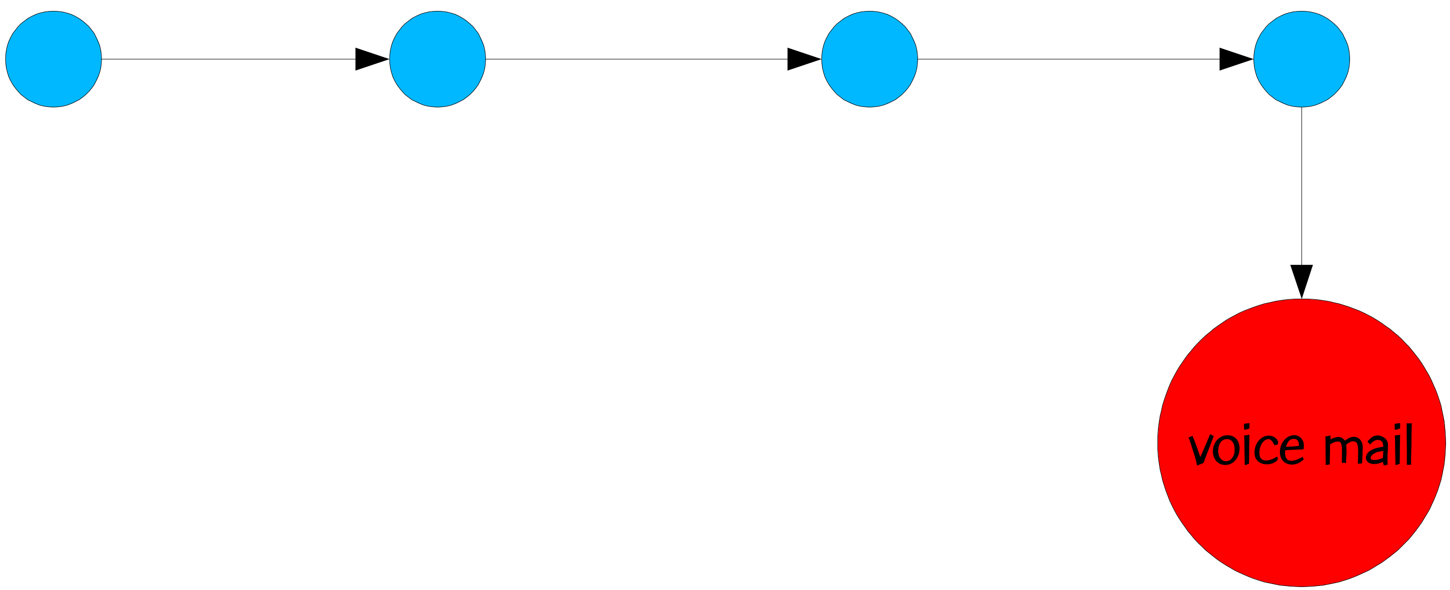
to .... next in series.



Series completion

If call not answered, it moves

to .... next in series ...

