Mauricio G. C. Resende
Algorithms & Optimization Research
AT&T Labs Research, Florham Park

4th MIC–Porto, July 18, 2001

GRASP with path-relinking and VNS for MAXCUT

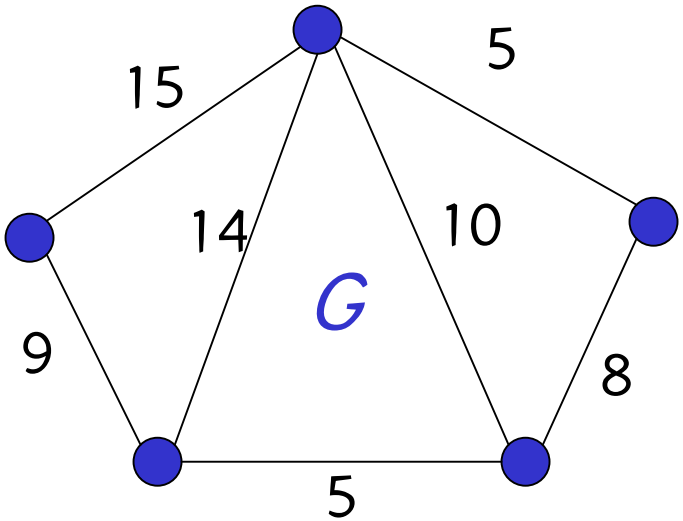
MAXCUT

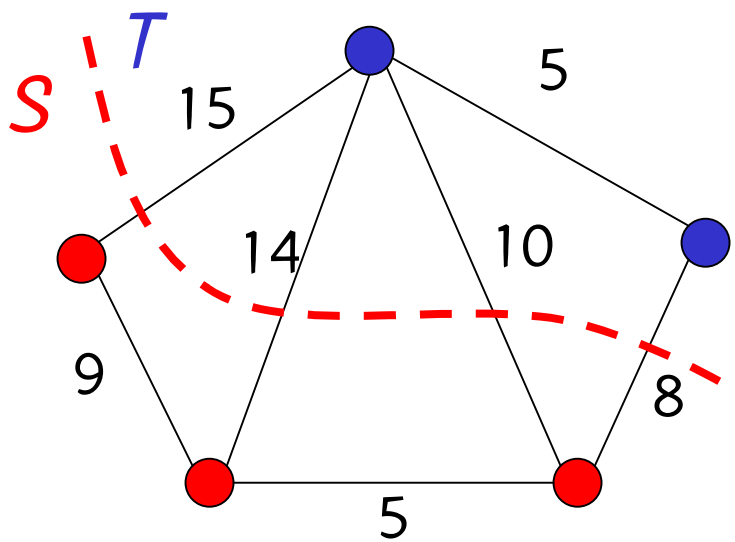
Given an undirected graph $G = (V, E)$ with weights $w_1, \ldots, w_m$ on the edges, find a vertex partition S, T such that the sum of the weights in the cut (S,T)

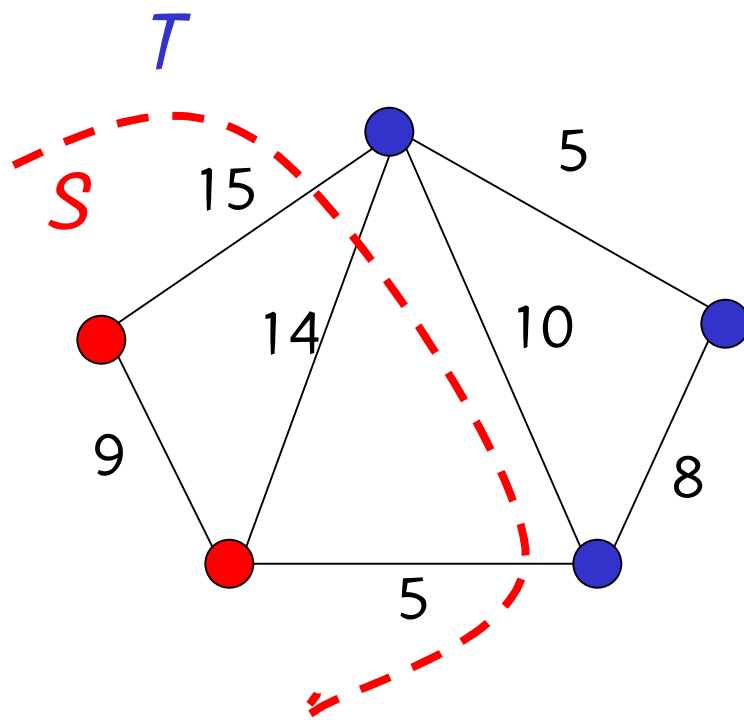
$$w(S,T) = \sum_{(i,j) \in E \ni i \in S, j \in T} w_{i,j}$$

is maximized.

MAXCUT







MAXCUT

- NP-hard (Karp, 1972) and remains NP-hard for unweighted version, i.e. with unit weights.
- .878-opt approximation algorithm of Goemans and Williamson (1995) based on semidefinite programming
- Many applications, including
  - VLSI design
  - Statistical physics
- Success claimed by semidefinite programming research community on approximation algorithms for MAXCUT.
- Problem has not been investigated much by the metaheuristics research community.

Outline

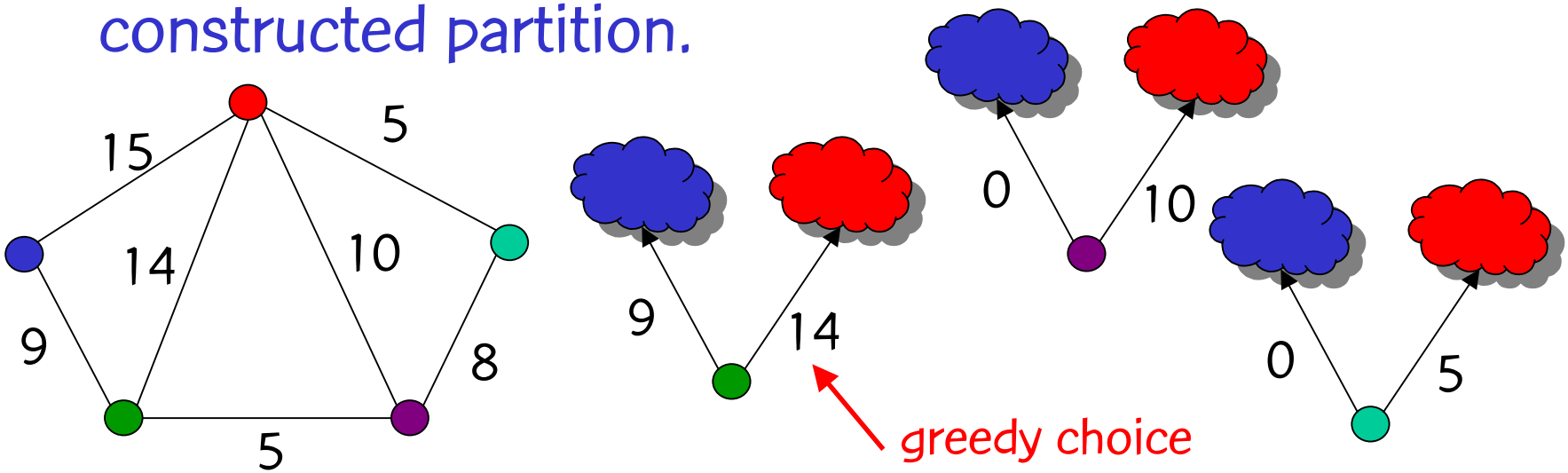
- Joint work with Paola Festa, Panos Pardalos, and Celso Ribeiro
- GRASP
- GRASP with path-relinking
- Variable neighborhood search (VNS)
- Other variants
- Preliminary computational study

GRASP for MAXCUT

- Multi-start procedure where each iteration consists of:
  - Construction of a greedy randomized feasible solution
  - Local search, starting from the constructed solution, produces a locally optimal solution

GRASP construction

- Initial edge is biased by its weight.
- Then, vertices are added, one at a time, biased by sum of weights of its edges incident to constructed partition.



