A hybrid Lagrangean heuristic with GRASP applied to set multicovering

Luciana Pessôa[1]

Celso C. Ribeiro[1]

Maurício G.C. Resende[2]

[1] Department of Computer Science, Universidade Federal Fluminense, Brazil.

[2] AT&T Research Labs, USA.

Workshop TRANSLOG

Reñaca, Chile

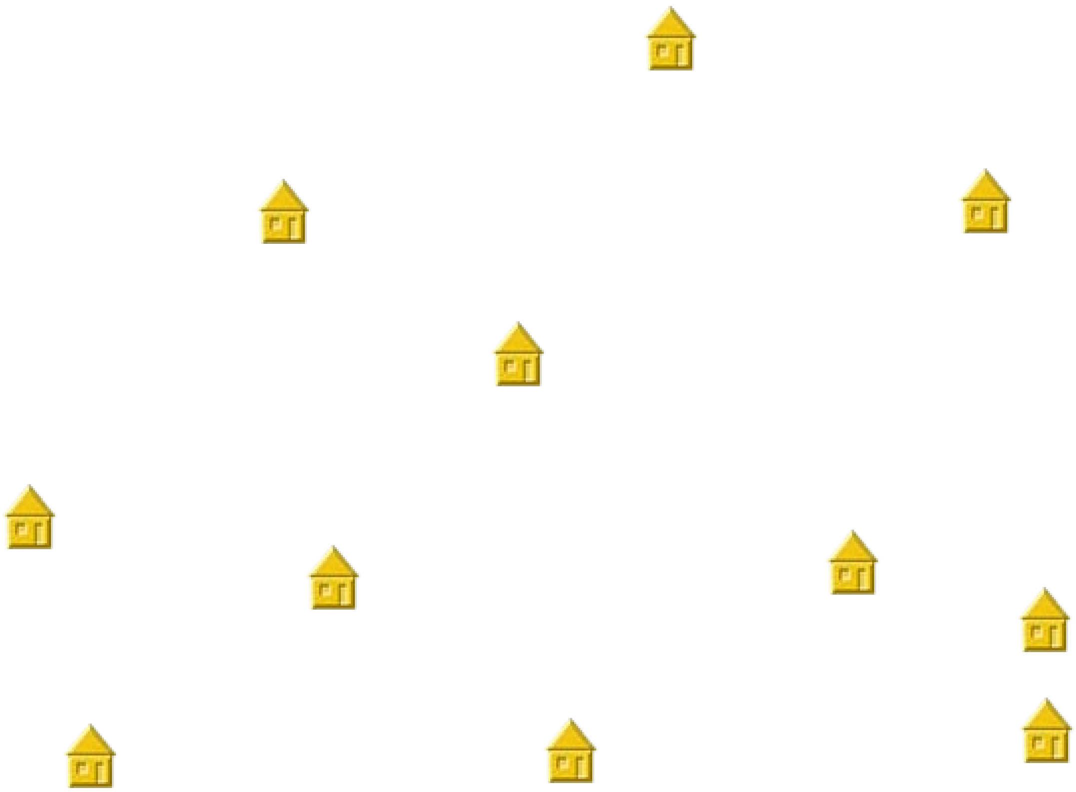
December 8-11, 2009

Summary

- Motivation: Redundant POP placement problem
- Set k-covering
- GRASP
- Lagrangean heuristics
  - Greedy Lagrangean heuristic
  - GRASP Lagrangean heuristic
- Experiments
- Concluding remarks

Redundant POP placement problem

- Given customers of a wireless network…



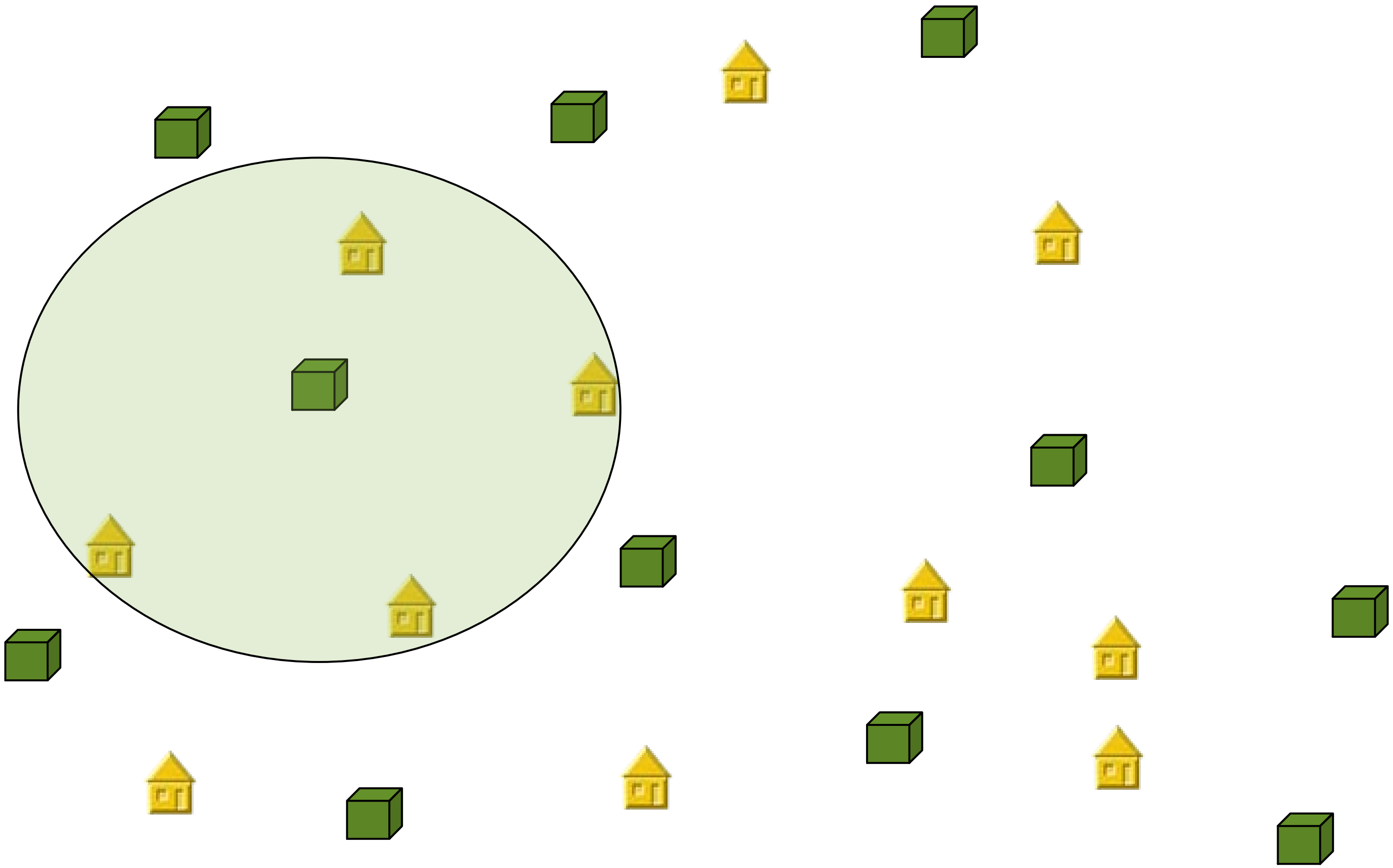
Redundant POP placement problem

- … and potential PoP locations, where an equipment can be placed.

A PoP (point of presence) may host, for example, an antenna (hubs, modens) which connects customers to the network.

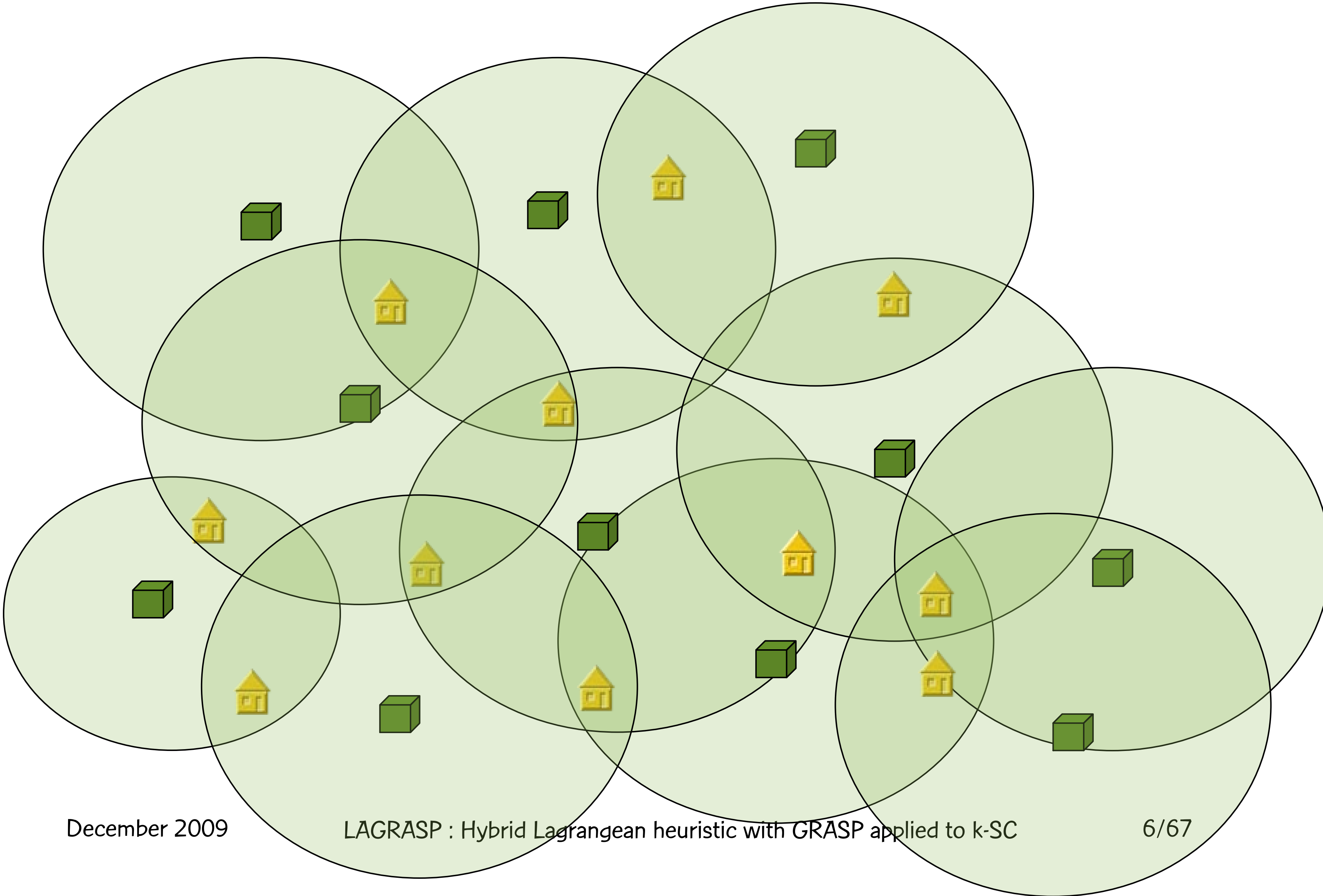
Redundant POP placement problem

- An equipment in a PoP covers some customers.