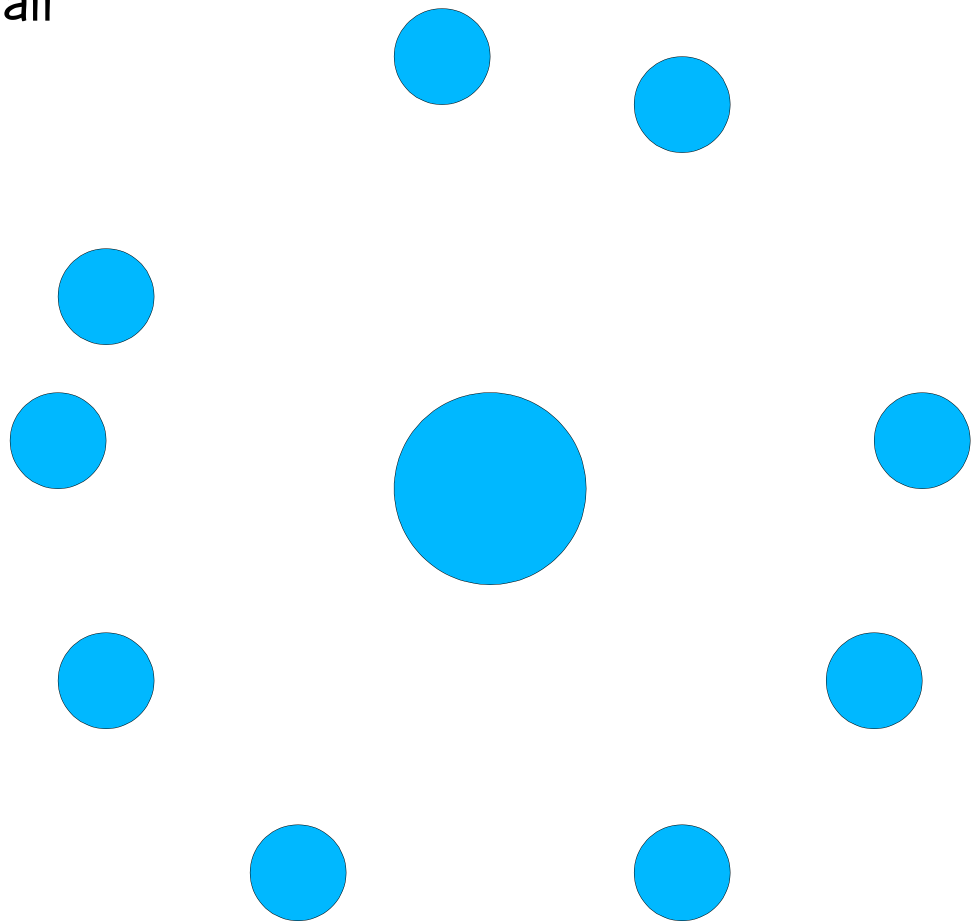


... until it is finally

answered by voice mail.