Restricted candidate list (RCL) mechanism

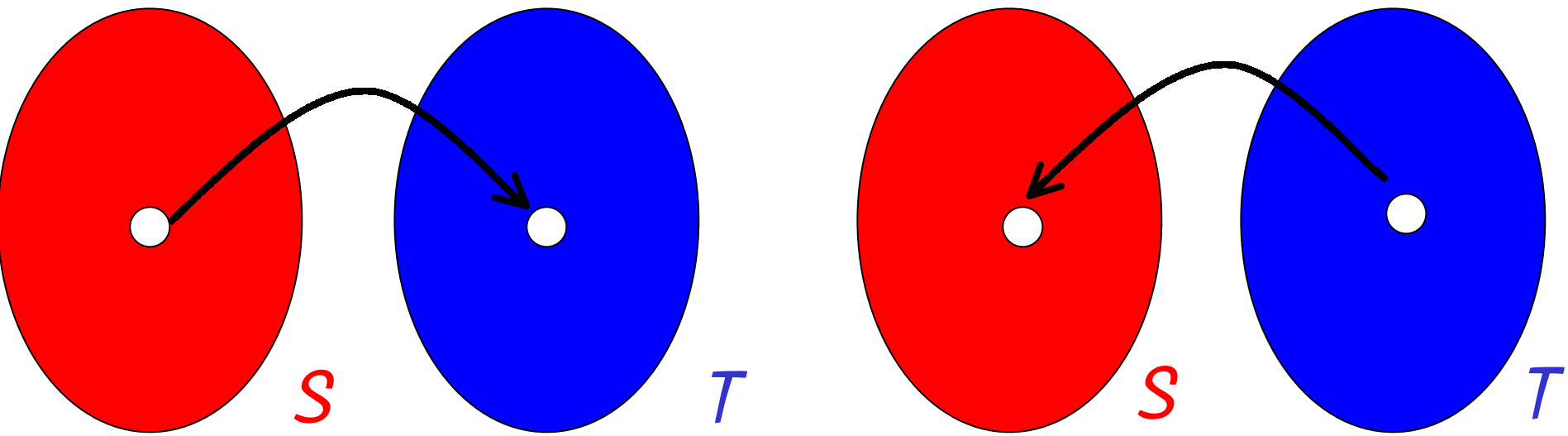
- Let $\sigma(v, S)$ and $\sigma(v, T)$ be the sum of edge weights between vertex v and partitions S and T, respectively.
- $\sigma^+ = \max \{\sigma(v, S), \sigma(v, T) \mid v \notin S \cup T\}$
- $\sigma^- = \min \{\sigma(v, S), \sigma(v, T) \mid v \notin S \cup T\}$
- RCL= $\{v \notin S \cup T \mid$

$$\sigma(v, S), \sigma(v, T) \geq \sigma^- + \alpha(\sigma^+ - \sigma^-)$$

$0 \leq \alpha \leq 1$

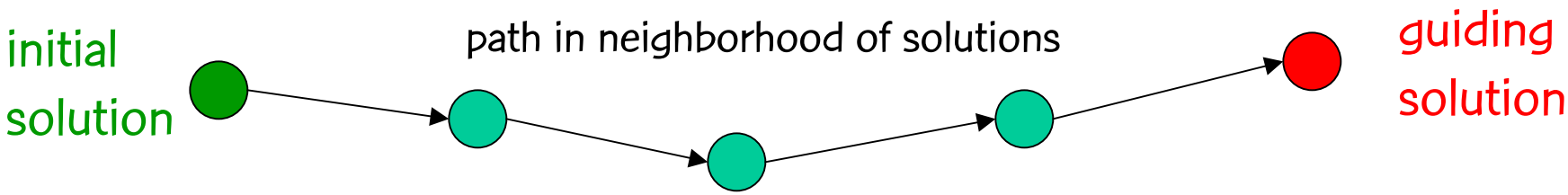
Local search

Neighborhood consists of all moves that change the partition of a single vertex.



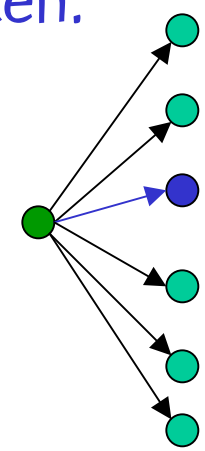
Path relinking

- Introduced in context of tabu search by Glover (1996):
  - Approach to integrate intensification & diversification in search.
- Consists in exploring trajectories that connect high quality solutions.



Path relinking

- Path is generated by selecting moves that introduce in the initial solution attributes of the guiding solution.
- At each step, all moves that incorporate attributes of the guiding solution are analyzed and best move is taken.

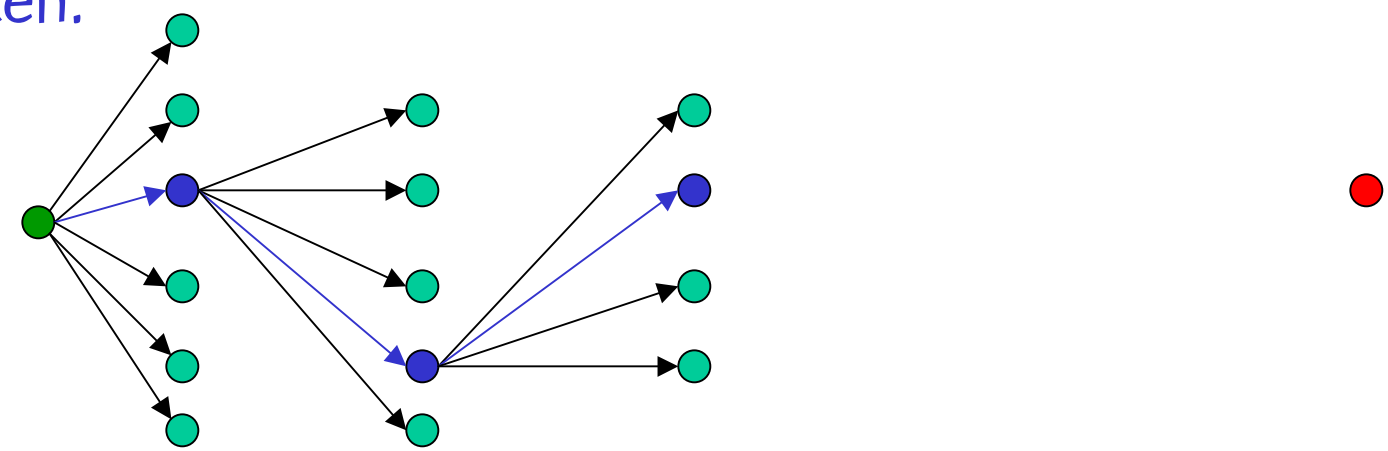


Path relinking

- Path is generated by selecting moves that introduce in the initial solution attributes of the guiding solution.
- At each step, all moves that incorporate attributes of the guiding solution are analyzed and best move is taken.