- A PoP location has a cost associated with it.

Redundant POP placement problem

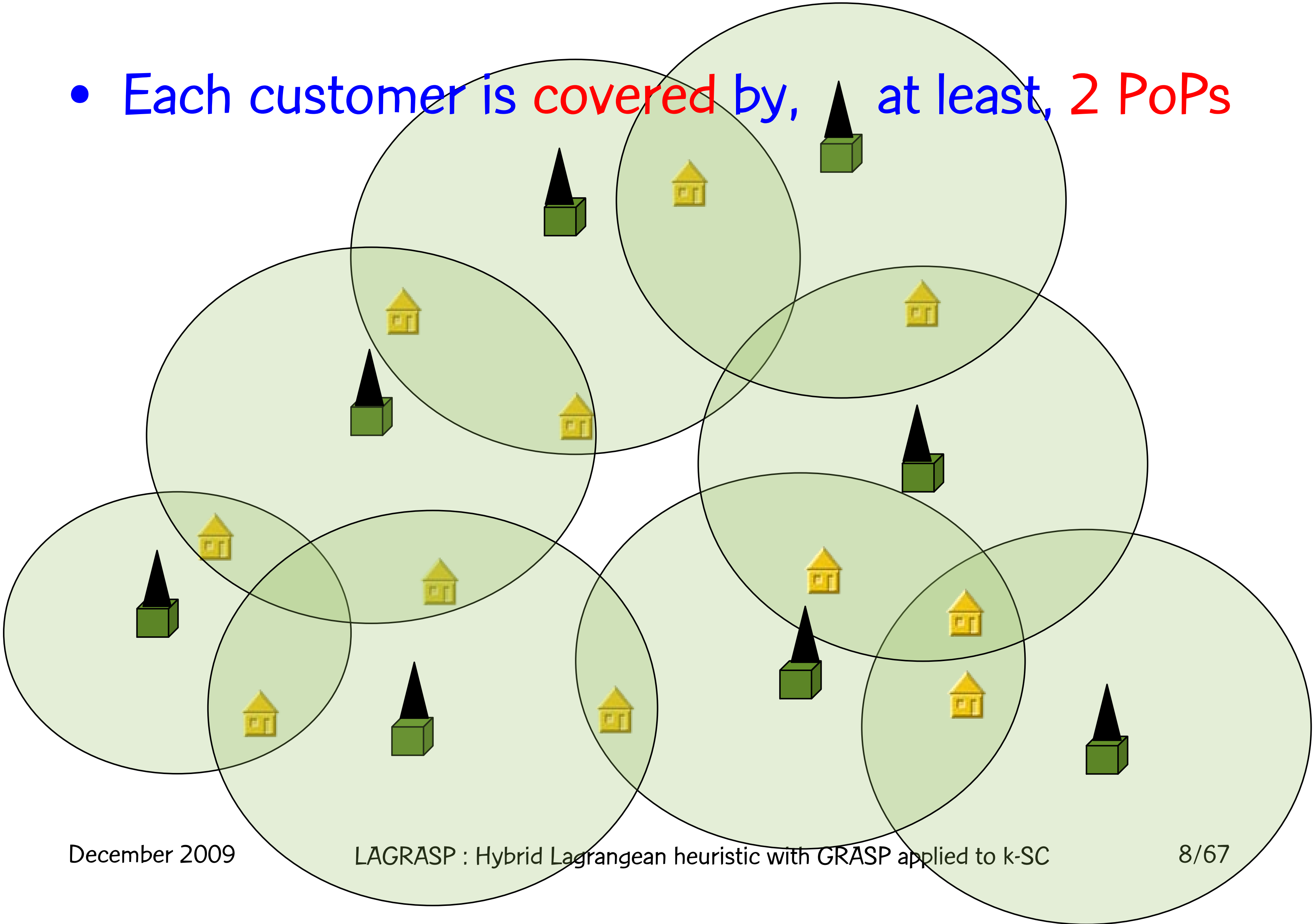


Redundant POP placement problem

- Determine in which PoPs locations to place the equipments:
  - Fault-tolerance (reliability) constraints: each customer must be covered by, at least, k antennas.
  - Minimize total PoP installation costs.

Redundant POP placement problem

- Each customer is covered by, at least, 2 PoPs



Set k-covering (multicovering)

- Mathematical formulation:

$$x_j = \begin{cases} 1, \text{ if equipment is placed in PoP location } j \\ 0, \text{ otherwise} \end{cases}$$

$$\min \sum_{j=1}^{n} c_j x_j$$

$$s.t. \sum_{j=1}^{n} a_{ij} x_j \geq k, \; i = 1, \ldots, m,$$

$$x_j \in \{0, 1\}, \; j = 1, \ldots, n.$$

k is the coverage factor

GRASP with path-relinking

- GRASP: multistart metaheuristic
  - Greedy randomized construction phase
  - Local search
- Path-relinking: memory-based intensification

GRASP with path-relinking

while .not.StoppingCriterion (max number of iterations) do:

- Build solution x with greedy randomized algorithm.
- Use local search to improve current solution x.
- Select locally optimal solution x' from elite set.
- Apply path-relinking to obtain the best solution x" in a trajectory between x and x'.
- Apply local search to improve solution x".
- Update elite set with x".
- If x" improves best solution x*, then replace x* by x".

end while

GRASP with path-relinking

- Construction phase
  - Repeat until complete solution is built:
    - Compute greedy evaluation r_j for each candidate element j
    - Rank all elements according to their greedy evaluations
    - Place well ranked elements defined by a treshold $0 \leq \alpha \leq 1$ in a restricted candidate list (RCL)
    - Select an element e from the RCL at random
    - Add selected element e to the solution

GRASP with path-relinking

- Construction phase

$X_j = 0$, for j=1,…,n;
L = {1,…,n};

  – Repeat until complete solution is built:

$r_j = c_j/cardinality_j$;

    - Compute greedy evaluation r_j for each candidate element j
    - Rank all elements according to their greedy evaluations

Identify r_{min} and r_{max};

    - Place well ranked elements defined by a treshold parameter $0 \leq \alpha \leq 1$ in a restricted candidate list (RCL)

$RCL = \{j \notin L \mid r_j \leq r_{min} + \alpha (r_{max} - r_{min})\}$;

    - Select an element e from the RCL at random
    - Add selected element e to the solution

Select e;
$X_e = 1$;
L=L\{e};

GRASP with path-relinking

- Local search:
  - There is no guarantee that constructed solutions are locally optimal, even with respect to simple neighborhood definitions.
  - Local search explores the neighborhood of a solution, looking for a cost-improving solution
  - (k,p)-exchange: exchange k elements in the solution by p elements not in the solution.

GRASP with path-relinking

- Local search:
  - Neighborhood (k,p)-exchange: exchange k elements in the solution by p elements not in the solution.
  - Starting from a solution x
  - Do
    - x ← (1,0)-exchange(x)

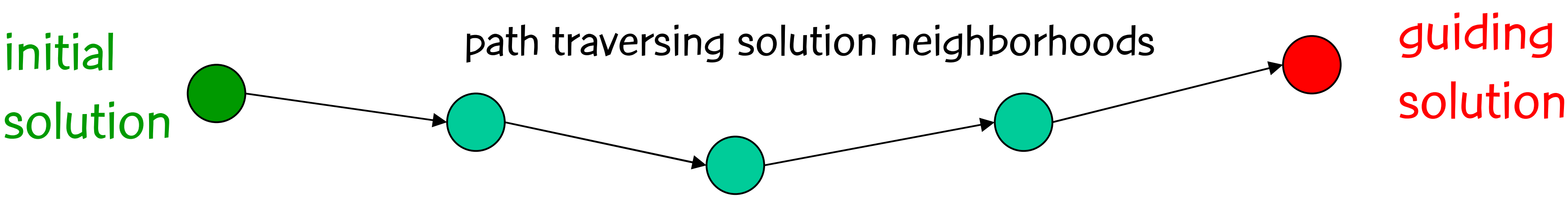
      to remove superfluous elements in the solution
    - x ← (1,1)-exchange(x)

      to replace a more expensive element in the solution by a less expensive one not in the solution

    while x is improved

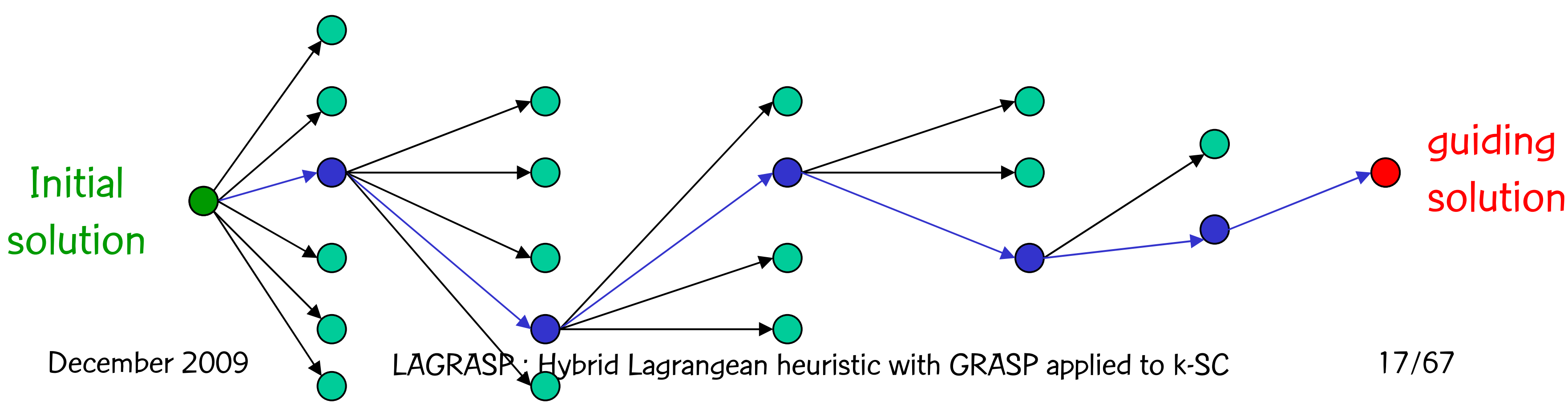
GRASP with path-relinking

- Path-relinking:
  - Introduced in the context of tabu search by Glover (1996)
    - Intensification strategy using set of elite solutions
  - Consists in exploring trajectories that connect high quality solutions.