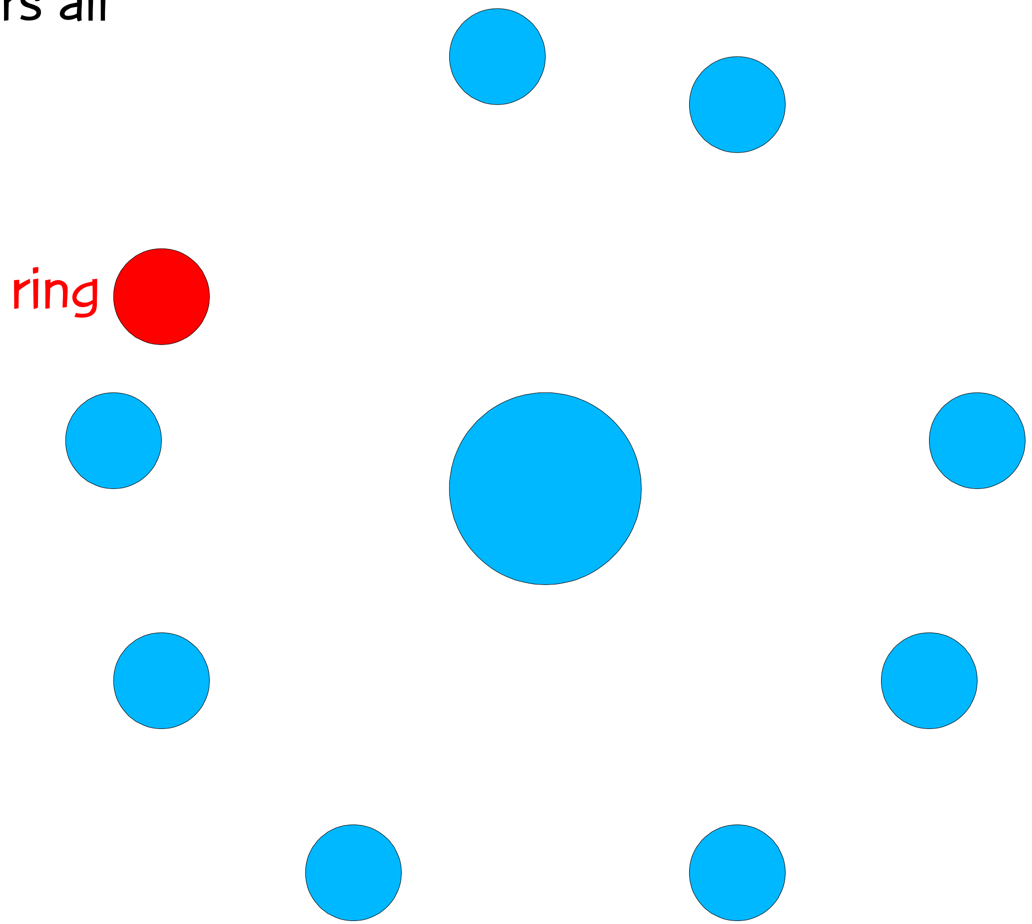
Shared TN

Assistant answers all
calls to group.



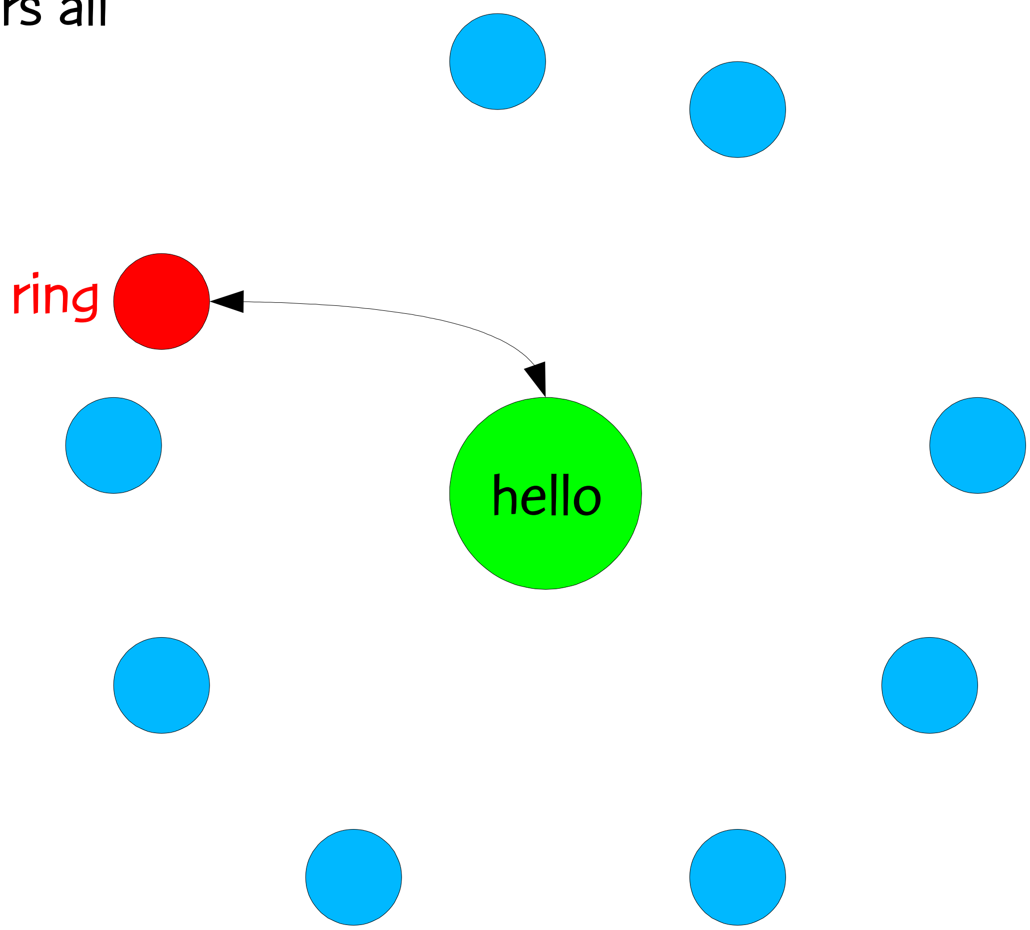
Shared TN

Assistant answers all calls to group.