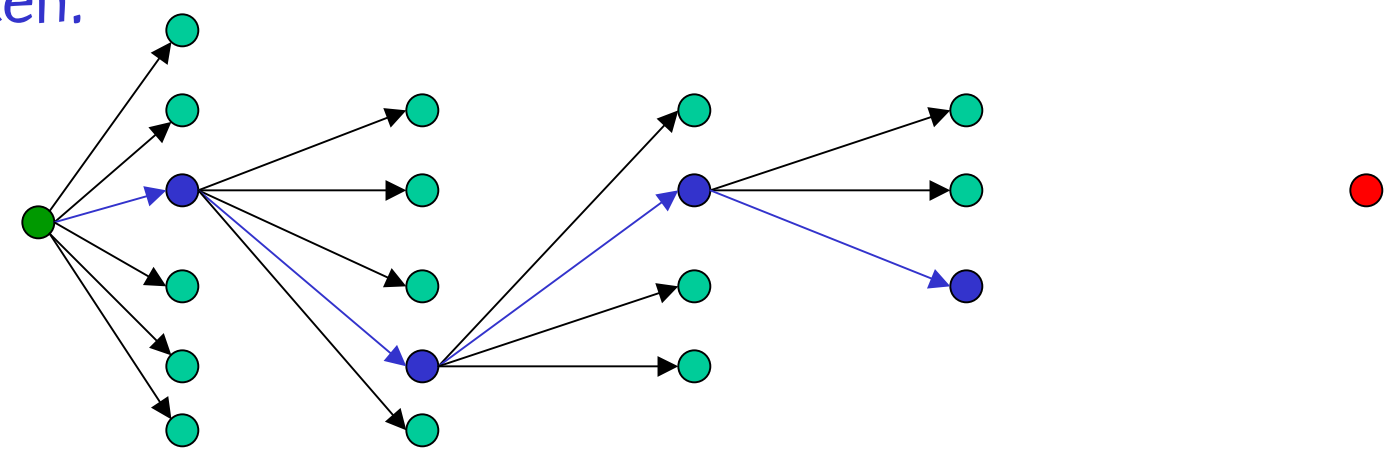
Path relinking

- Path is generated by selecting moves that introduce in the initial solution attributes of the guiding solution.
- At each step, all moves that incorporate attributes of the guiding solution are analyzed and best move is taken.



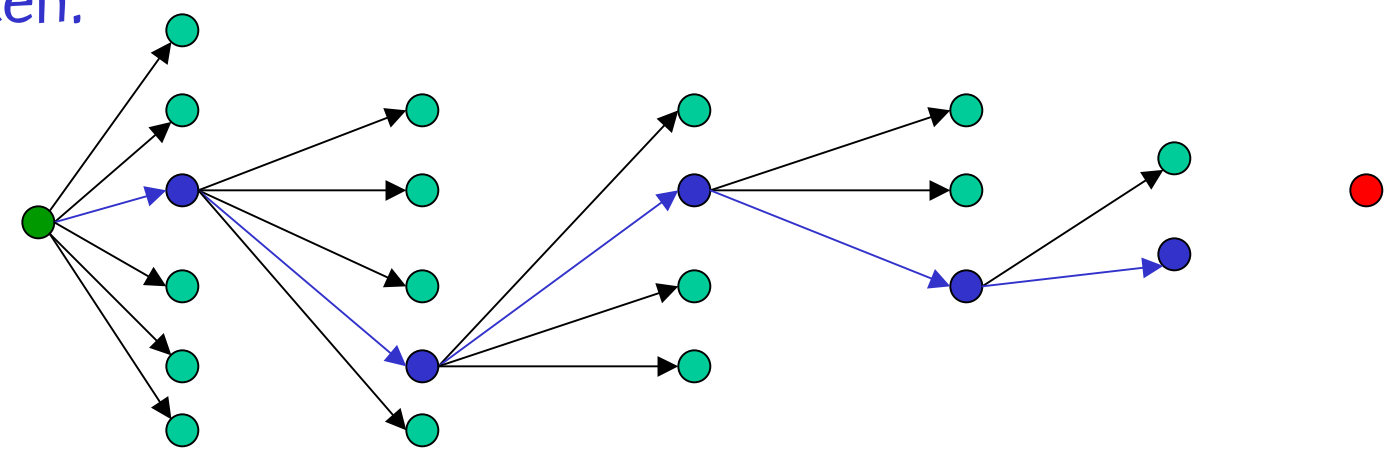
Path relinking

- Path is generated by selecting moves that introduce in the initial solution attributes of the guiding solution.
- At each step, all moves that incorporate attributes of the guiding solution are analyzed and best move is taken.



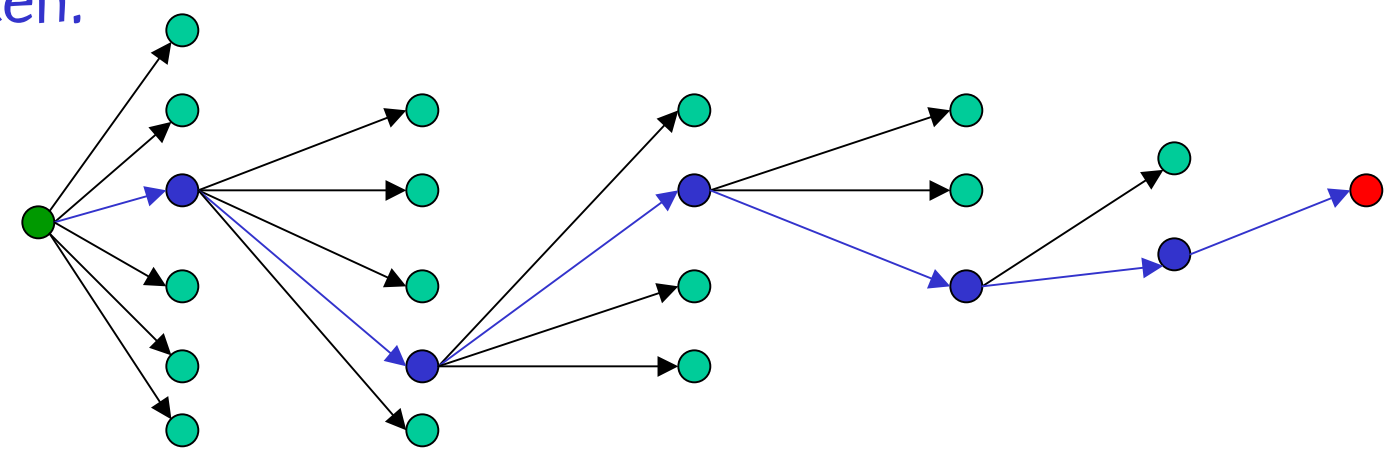
Path relinking

- Path is generated by selecting moves that introduce in the initial solution attributes of the guiding solution.
- At each step, all moves that incorporate attributes of the guiding solution are analyzed and best move is taken.



Path relinking

- Path is generated by selecting moves that introduce in the initial solution attributes of the guiding solution.
- At each step, all moves that incorporate attributes of the guiding solution are analyzed and best move is taken.



Path relinking in GRASP

- Introduced by Laguna & Martí (1999)
- Maintain an elite set of solutions found during GRASP iterations.
- After each GRASP iteration (construction & local search):
  - Select an elite solution at random: guiding solution.
  - Use GRASP solution as initial solution.
  - Do path relinking from initial solution to guiding solution.