GRASP with path-relinking

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

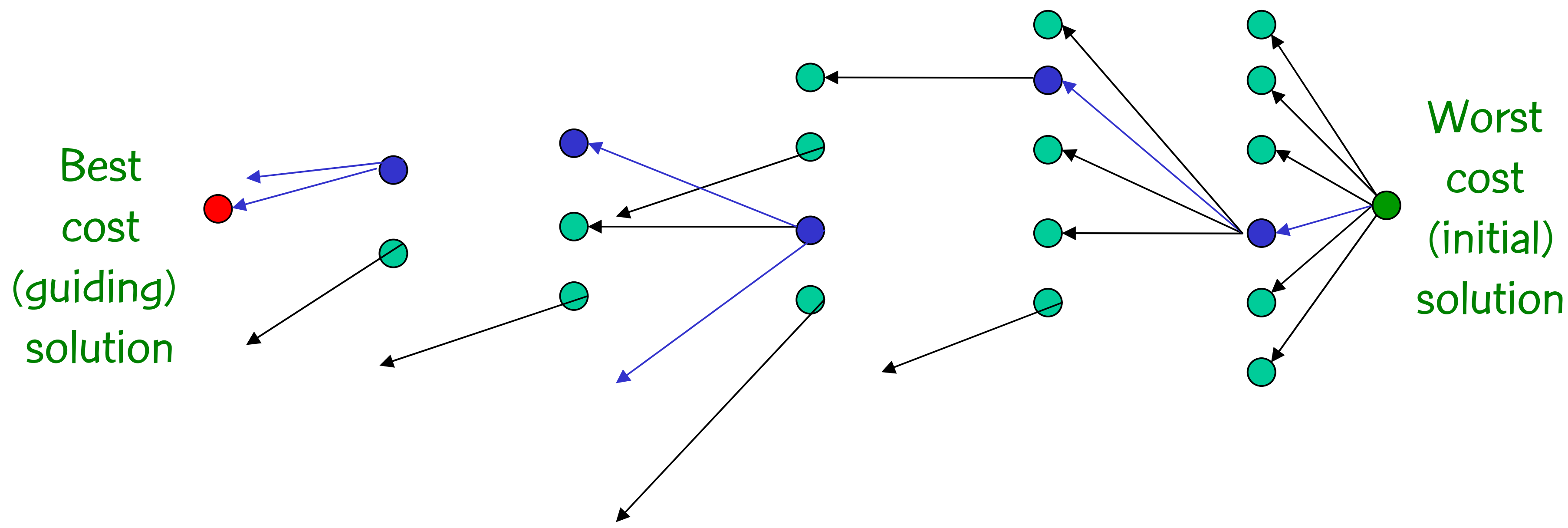


GRASP with path-relinking

- Path-relinking phase:
  - Maintain an elite set of diverse high-quality solutions found during previous GRASP iterations.
  - After each GRASP iteration (construction & local search):
    - x_g is the locally optimal GRASP solution
    - Select an elite solution, x_p, at random
    - Perform path-relinking between x_g and x_p

GRASP with path-relinking

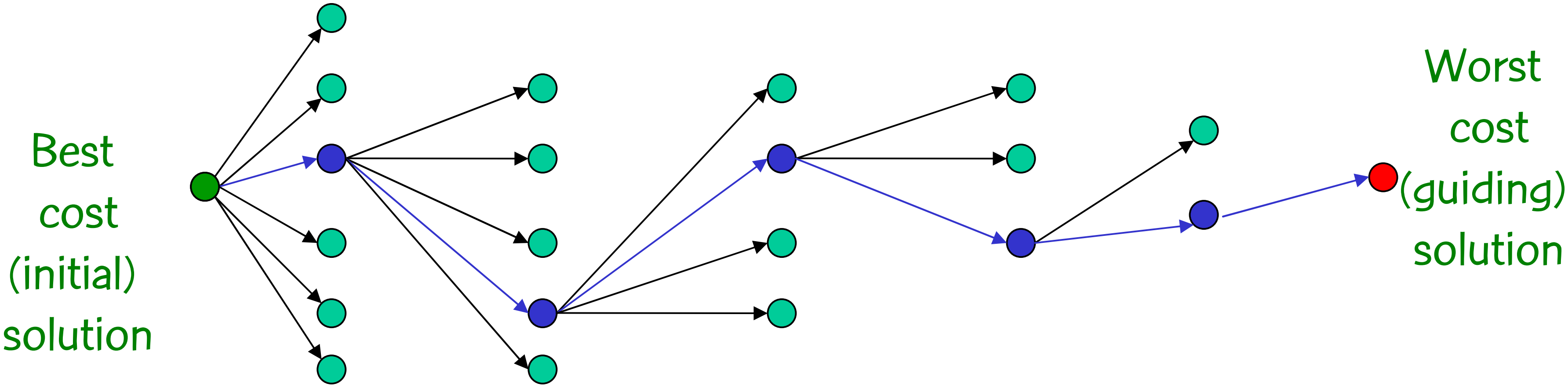
- Variants of GRASP with path-relinking
  - GRASP with forward path-relinking (GPRf):



- Starting solution is the worst between x_g and x_p.

GRASP with path-relinking

- Variants of GRASP with path-relinking
  - GRASP with backward path-relinking (GPRb):



Performs systematically better than forward PR.

- Starting solution is the best between x_g and x_p.

GRASP with path-relinking

- Variants of GRASP with path-relinking
  - GRASP with mixed path-relinking (GPRm):

- Starting and guiding solutions are interchanged at each step.

GRASP with path-relinking

- Variants of GRASP with path-relinking
  - GRASP with mixed path-relinking (GPRm):

- Starting and guiding solutions are interchanged at each step.

GRASP with path-relinking

- Variants of GRASP with path-relinking
  - GRASP with mixed path-relinking (GPRm):

- Starting and guiding solutions are interchanged at each step.

GRASP with path-relinking

- Variants of GRASP with path-relinking
  - GRASP with mixed path-relinking (GPRm):

- Starting and guiding solutions are interchanged at each step.

GRASP with path-relinking

- Variants of GRASP with path-relinking
  - GRASP with mixed path-relinking (GPRm):

- Starting and guiding solutions are interchanged at each step.

GRASP with path-relinking

- Variants of GRASP with path-relinking
  - GRASP with mixed path-relinking (GPRm):

- Starting and guiding solutions are interchanged at each step.

GRASP with path-relinking

- Variants of GRASP with path-relinking
  - GRASP with mixed path-relinking (GPRm):

- Starting and guiding solutions are interchanged at each step.

GRASP with path-relinking

- Variants of GRASP with path-relinking
  - GRASP with mixed path-relinking (GPRm):

- Starting and guiding solutions are interchanged at each step.

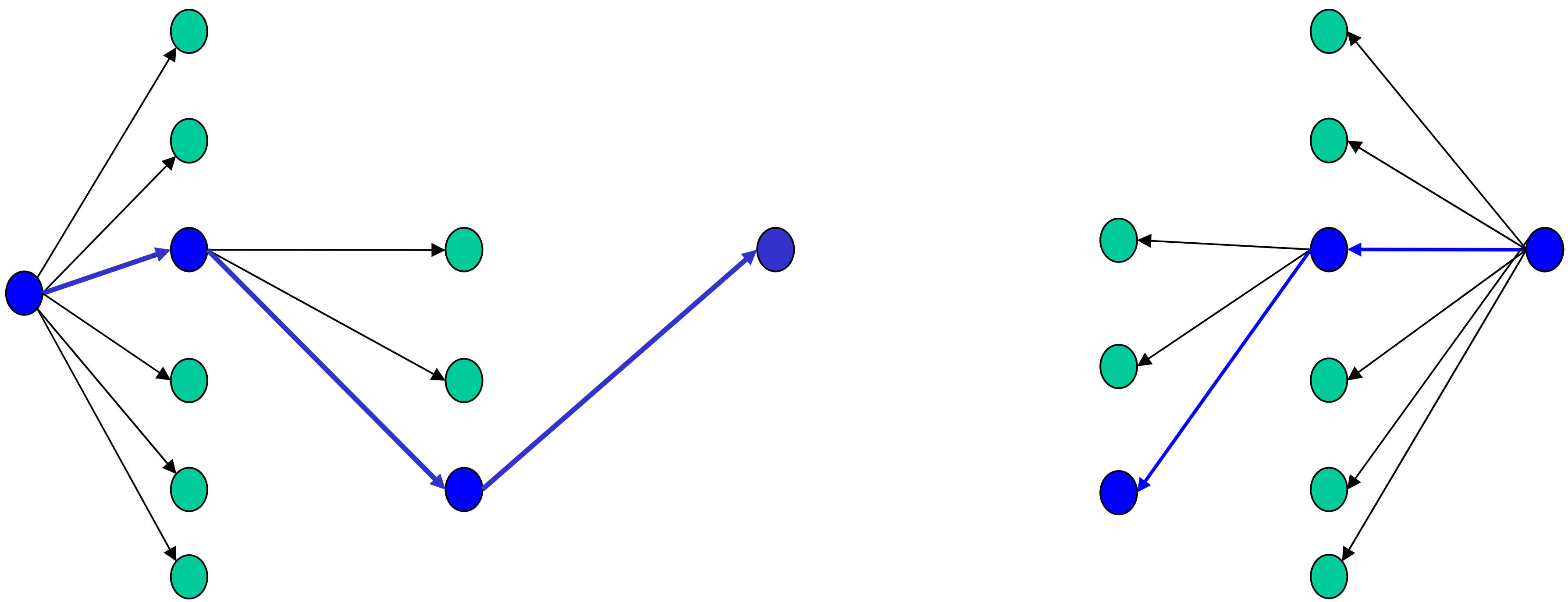
GRASP with path-relinking

- Variants of GRASP with path-relinking
  - GRASP with mixed path-relinking (GPRm):

- Starting and guiding solutions are interchanged at each step

GRASP with path-relinking

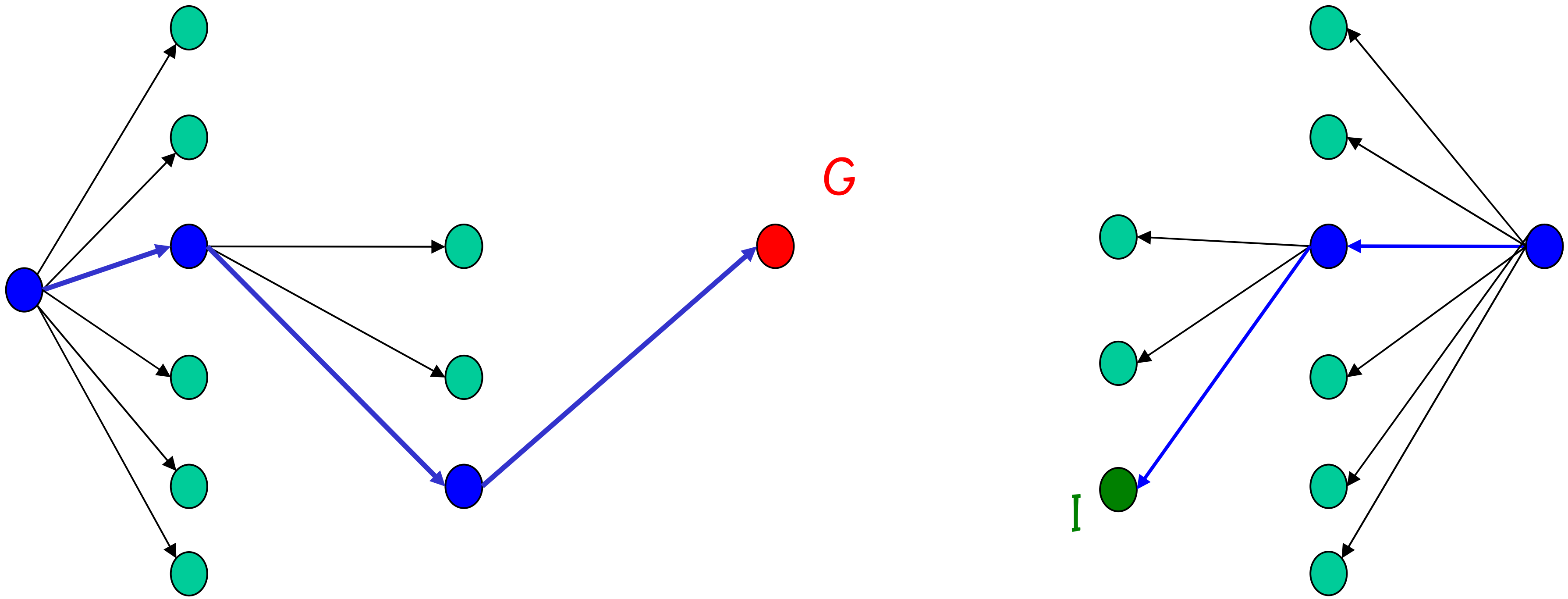
- Variants of GRASP with path-relinking
  - GRASP with mixed path-relinking (GPRm):



- Starting and guiding solutions are interchanged at each step.