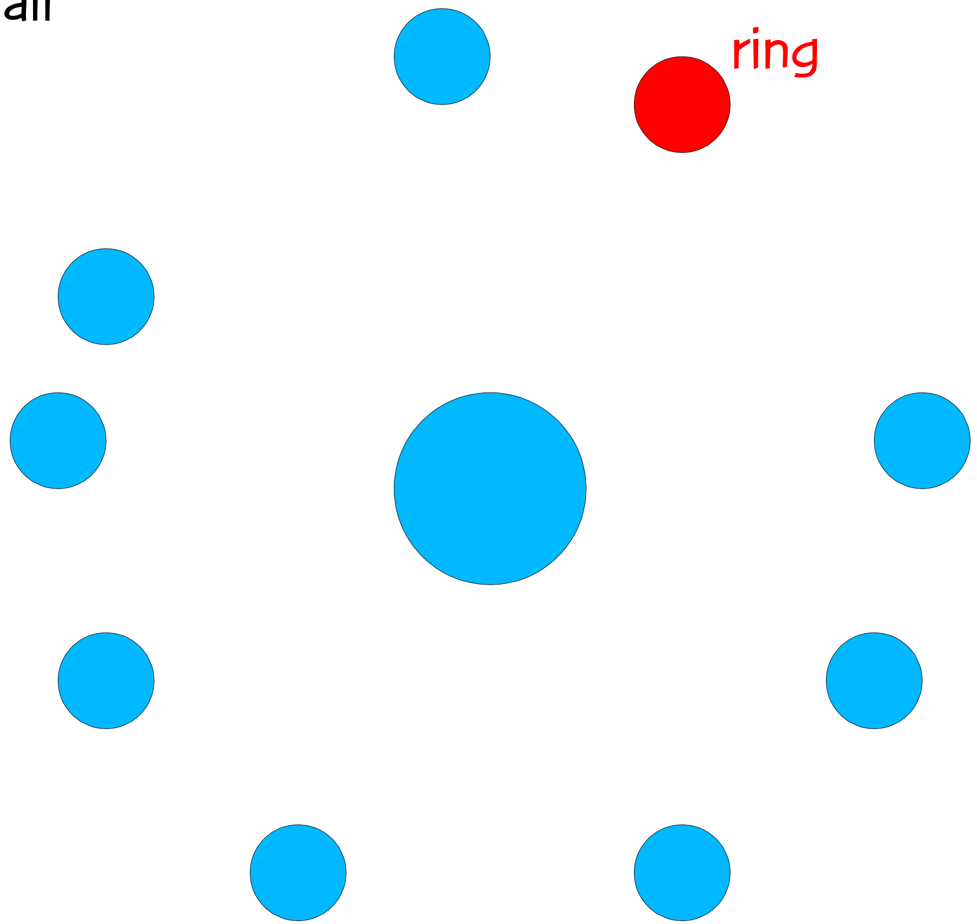
Shared TN

Assistant answers all calls to group.