Elite set membership rule

(Fleurent & Glover, 1999)

- Let x be a candidate for inclusion into the Elite set. x is accepted into the set if:
  - x is better fit than the best elite set solution
  - x is not as fit as the best elite set solution, but better fit than the worst and sufficiently different from all elite solutions
- If x is accepted, it replaces the worst elite set member.

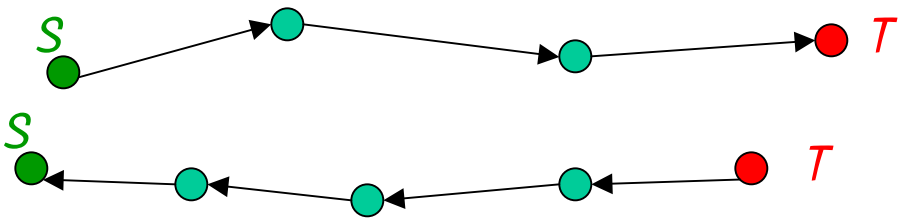
Path relinking: Intensification & post-optimization

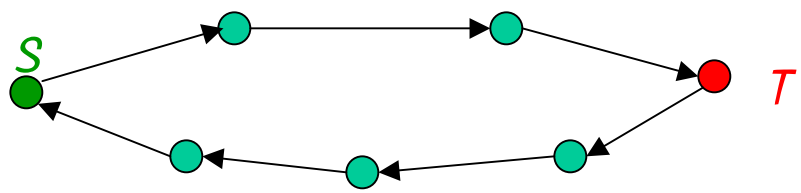
(Aiex, R., Pardalos, & Toraldo, 2001)

- Elite set intensification (periodically or as post-optimization phase):
  - Apply path relinking between all pairs of elite set solutions.
  - Update elite set, if necessary, and repeat until no change occurs.
- If done as post-optimization:
  - Apply local search to each elite set solution.
  - Repeat if necessary.

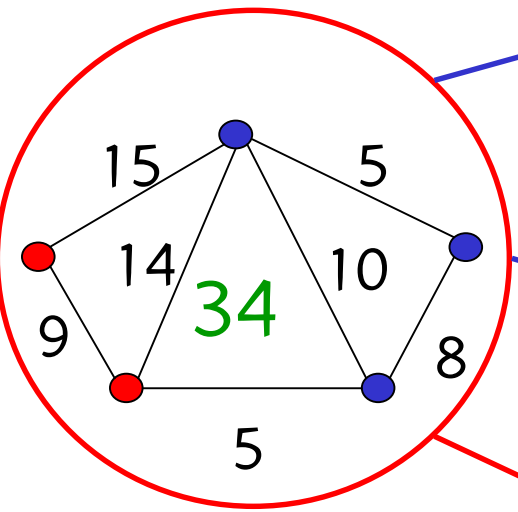
Path relinking: Variants

- How targets are chosen:
  - Select a subset of targets $\underline{P} \subseteq P$ from elite set.
  - Examples: $|\underline{P}| = 1$ and $|\underline{P}| = |P|$.
- Direction of path relinking:
  - Forward: from S to T.
  - Backward: from T to S.
  - Forward and back: from S to T, then from T to S.

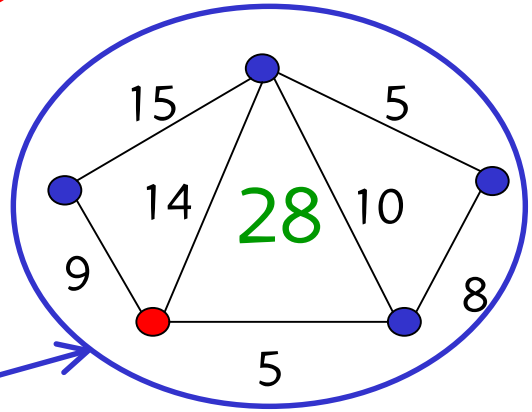


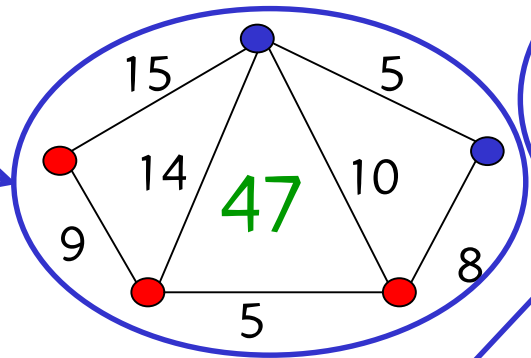


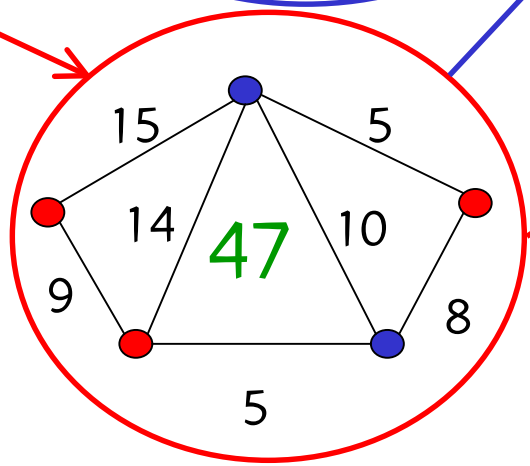
Path-relinking for MAXCUT (forward)