GRASP with path-relinking

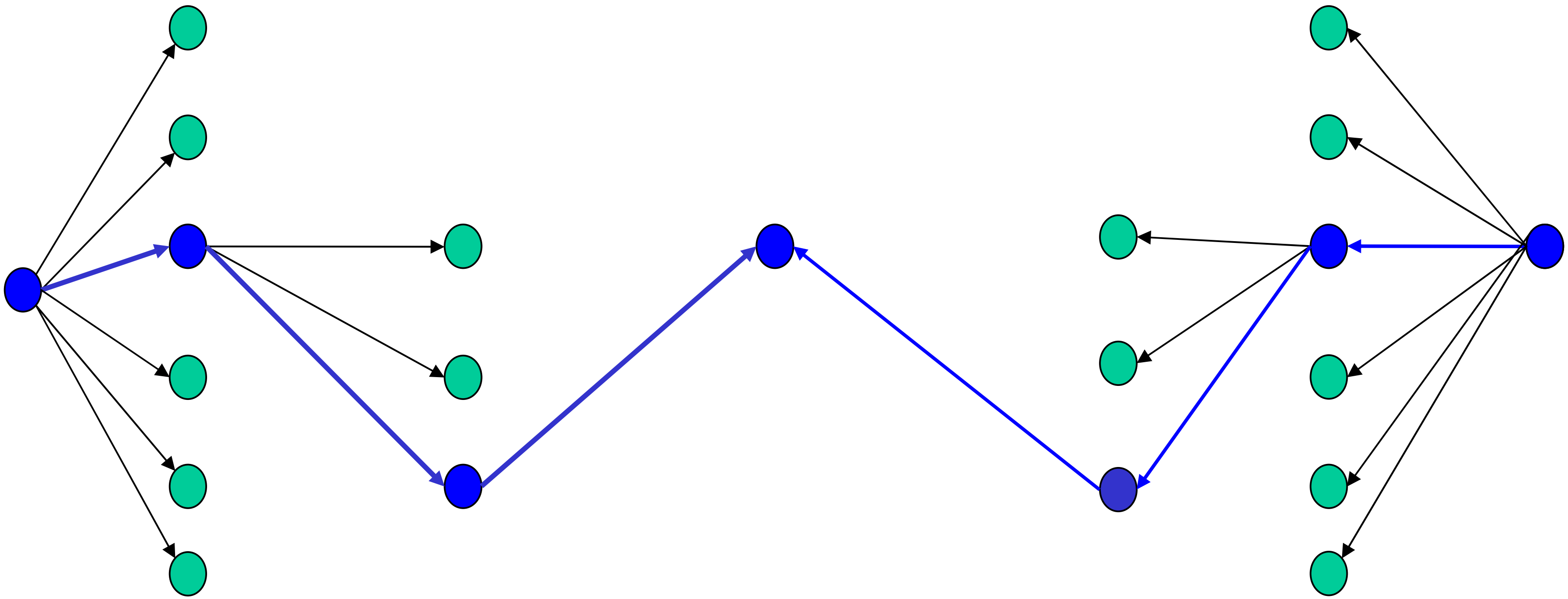
- Variants of GRASP with path-relinking
  - GRASP with mixed path-relinking (GPRm):



- Starting and guiding solutions are interchanged at each step.

GRASP with path-relinking

- Variants of GRASP with path-relinking
  - GRASP with mixed path-relinking (GPRm):



- Starting and guiding solutions are interchanged at each step.

Experiments with GRASP

- 135 test problems:
  - Derived from 45 OR-Library instances for the set covering problem.

classes	dimension	density	quantity
scp4	200×1000	2%	10
scp5	200×2000	2%	10
scp6	200×1000	5%	5
scpa	300×3000	2%	5
scpb	300×3000	5%	5
scpc	400×4000	2%	5
scpd	400×4000	5%	5

  - Three coverage factors:
    - k_{min}: $k = 2$ for all instances;
    - k_{max}: $k = \min_{i=1,\ldots,m} \sum_{j=1}^{n} a_{ij}$;
    - k_{med}: $k = \lceil (k_{min} + k_{max})/2 \rceil$

Experiments with GRASP

- Four versions: Gpure, GPRb, GPRf, and GPRm
- Parameter α self-adjusted with Reactive GRASP (Prais and Ribeiro, 2000)
- Stopping criterion: running time needed to perform 1,000 iterations of pure GRASP

classes	k_{min}	k_{med}	k_{kmax}
scp4	5	15	27
scp5	10	45	90
scp6	5	20	38
scpa	21	141	265
scpb	17	235	288
scpc	39	329	580
scpd	26	489	544

Time in seconds

- 8 runs for each instance and algorithm on Intel Xeon Quadcore 2.33GHz

Experiments with GRASP

- GRASP results compared with CPLEX solutions.
- CPLEX running times limited to 24 hours on SGI Altix 3700 Supercluster of 1.5GHz Itanium processors.
- CPLEX found optimal solutions for:
  - kmin: 41 out of 45 instances
  - kmed: 15 out of 45 instances
  - kmax: 6 out of 45 instances
- Largest integrality gap was 0.8%.

Experiments with GRASP

	CPLEX	Gpure	GPRb	GPRf	GPRm
MDif	0.00 %	4.84 %	3.45 %	3.51 %	3.51 %
#Best	135	0	0	0	0
Score	0	324	304	319	320

- MDif: average relative deviation with respect to best CPLEX solution values over all instances
- #Best: number of instances for which each method found solutions as good as best CPLEX solutions
- Score: number of times (sum over all instances) other methods found better solutions (the lower the value of Score, the better the method)

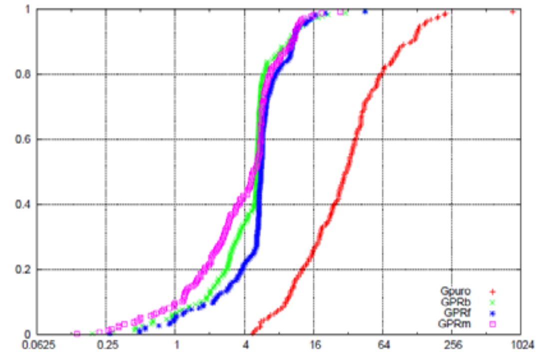
Experiments with GRASP

	CPLEX	Gpure	GPRb	GPRf	GPRm
MDif	0.00 %	4.84 %	3.45 %	3.51 %	3.51 %
#Best	135	0	0	0	0
Score	0	324	304	319	320

- GRASP was not able to find good solutions matching the best solutions obtained by CPLEX.
- GPRb found better solutions, on average, than the other versions of GRASP.

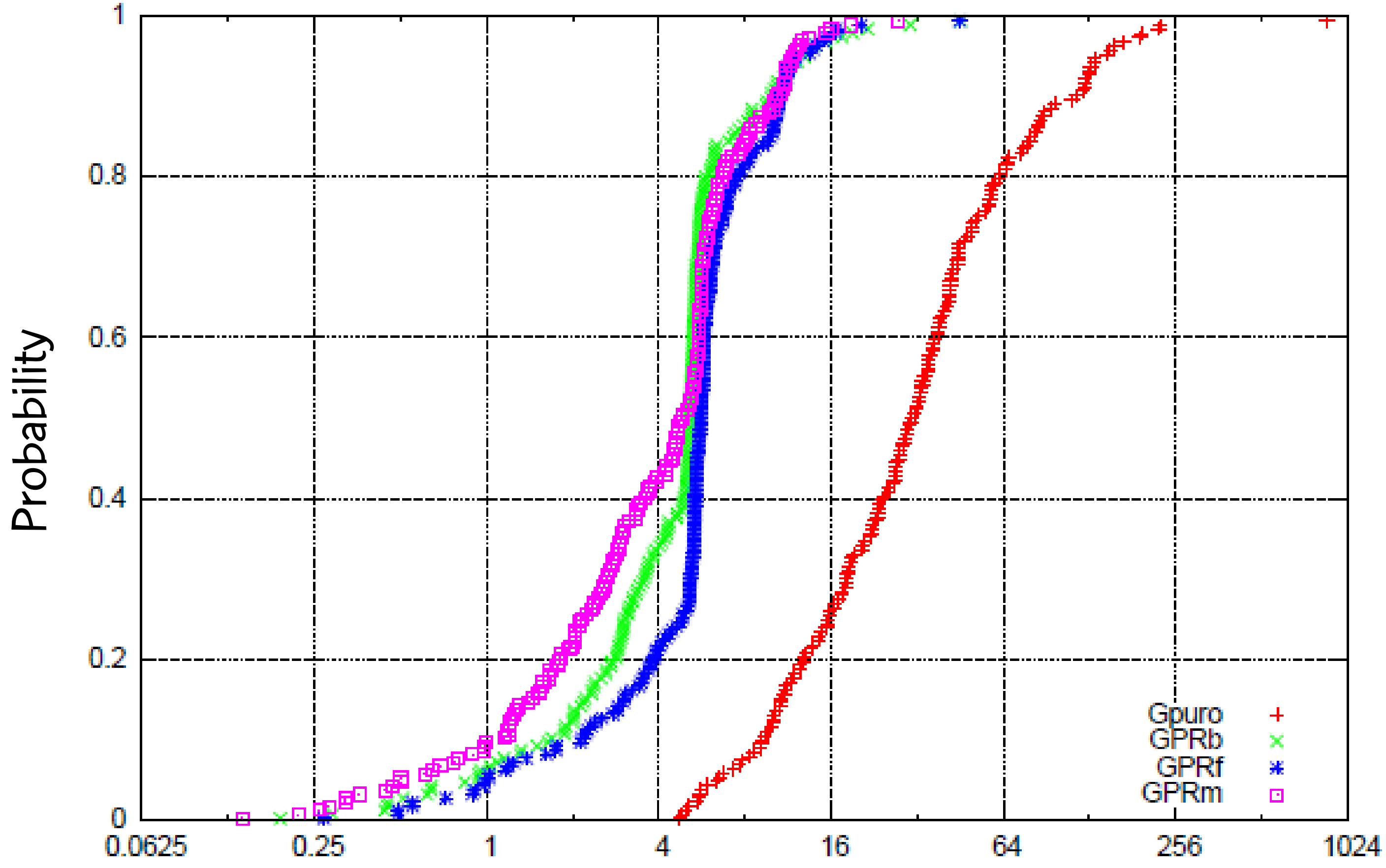
Experiments with GRASP

- Time-to-target-value plots (Aiex, Resende and Ribeiro, 2002) or run time distributions:
  - Probability of finding a solution at least as good as a target value within some running time
  - Select instance and target value.
  - For each variant of GRASP:
    - Perform 200 runs from different seeds.
    - Stop when a solution at least as good as target is found.
    - For each run, measure the time-to-target-value
    - Plot the run time distribution of finding a solution at least as good as target within some computation time.