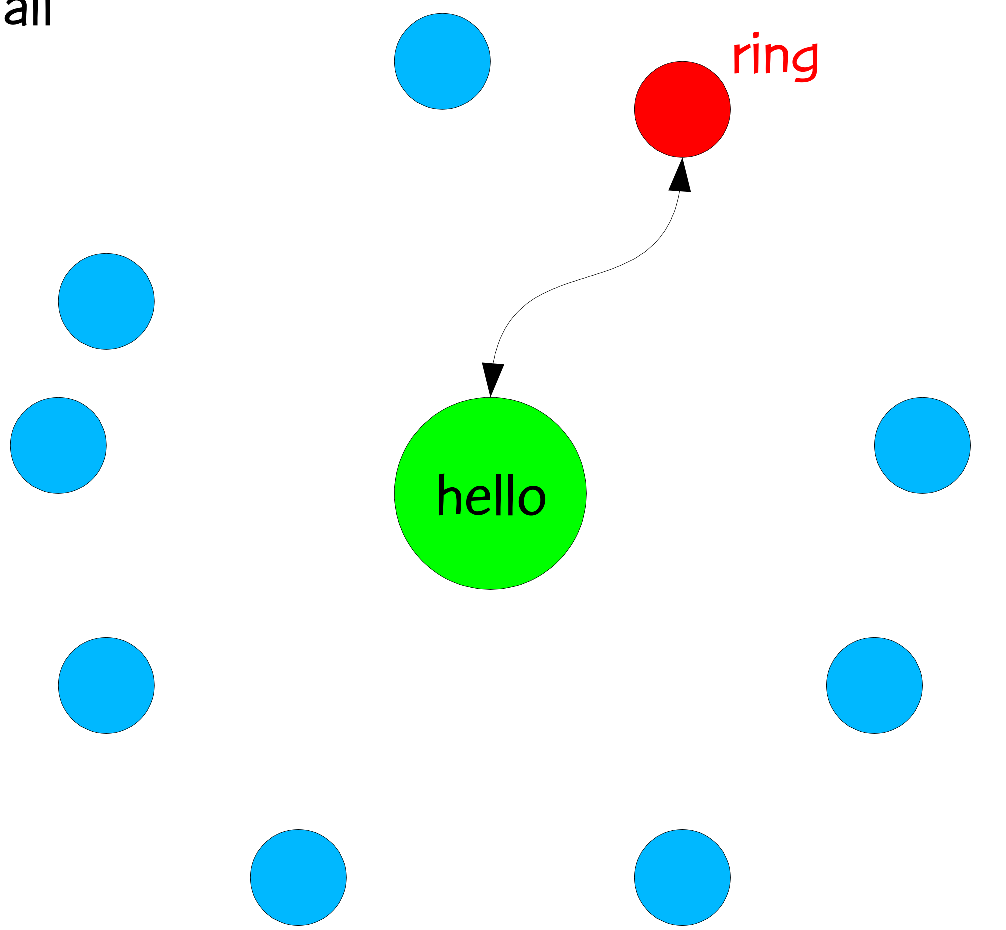
Shared TN

Assistant answers all calls to group.



Shared TN

Assistant answers all calls to group.