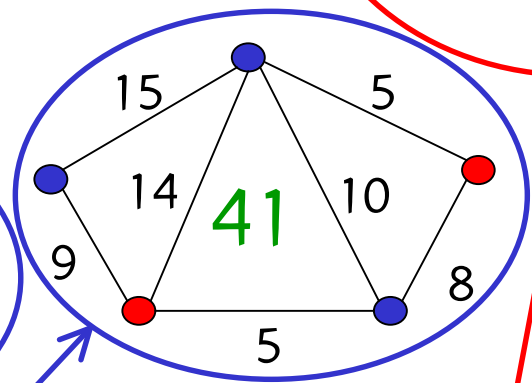


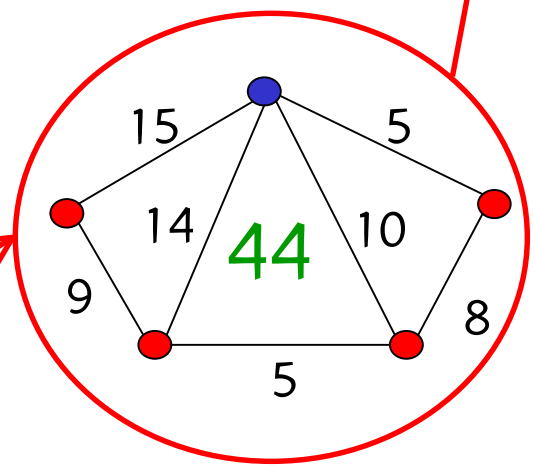
initial solution

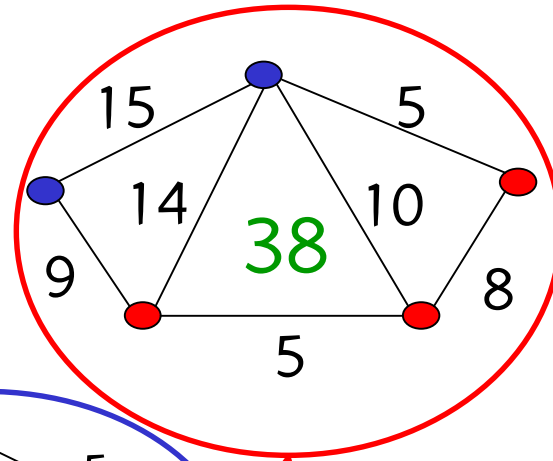








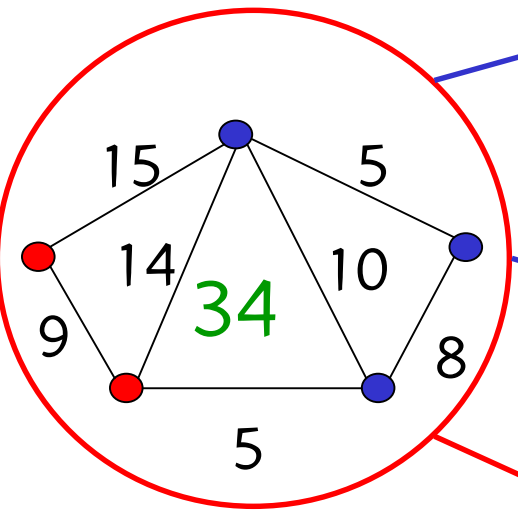




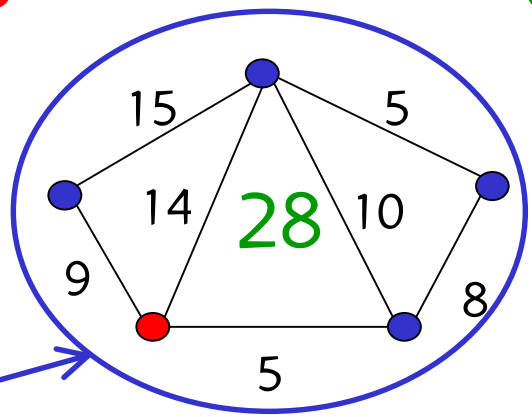
guiding solution

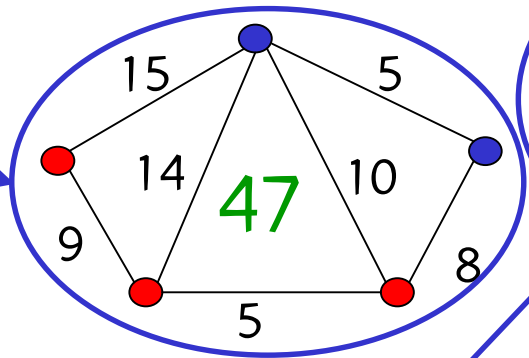
AT&T

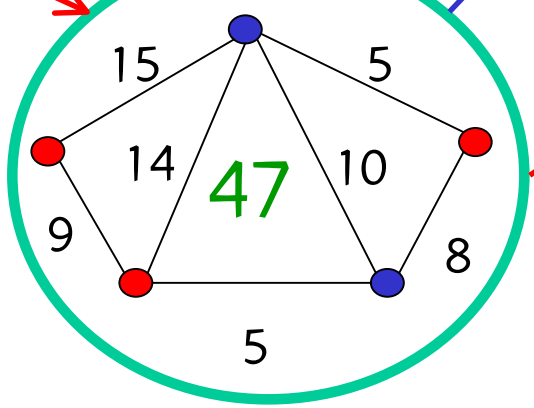
Path-relinking for MAXCUT (forward)



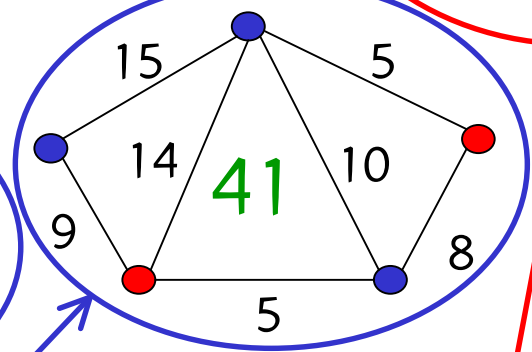
initial solution

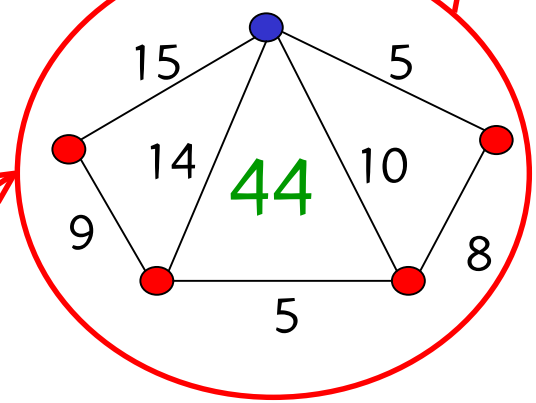


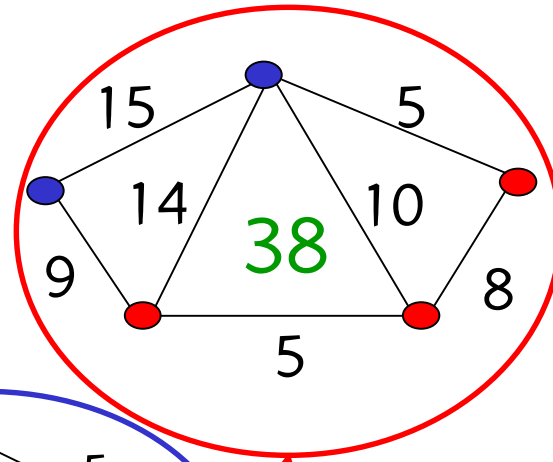




optimal







guiding solution

GRASP with path-relinking

```
repeat{
    x = ConstructGreedyRandomized ( );
    x = LocalSearch(x );
    if (EliteSet is full) x = PathRelinking(x, EliteSet );
    UpdateEliteSet(x, EliteSet );
    if (IntensificationFrequency) Intensify( EliteSet );
} until termination criterion satisfied;
PostOptimize( EliteSet );
```

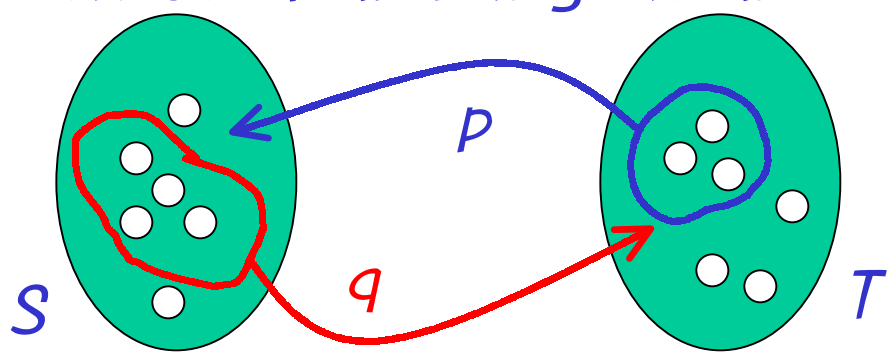
VNS for MAXCUT

Let x represent a MAXCUT solution, i.e.

$x_i = 1$ if $i \in S$,

$x_i = 0$ if $i \in T$.

The k-th order neighborhood $N^k(x)$ consists of all solutions x' whose Hamming distance from x is exactly k.