Experiments with GRASP

- Typical time-to-target-value (ttt) plots:



Time (s)

Instance scpa2-k_{min}

Experiments with GRASP

- In conclusion:
  - Pure GRASP found solutions with cost, on average, 4.84% off of CPLEX values.
  - Path-relinking improved pure GRASP.
  - GRASP with backward path-relinking obtained, on average and over all test instances, the best results:
    - *Average cost of solutions found by GPRb is 3,45% off of the cost of CPLEX solutions.*

Lagrangean heuristic

- Constraint ... $\geq$ k is dualized with multipliers λ.
- Dual problem solved by subgradient optimization:
  - Multipliers adjustment following Held, Wolfe and Crowder, 1974 (see also Beasley, 1993)
  - At every subgradient optimization iteration:
    - Let $x(\lambda)$ be the optimal solution to Lagrangean problem.
    - Make use of a basic heuristic to produce a primal solution.
    - Upper bound given by the primal solution is used to update the step-size of the process that adjusts the multipliers.
  - Similar to Caprara, Fischetti and Toth, 1999

Lagrangean heuristic

- Basic heuristic builds primal solution x from initial solution x^0 using modified costs γ
  - Initial solution x^0:
    - $X^0 = x(\lambda)$
    - $x_j^0 = 0$, for $j = 1, \ldots, n$
  - Modified costs γ:
    - Lagrangean costs c'

$$c_j' = c_j - \sum_{i=1}^{m} \lambda_i . a_{ij}$$

    - Complementary costs $\bar{c}$

$$\bar{c}_j = (1 - x_j(\lambda)).c_j$$

Lagrangean heuristic

- Greedy basic heuristic:
  - Greedy construction
    - Starting from x^0, iteratively build a solution x by setting to 1 the variable x_j with the smallest ratio between its modified cost γ_j and the number of still uncovered rows that it covers.
  - Local search
    - Same local search used by GRASP
    - Apply (1,0)-exchange and (1,1)-exchange to the greedy solution, using the original costs.

Hybrid Lagrangean heuristic with GRASP

- GRASP basic heuristic:
  - Slightly modified version of GRASP procedure
  - Repeat for max number of iterations:
    - Greedy randomized construction:
      - Make use of modified costs γ instead of original costs.
      - Build a feasible solution x from x^0 (not necessarilly from scratch).
    - Apply local search.
    - Apply path-relinking.
- Hybrid Lagrangean heuristic with GRASP: LAGRASP

Hybrid Lagrangean heuristic with GRASP

- Greedy Lagrangean heuristic:

At each iteration of the subgradient method:

- Perform greedy basic heuristic.

- Hybrid Lagrangean with GRASP heuristic (LAGRASP):

After every H iterations of the subgradient method:

- Perform GRASP basic heuristic with probability ß or greedy basic heuristic with probability (1-ß).

Experiments with Lagrangean heuristics

- Lagrangean heuristics
  - Stopping criterion:
    - Step-size parameter $\eta \leq 10^{-4}$ (initially set at 2 and halved after every 50 consecutive iterations without improvement in the lower bound)
    - Lower bound matches upper bound, i.e. UB – LB < 1
  - Each version: 8 runs
  - Experiments on Intel Xeon Quadcore 2.33GHz
  - Best results compared to CPLEX solutions.

Experiments with Lagrangean heuristics

- Greedy Lagrangean heuristic
  - Four versions according to modified cost scheme and initial solution used by basic heuristic:
    - GLH1-LL: Lagrangean modified costs to build a feasible solution from the Lagrangean problem solution
    - GLH2-CL: Complementary modified costs to build a feasible solution from Lagrangean problem solution
    - GLH3-LS: Lagrangean modified costs to build a feasible solution from scratch
    - GLH4-CS: Complementary modified costs to build a feasible solution from scratch

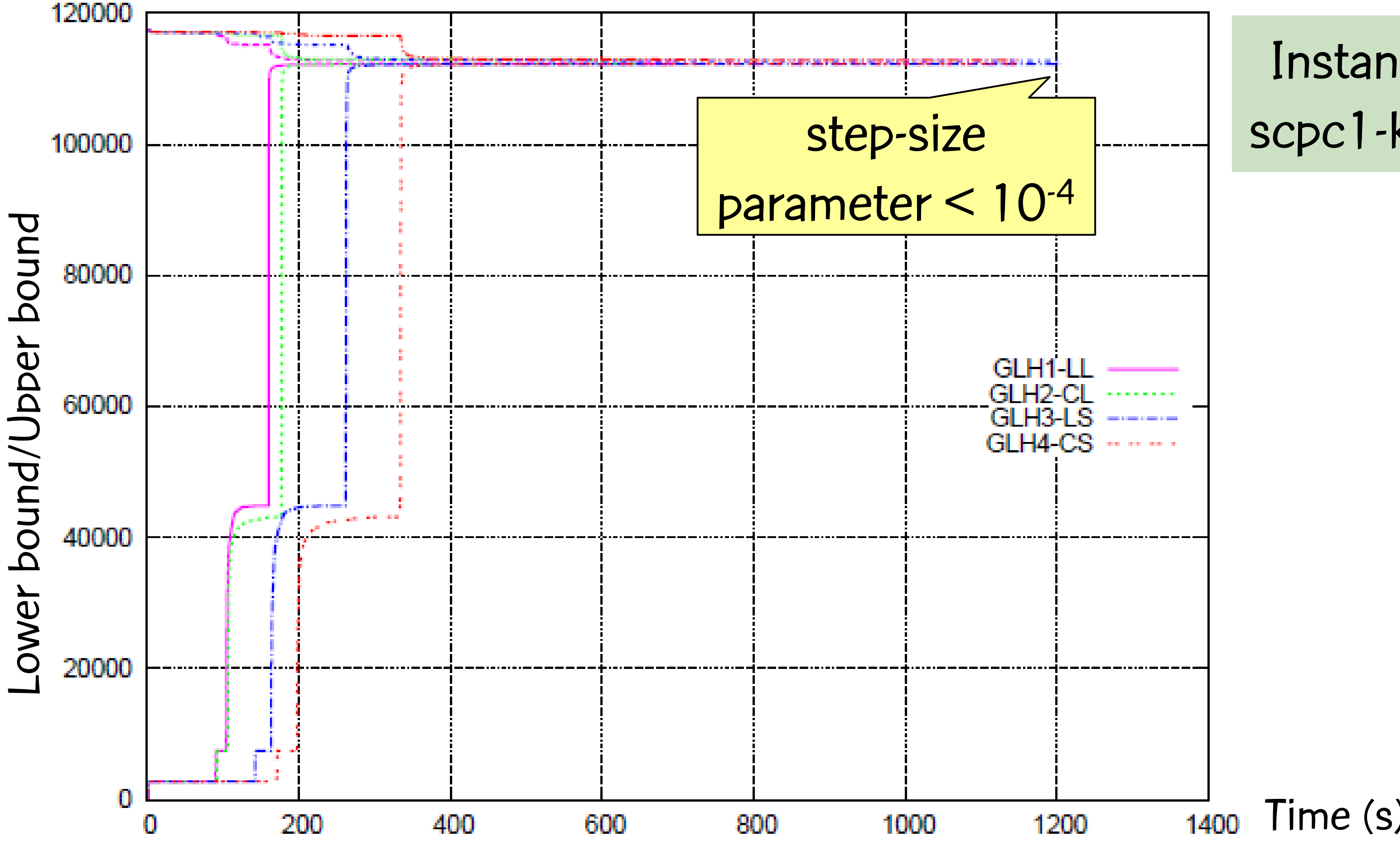
Experiments with Lagrangean heuristics

	CPLEX	GLH1-LL	GLH2-CL	GLH3-LS	GLH4-CS
MDif	0.00 %	0.30 %	0.32 %	0.30 %	0.30 %
#Best	135	24	21	24	24
Score	0	194	330	209	264
Time (s)	–	24274.71	22677.02	37547.50	41804.25

- Building primal feasible solutions from Lagrangean problem solutions appears to be faster: similar times for GLH1-LL and GLH2-CL.
- All versions found, at least, 21 solutions as good as CPLEX over the 135 problem instances.
- Best overall results obtained, on average, by GLH1-LL.

Experiments with Lagrangean heuristics

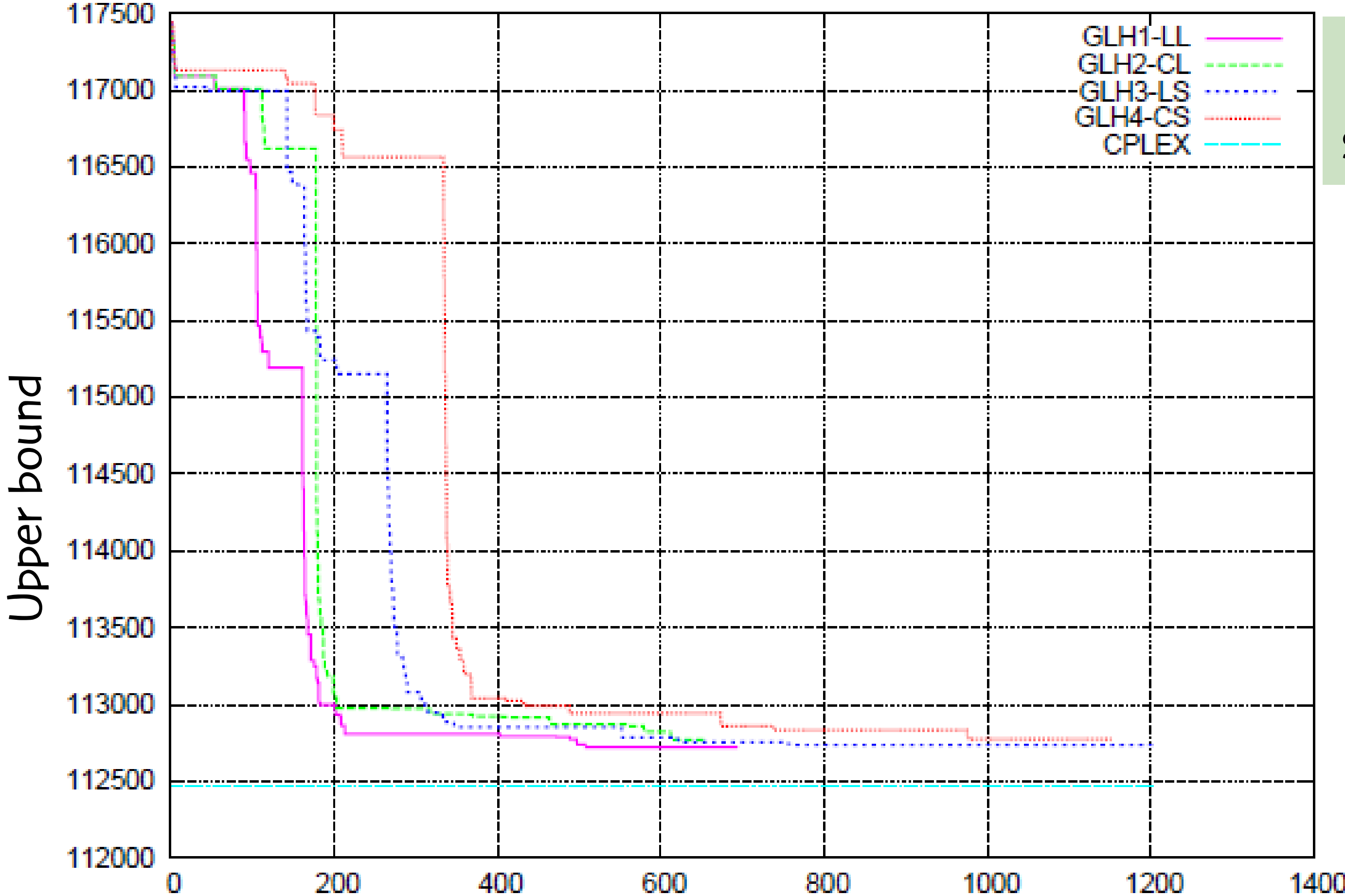
- Lower and upper bounds with running time:



Instance scpc1-k_{max}

Experiments with Lagrangean heuristics

- Upper bound with running time (same instance):



Instance scpc1-k_{max}

Experiments with Lagrangean heuristics

- Hybrid Lagrangean with GRASP heuristic:
  - Combines best GRASP with path-relinking strategy with greedy Lagrangean heuristic
    - Basic (greedy and GRASP) heuristics
      - Make use of Lagrangean modified costs to build feasible primal solution from Lagrangean problem solution
    - GRASP basic heuristic
      - Backward path-relinking
      - Elite set with, at most, 100 solutions

Experiments with Lagrangean heuristics

- Hybrid Lagrangean with GRASP heuristic
  - Parameter settings:
    - 21 instances (first of each class, each k-value)
    - LAGRASP(ß, H, max number of GRASP iterations)

{1, 5, 10, 50}

{1, 5, 10, 50}

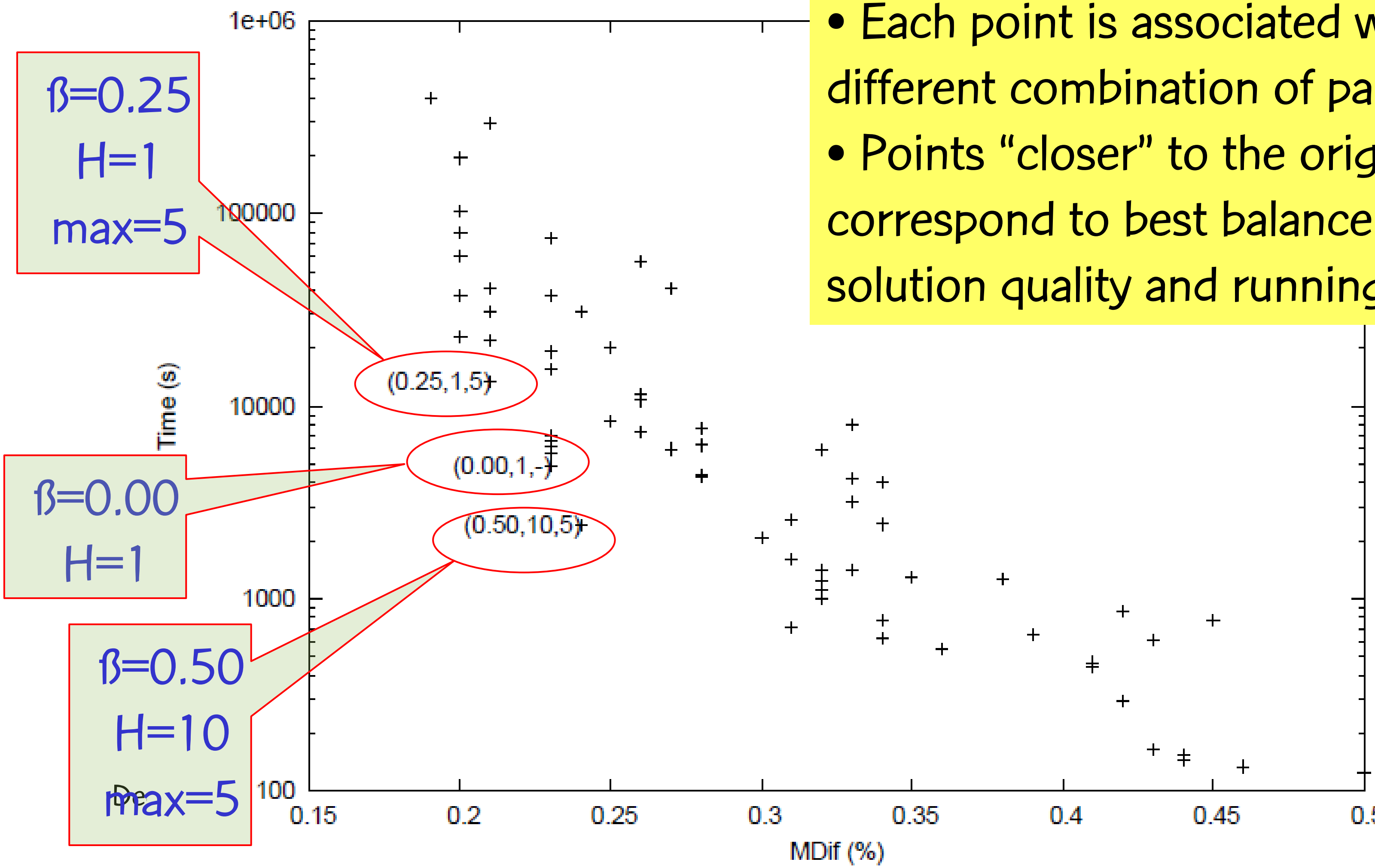
Frequency to call basic heuristic

{0, 0.25, 0.50, 0.75, 1}

Probability of calling GRASP basic heuristic

Experiments with Lagrangean heuristics

- Hybrid Lagrangean with GRASP heuristic



- Each point is associated with a different combination of parameters.
- Points "closer" to the origin correspond to best balance between solution quality and running time.

Experiments with Lagrangean heuristics

- Parameter setting
  - Three versions (i.e., parameter settings) of LAGRASP selected for the next experiment:
    - LAGRASP(0,1,-): makes use exclusively of the greedy basic heuristic.
    - LAGRASP(0.25,1,5): better average solution values than LAGRASP(0,1,-), at the cost of an increase in running time.
    - LAGRASP(0.50,10,5): smaller running times than LAGRASP(0,1,-), at the cost of finding worse solutions.

Experiments with Lagrangean heuristics

- Computational results over all 135 test instances

	CPLEX	LAGRASP $(0, 1, -)$	LAGRASP $(0.25, 1, 5)$	LAGRASP $(0.50, 10, 5)$
MDif	0.00 %	0.30 %	0.27 %	0.33 %
#Best	135	24	27	23
Score	0	191	133	272
Time (s)	–	24274.71	63603.06	11401.26

  – *All LAGRASP* versions found optimal or near-optimal solutions for all 135 instances (total time over all instances smaller than 18 hours)

Experiments with Lagrangean heuristics

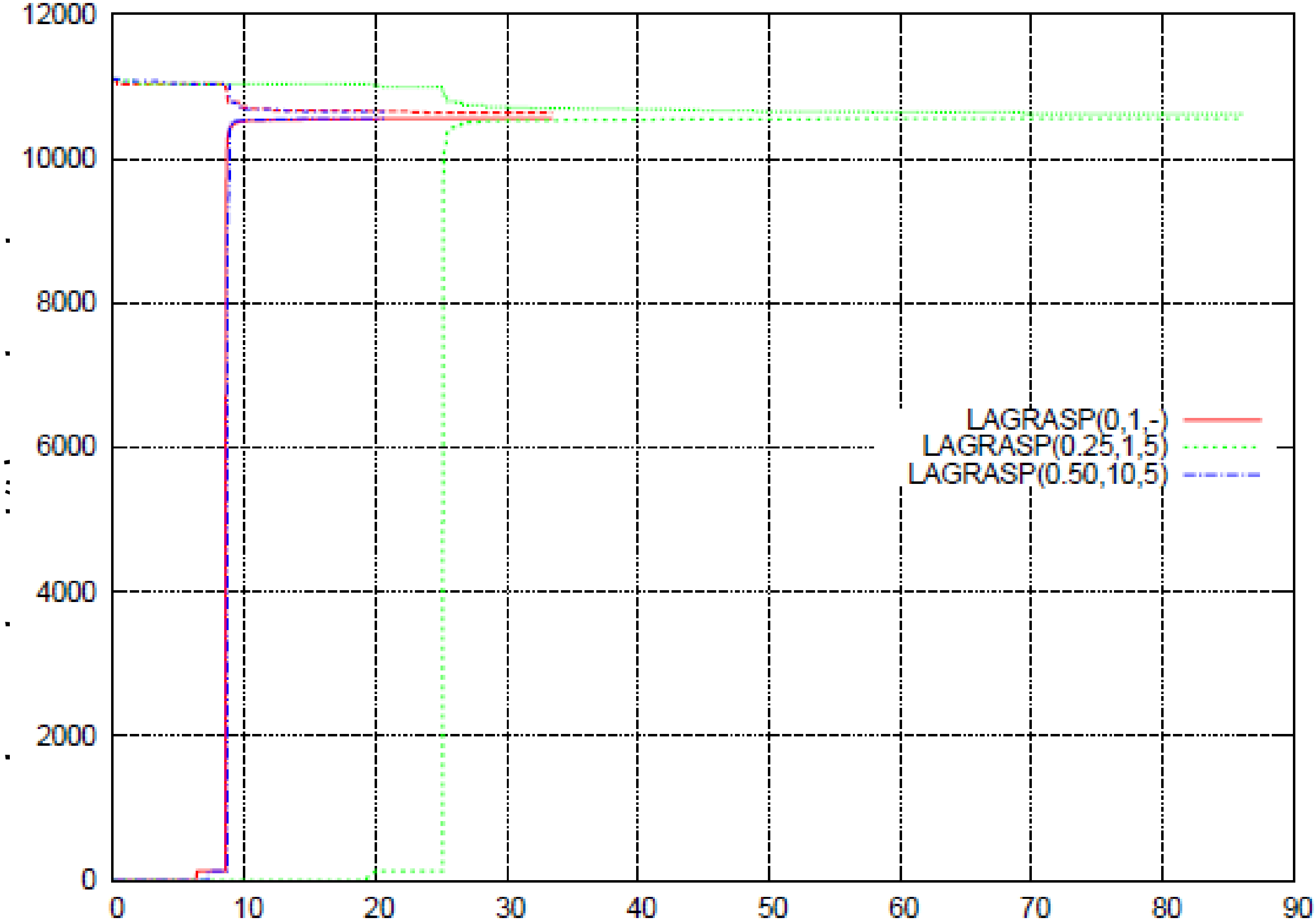
- Computational results over all 135 test instances

	CPLEX	LAGRASP (0, 1, –)	LAGRASP (0.25, 1, 5)	LAGRASP (0.50, 10, 5)
MDif	0.00 %	0.30 %	0.27 %	0.33 %
#Best	135	24	27	23
Score	0	191	133	272
Time (s)	–	24274.71	63603.06	11401.26

  - LAGRASP(0.25,1,5) reached the best results in quality metrics (MDif, #Best and Score) using the same time magnitude than the other versions of LAGRASP.