Real-world example

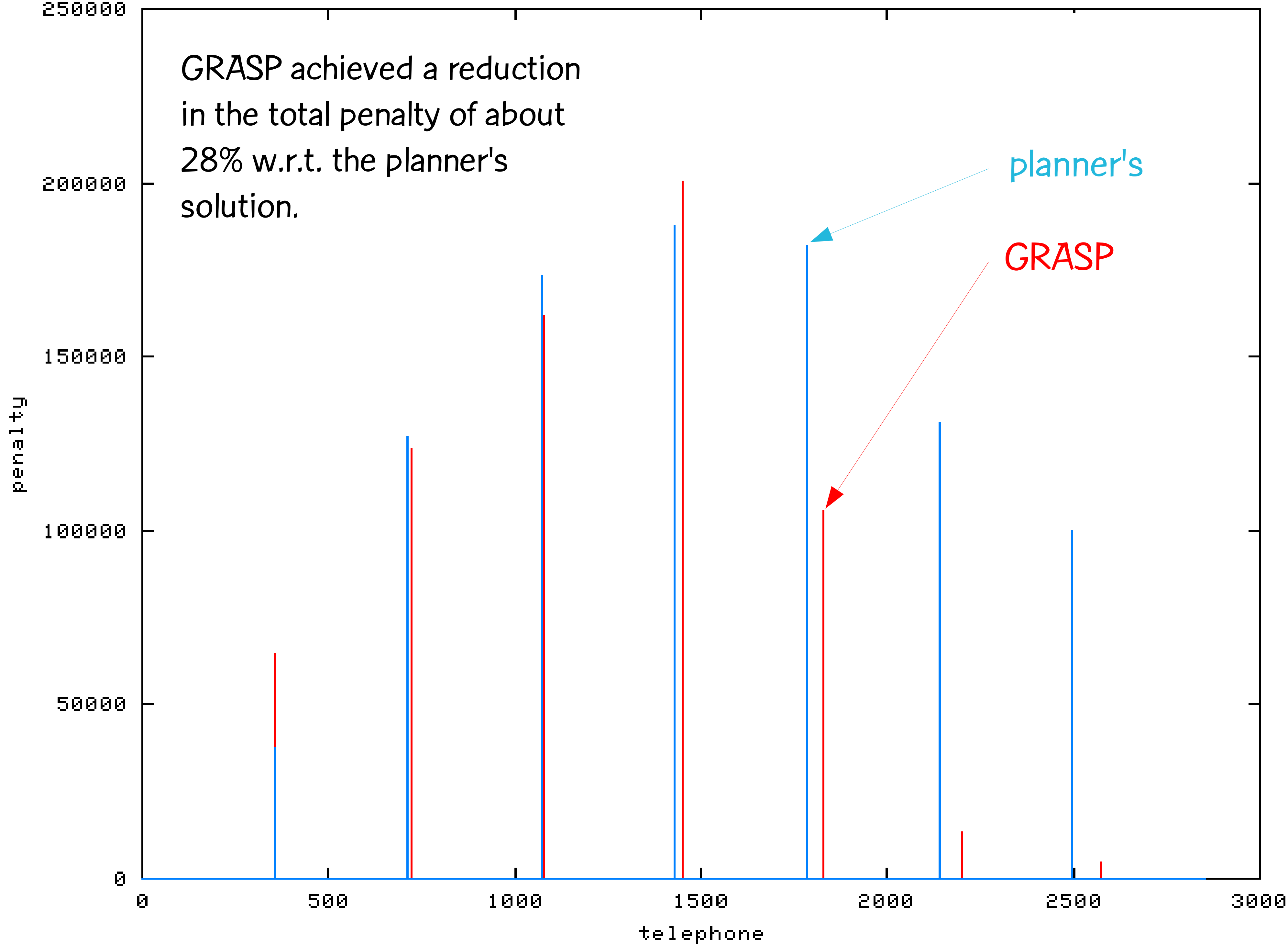
- 8 periods, 2855 phone numbers, 397 groups
- At most 375 phones can be moved in a period.
- Penalties:
  - MLHG: 10
  - CPU : 4
  - ICOM : 3
  - SC : 2
  - STN : 1