$p + q = k$

VNS and GRASP + VNS for MAXCUT

```
for t = 1,...,maxIterations {
    x = GenerateInitialSolution( );
    k = 1;
    for k ≤ k_max {
        randomly generate x' ∈ N^k(x) ;
        x'' = LocalSearch (x');
        if ( f(x'') > f(x) ){
            x = x'';
            k = 1;
        }
        else k = k+1;
    }
}
```

ConstructRandom( );

VNS

ConstructGreedyRandomized( );

GRASP + VNS

Computational experiments

- Compare to results obtained by semidefinite programming research community (.878-opt approximation algorithm)
- Compare variants:
  - GRASP
  - GRASP with path-relinking
  - VNS
  - GRASP with VNS

Test problems

- Used by semidefinite programming research community
- Type I: Generated by Helmberg & Rendl (1997) with a network generator written by G. Rinaldi
- Type II: Sparse randomly generated instances by Fujisawa et al. (1999)
- Type III: Spin glass instances from the 7th DIMACS Implementation Challenge by Jünger & Liers.

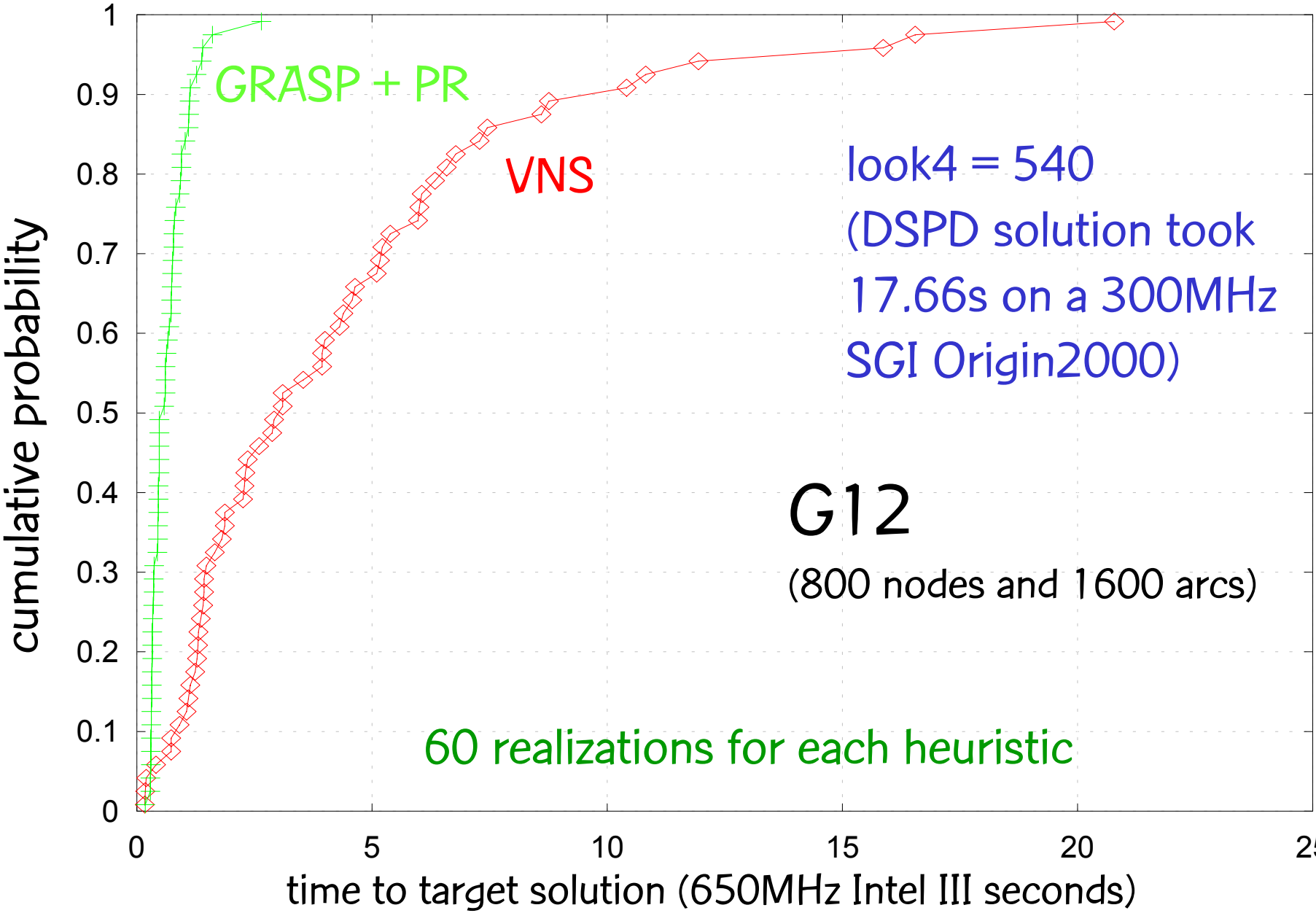
Helmberg & Rendl instances

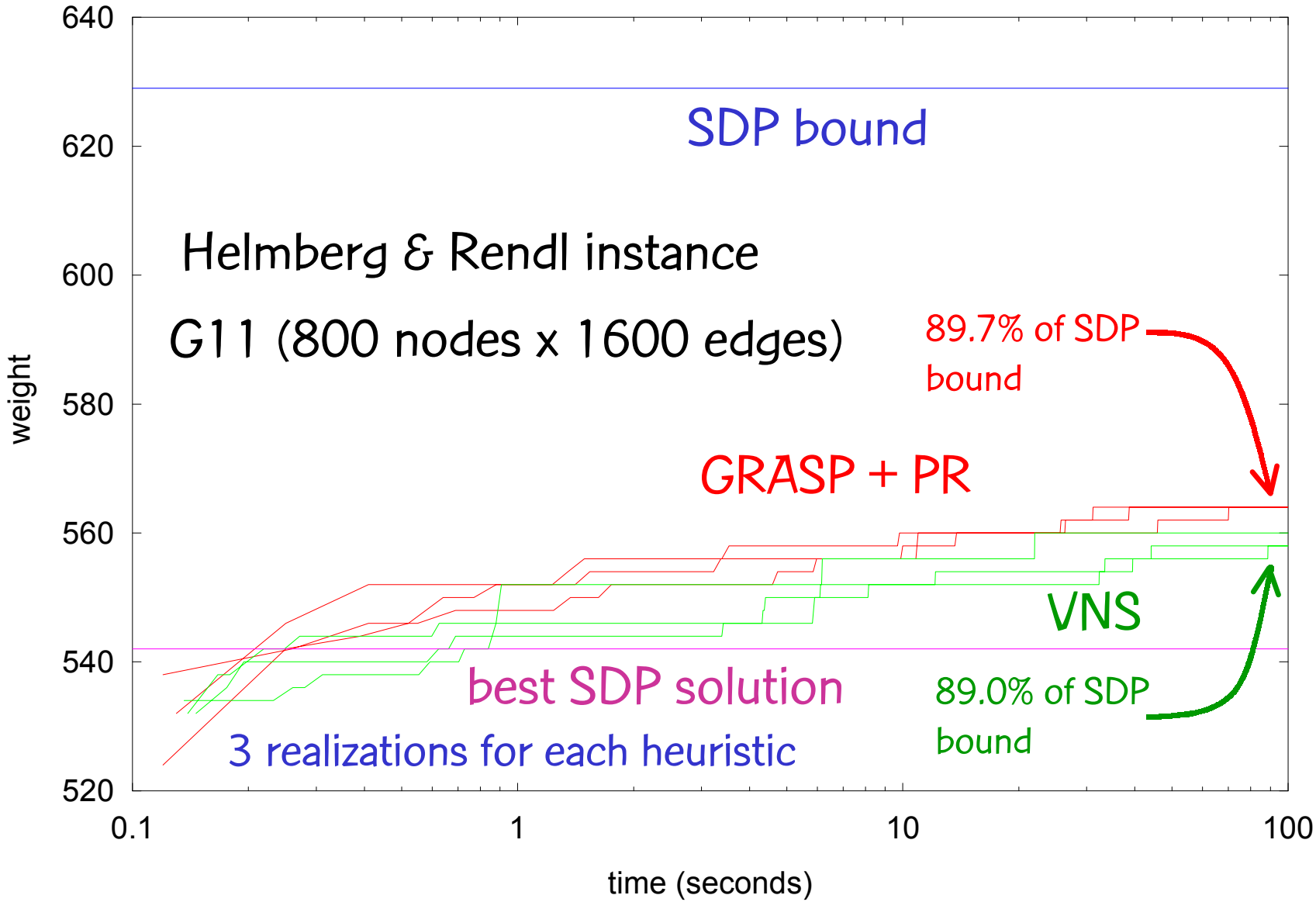
Benson, Ye, Zhang (2000)