Experiments with Lagrangean heuristics

- Lower and upper bounds with running time:

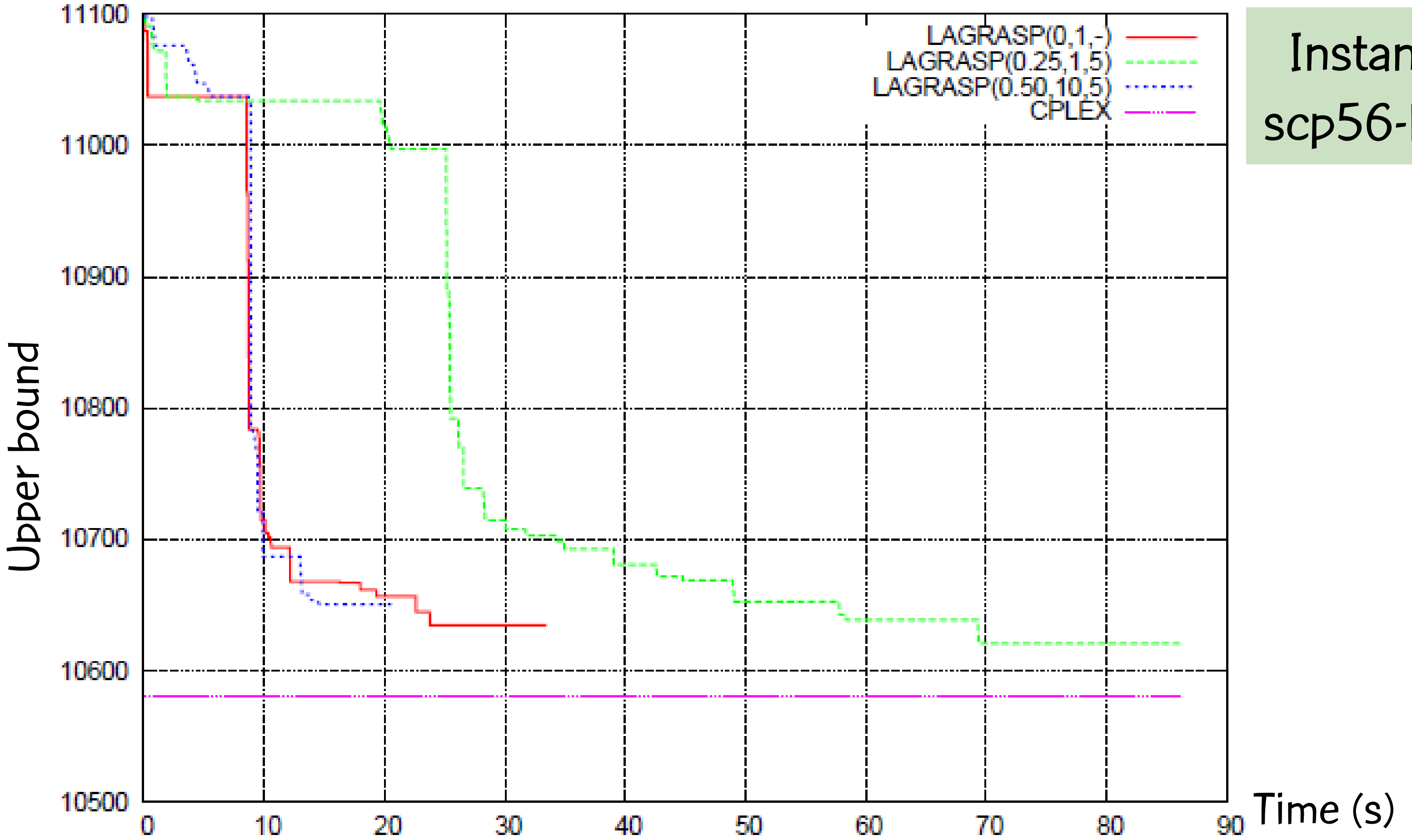


Instance scp56-k_{med}

Time (s)

Experiments with Lagrangean heuristics

- Upper bound with running time (same instance):



Instance scp56-k_{med}

Experiments with Lagrangean heuristics

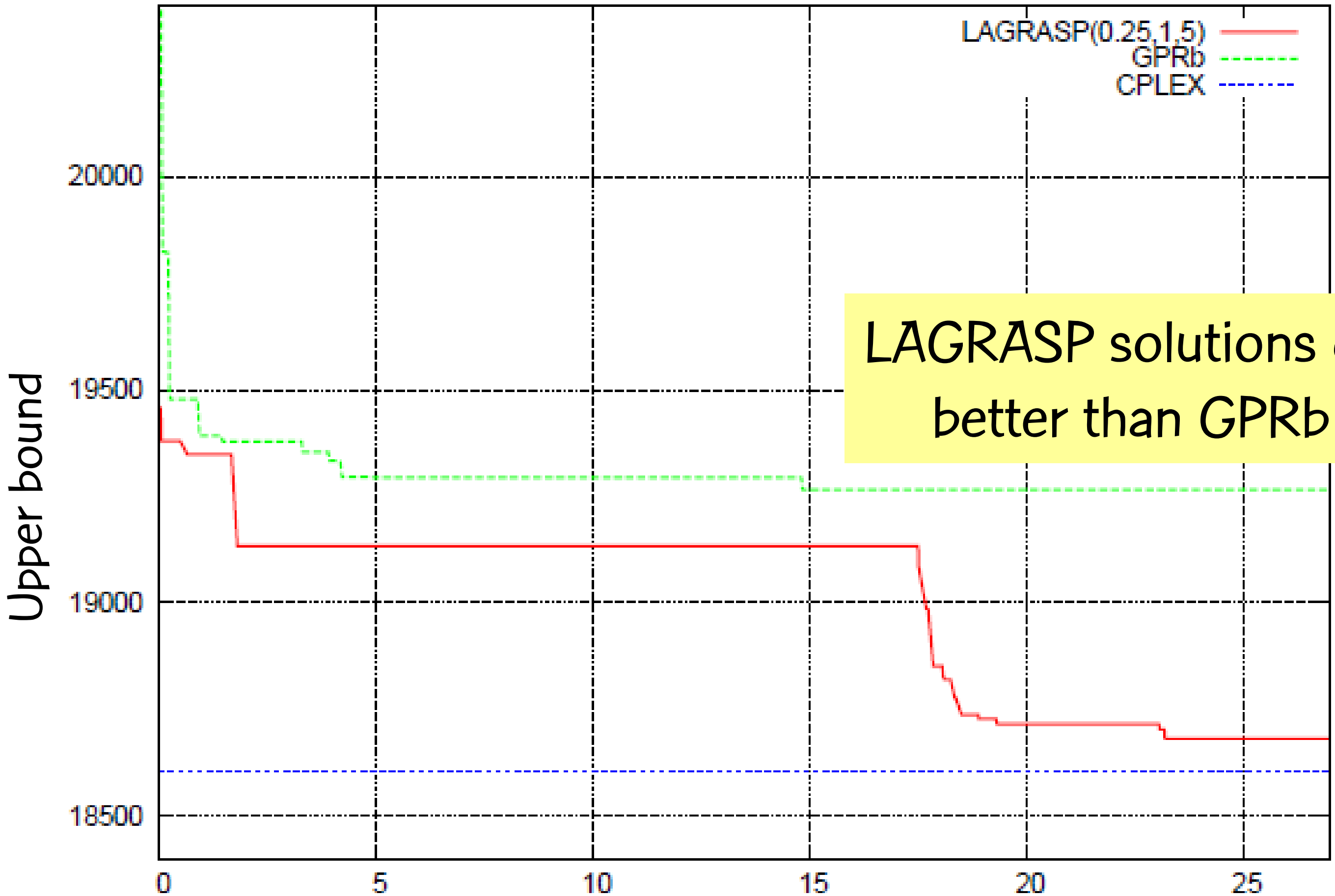
- Comparative results of LAGRASP and GPRb
  - Both heuristics used the same time limit as stopping criterion (same time limits used in the GRASP experiment):

	CPLEX	LAGRASP(0.25,1,5)	GPRb
MDif	0.00 %	0.43 %	3.46 %
#Best	135	22	0
Score	0	113	270

  - LAGRASP (0.25, 1, 5) outperformed GPRb for all metrics: smaller average deviation and solutions as good as CPLEX solutions for 22 out of 135 instances.

Experiments with Lagrangean heuristics

- Upper bound with running time:



Instance scp48-k_{max}

LAGRASP solutions consistently better than GPRb solutions

Concluding remarks (1/3)

- Redundant PoP placement problem formulated as a set k-covering problem in communications network design.
- AT&T real life instances of redundant PoP placement for dial-up internet service and fixed wireless broadband may have up to 65,000 possible locations.
- Patent titled "Designing networks with redundant points of presence using approximation methods and systems" with the US Patent Office filed in April 2009.