Real-world example

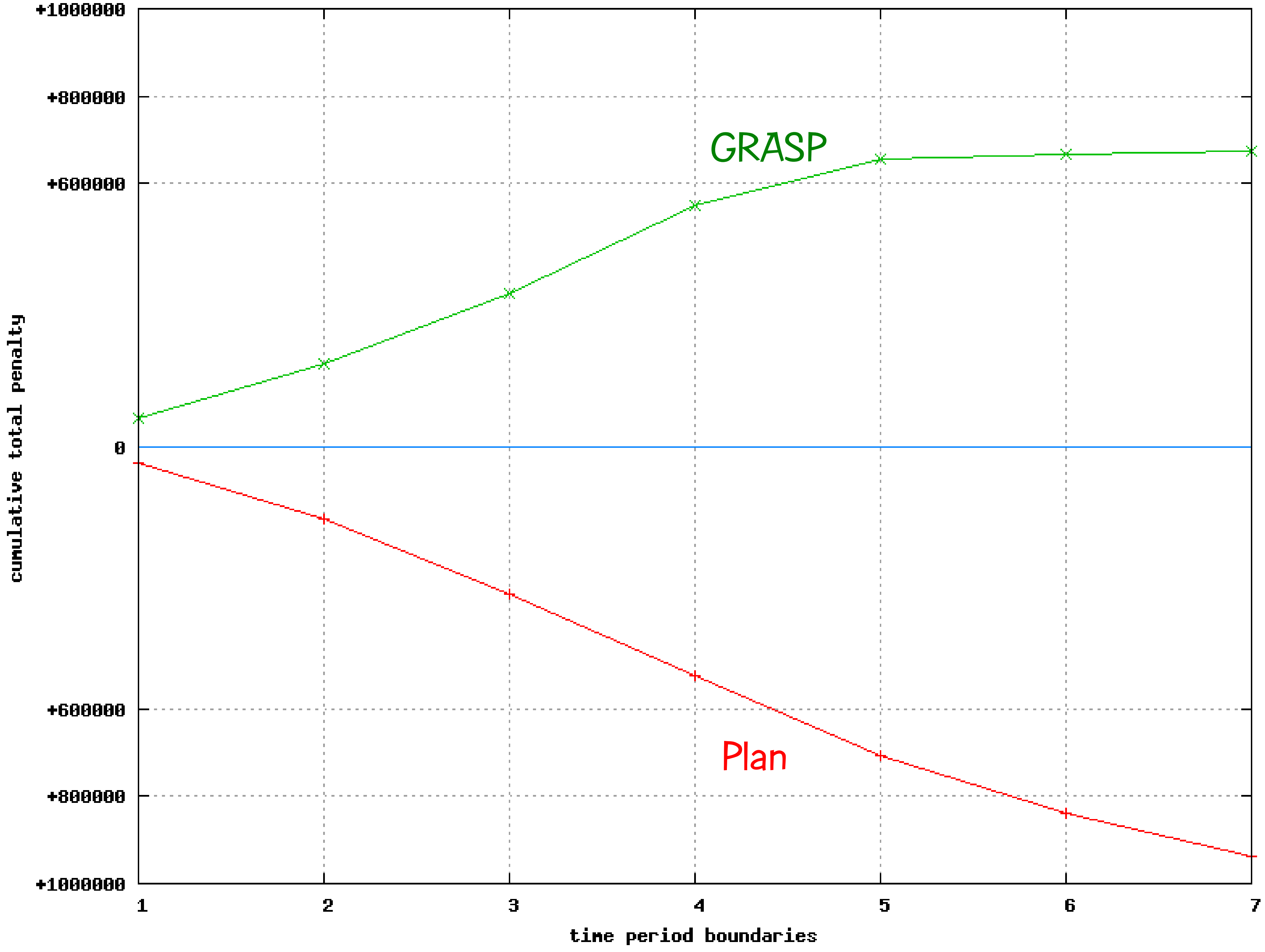
- 8 periods, 2855 phone numbers, 397 groups
- At most 375 phones can be moved in a period.
- Penalties:
  - MLHG: 10
  - CPU : 4
  - ICOM : 3
  - SC : 2
  - STN : 1

Real-world example

- 8 periods, 2855 phone numbers, 397 groups
- At most 375 phones can be moved in a period.
- Penalties:
  - MLHG: 10
  - CPU : 4
  - ICOM : 3
  - SC : 2
  - STN : 1



Your world. Delivered.



at&t
Your world. Delivered.

Concluding remarks

- A special case of batch scheduling of multi-grouped units is a network migration problem that arises when traffic is migrated from an "old" (switched) network to a "new" (IP) network.
- One needs to determine the order in which switches are decommissioned so as to minimize the temporary capacity needed to carry traffic between the "old" network and the "new" network.
- Aiex, Andrade, & Resende (2006) "A parallel GRASP with path-relinking for linear arrangement."

Concluding remarks

- A special case of batch scheduling of multi-grouped units is a network migration problem that arises when traffic is migrated from an "old" (switched) network to a "new" (IP) network.
- One needs to determine the order in which switches are decommissioned so as to minimize the temporary capacity needed to carry traffic between the "old" network and the "new" network.
- Aiex, Andrade, & Resende (2006) "A parallel GRASP with path-relinking for linear arrangement."

at&t
Your world. Delivered.

Concluding remarks

- A special case of batch scheduling of multi-grouped units is a network migration problem that arises when traffic is migrated from an "old" (switched) network to a "new" (IP) network.
- One needs to determine the order in which switches are decommissioned so as to minimize the temporary capacity needed to carry traffic between the "old" network and the "new" network.
- Aiex, Andrade, & Resende (2006) "A parallel GRASP with path-relinking for linear arrangement."

My coauthor

Diogo V. Andrade
Rutgers University

The End