Burer, Monteiro, Zhang (2000)

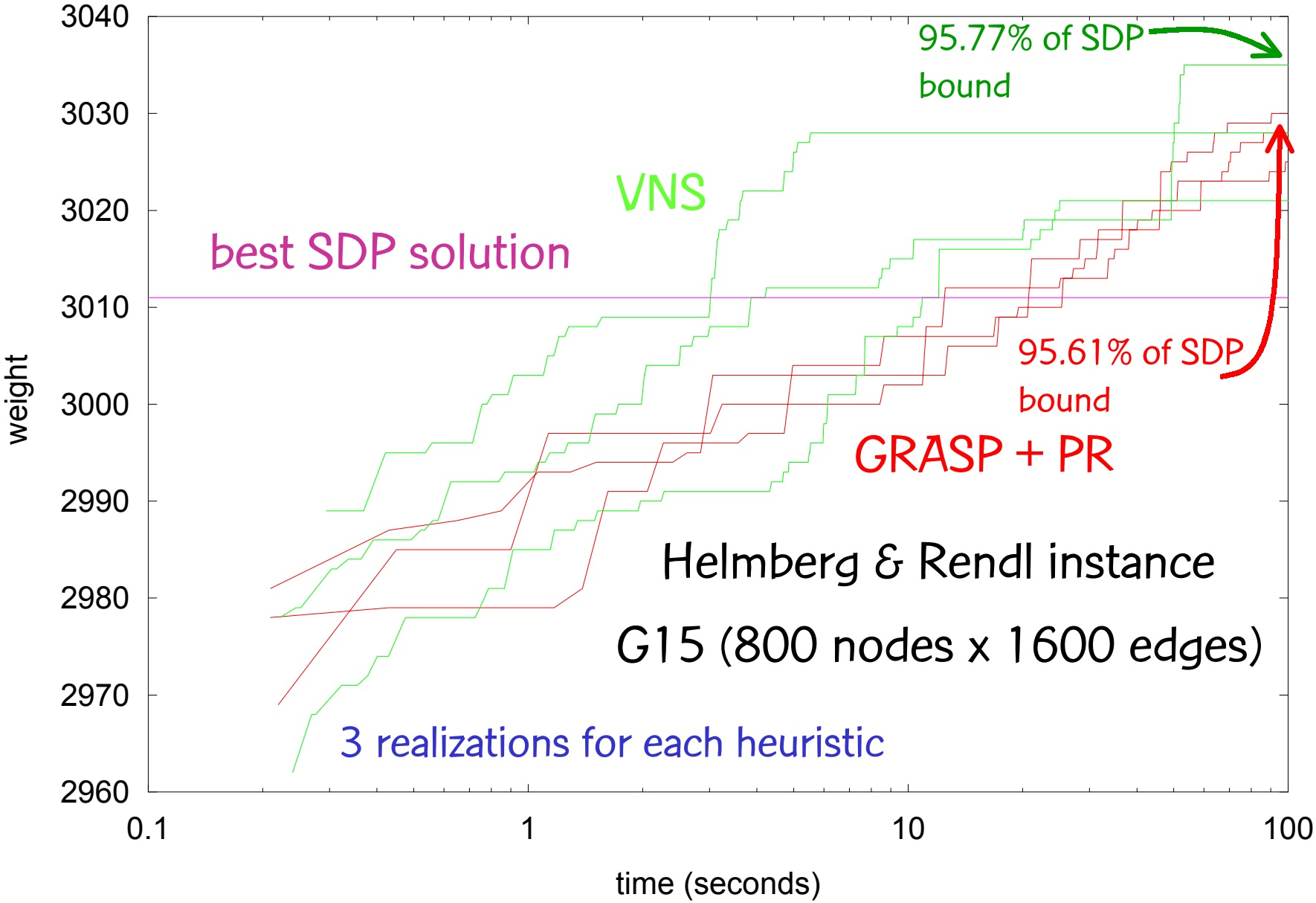
prob	size	SDP UB	DSDP	CirCut	G+PR	VNS
G11	800 x 1600	627.04	.8644	.8357	.8995	.8867
G12	800 x 1600	621.61	.8687	.8237	.8945	.8848
G23	2000 x 19990	14041.3	.9263	.9399	.9358	.9397
G32	2000 x 4000	1560.75	.8573	.8367	.8880	.8816
G34	2000 x 4000	1541.66	.8653	.8277	.8835	.8744
G48	3000 x 6000	6000	1.000	.9460	1.000	1.000

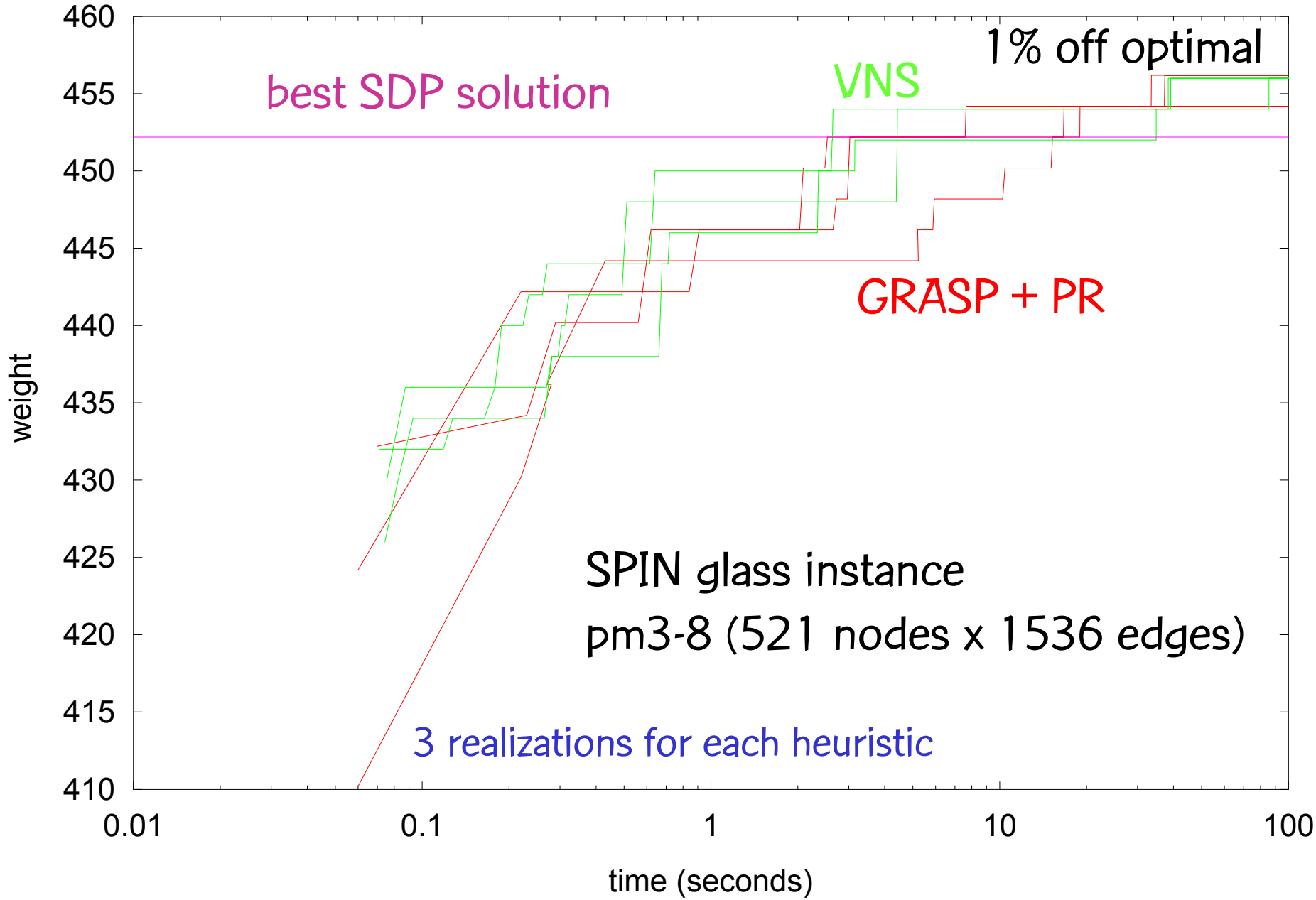
G+PR & VNS run100s on 650MHz Pentium III