Concluding remarks (2/3)

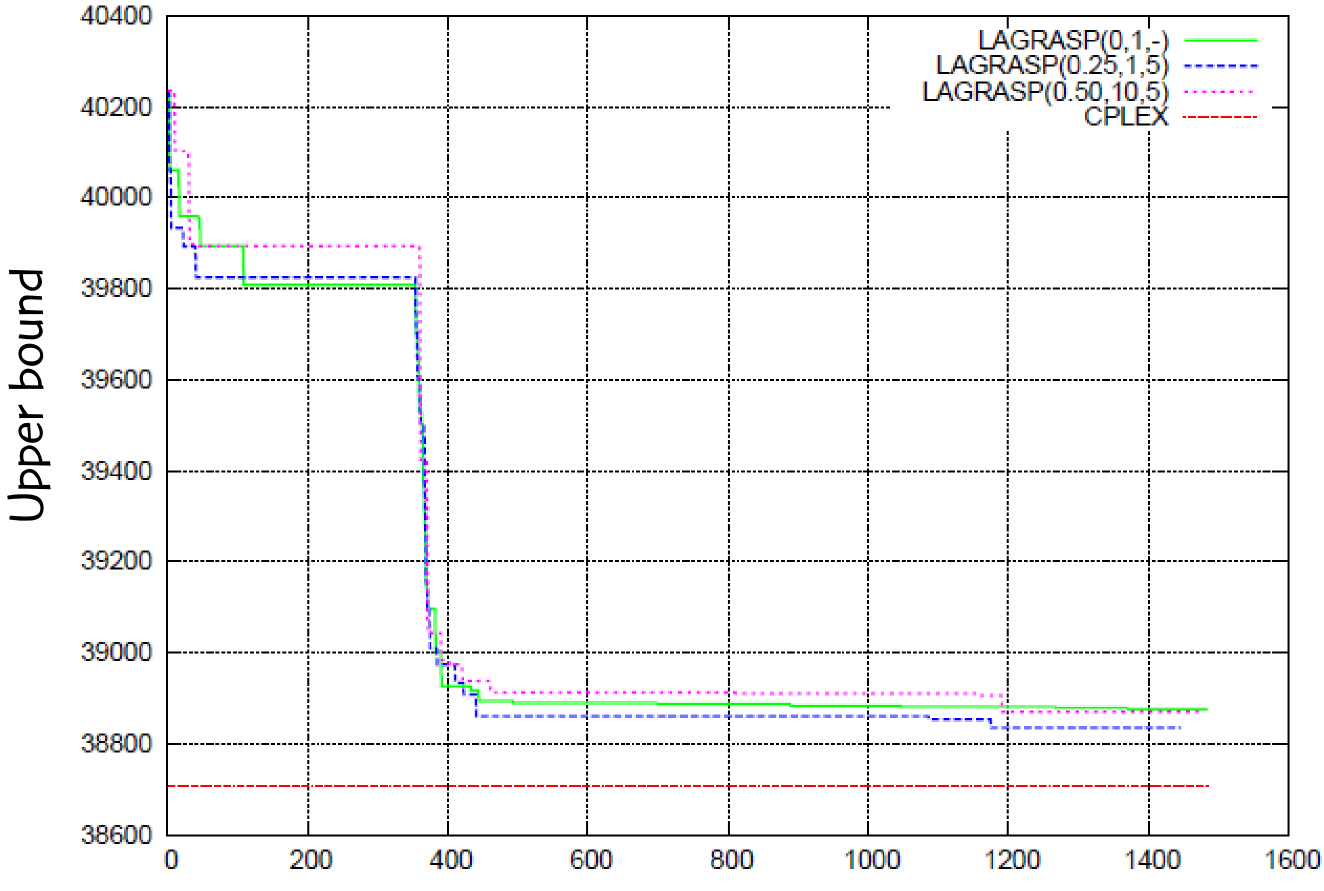
- A variety of challenging problems arising in computational biology when formulated as partition-distinguishing optimization problems can be cast into a common general framework:
  - Minimal Informative Subset probem: given a set of objects, find a minimal set of "attributes" of the objects that are "informative" with respect to the optimally distinguished partitions (Istrail, 2003).
    - Formulation as a set-covering based feature-selection.
    - Minimum Robust Tag SNPs problem: coverage factor k ensures comparison of haplotypes when some SNPs are missing.

Concluding remarks (3/3)

- Set of 135 new set k-covering test instances.
- Hybridization of GRASP with a Lagrangean heuristic improves the quality of solutions found when only a greedy basic heuristic is applied.
- LAGRASP framework being applied and tested to other problems.
- Typically, hybrid Lagrangean with GRASP heuristic is able to improve primal solutions even when dual information and greedy Lagrangean heuristic have stabilized.

Experiments with Lagrangean heuristics

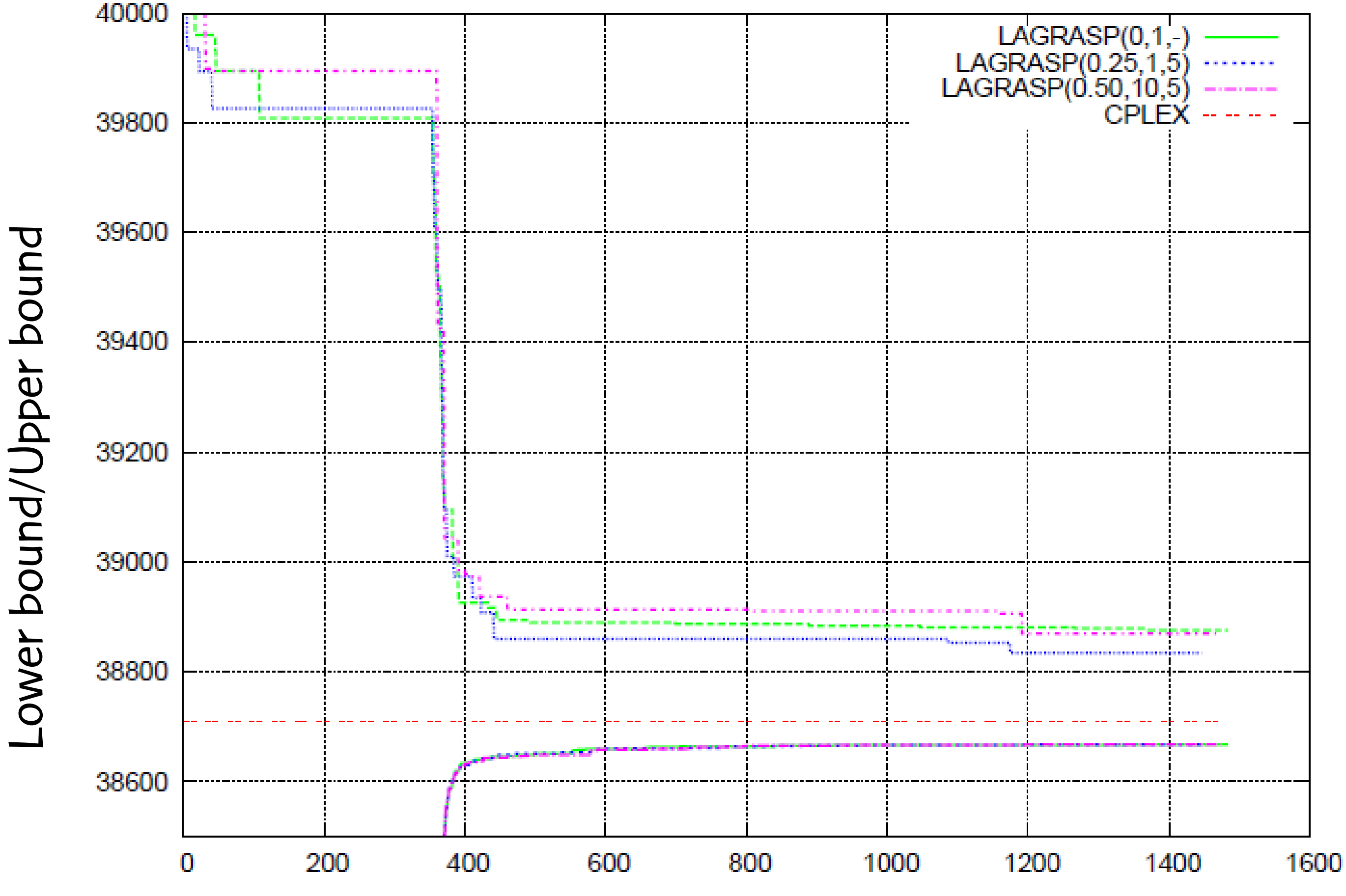
- Upper bounds with iterations:



Instance scp510-k_{max}

Experiments with Lagrangean heuristics

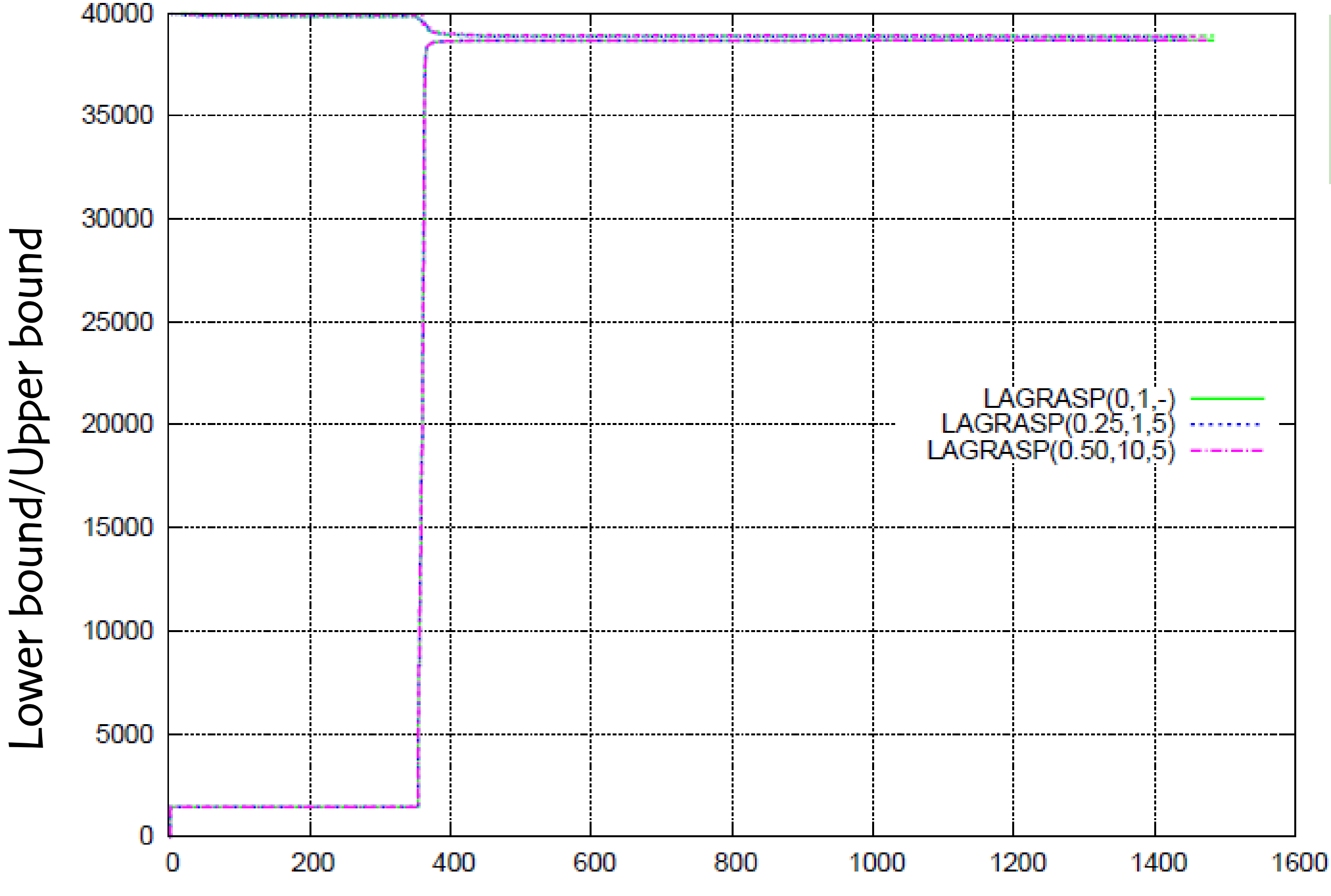
- Lower and upper bounds with iterations (zoom):



Instance scp510-k_{max}

Experiments with Lagrangean heuristics

- Lower and upper bounds with iterations:



Instance scp510-k_{max}