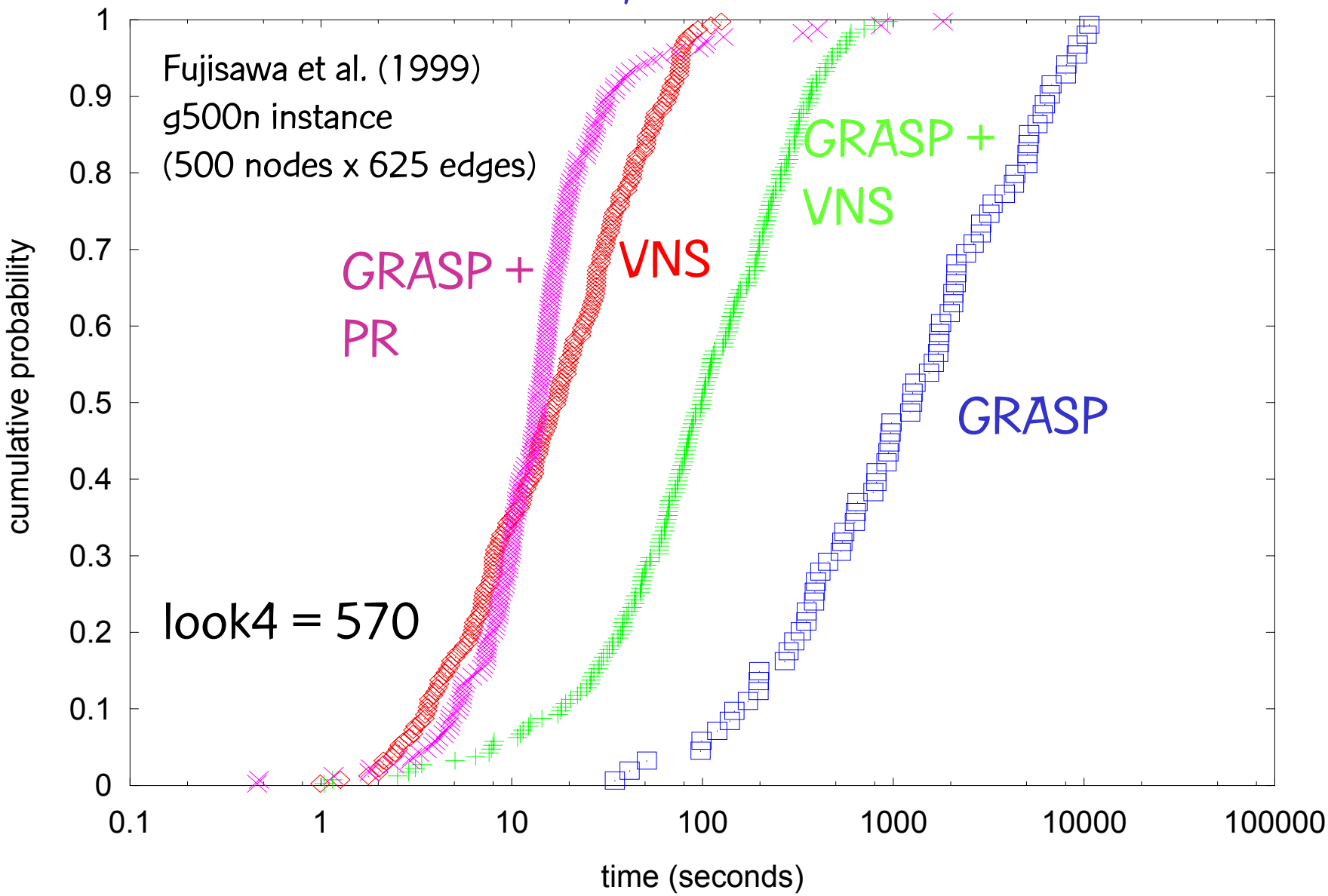

GRASP + PR
VNS
look4 = 540
(DSPD solution took
17.66s on a 300MHz
SGI Origin2000)
G12
(800 nodes and 1600 arcs)
60 realizations for each heuristic
cumulative probability
time to target solution (650MHz Intel III seconds)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25


640
620
600
580
560
540
520
weight
0.1
1
10
100
time (seconds)
SDP bound
Helmberg & Rendl instance
G11 (800 nodes x 1600 edges)
89.7% of SDP bound
GRASP + PR
VNS
best SDP solution
89.0% of SDP bound
3 realizations for each heuristic

AT&T


3040
3030
3020
3010
3000
2990
2980
2970
2960
weight
0.1
1
10
100
time (seconds)
95.77% of SDP
bound
VNS
best SDP solution
95.61% of SDP
bound
GRASP + PR
Helmberg & Rendl instance
G15 (800 nodes x 1600 edges)
3 realizations for each heuristic


1% off optimal
best SDP solution
VNS
GRASP + PR
SPIN glass instance
pm3-8 (521 nodes x 1536 edges)
3 realizations for each heuristic
weight
time (seconds)
460
455
450
445
440
435
430
425
420
415
410
0.01
0.1
1
10
100


77 realizations for GRASP; 200 for others
Fujisawa et al. (1999)
g500n instance
(500 nodes x 625 edges)
GRASP + PR
VNS
GRASP + VNS
GRASP
look4 = 570
cumulative probability
time (seconds)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.1
1
10
100
1000
10000
100000

Concluding remarks

- Metaheuristics appear to be a challenge for MAXCUT algorithms based on semidefinite programming
- Full paper will have extensive experimental results using
  - the variants described in this talk
  - other hybrids, e.g. GRASP+VNS with path-relinking and VNS with path-relinking
- These slides are available at http://www.research.att.com/~mgcr