Restart strategies for GRASP with path-relinking heuristics

Talk given at the 10th International Symposium on Experimental Algorithms (SEA 2011)
Chania, Crete, Greece ✤ May 5, 2011

Mauricio G. C. Resende
AT&T Labs Research
Florham Park, New Jersey
mgcr@research.att.com

Joint work with Celso C. Ribeiro

Combinatorial optimization

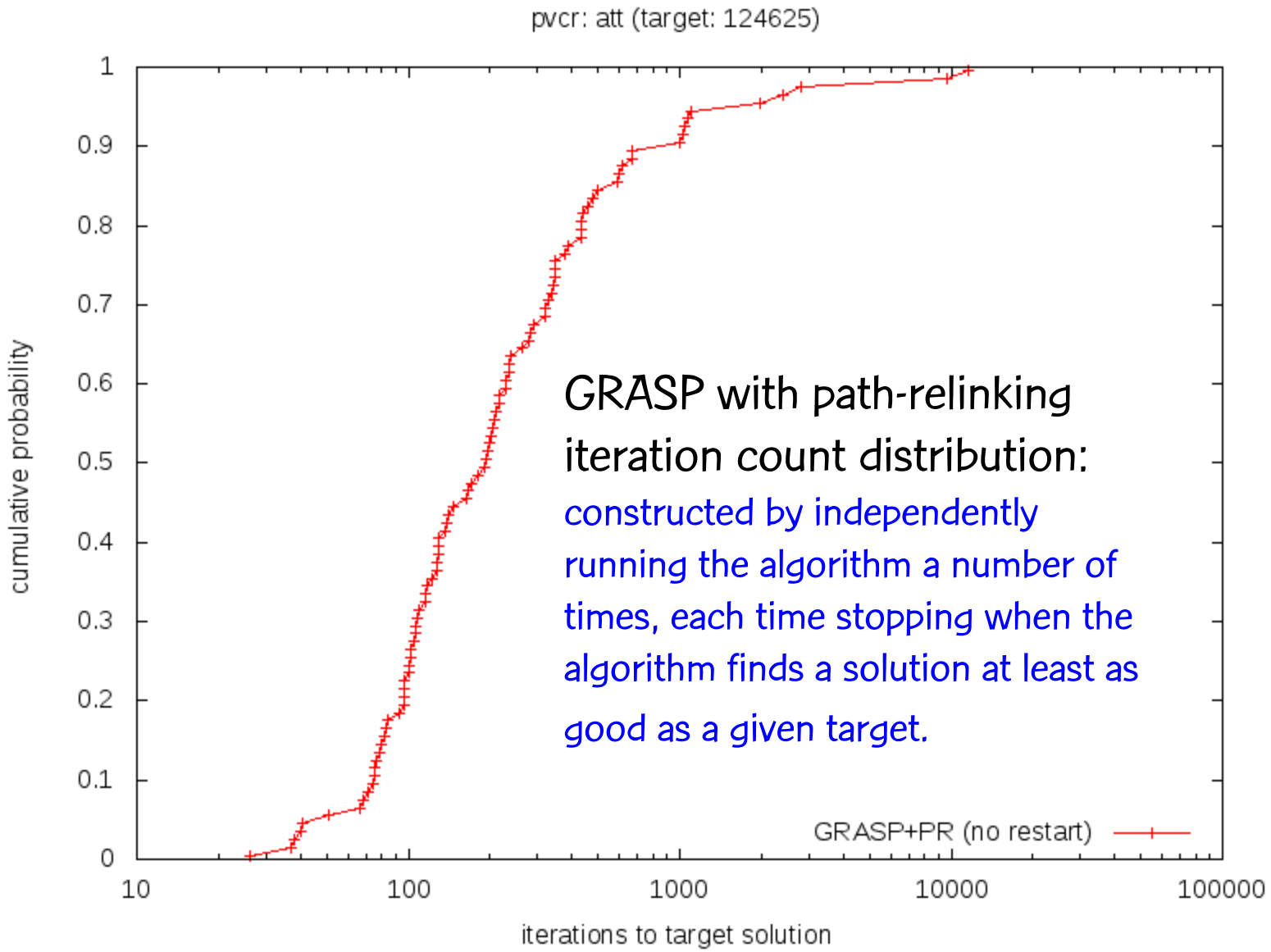
- Is defined by
  - Finite ground set $E = \{1, 2, \ldots, n\}$
  - Set of feasible solutions $F \subseteq 2^E$
  - Objective function $f: 2^E \rightarrow R$
- In its minimization version, we seek a global optimum
  - $x^* \in F$ such that
  - $f(x^*) \leqslant f(x)$, for all $x \in F$

Combinatorial optimization

- Metaheuristics are high level procedures for combinatorial optimization that coordinate simple heuristics, such as local search, to find solutions that are of better quality than those found by the simple heuristics alone.
- Many metaheuristics have been introduced in the last thirty years.
- Among these, we find genetic algorithms, tabu search, variable neighborhood search, scatter search, iterated local search, path-relinking, and GRASP.

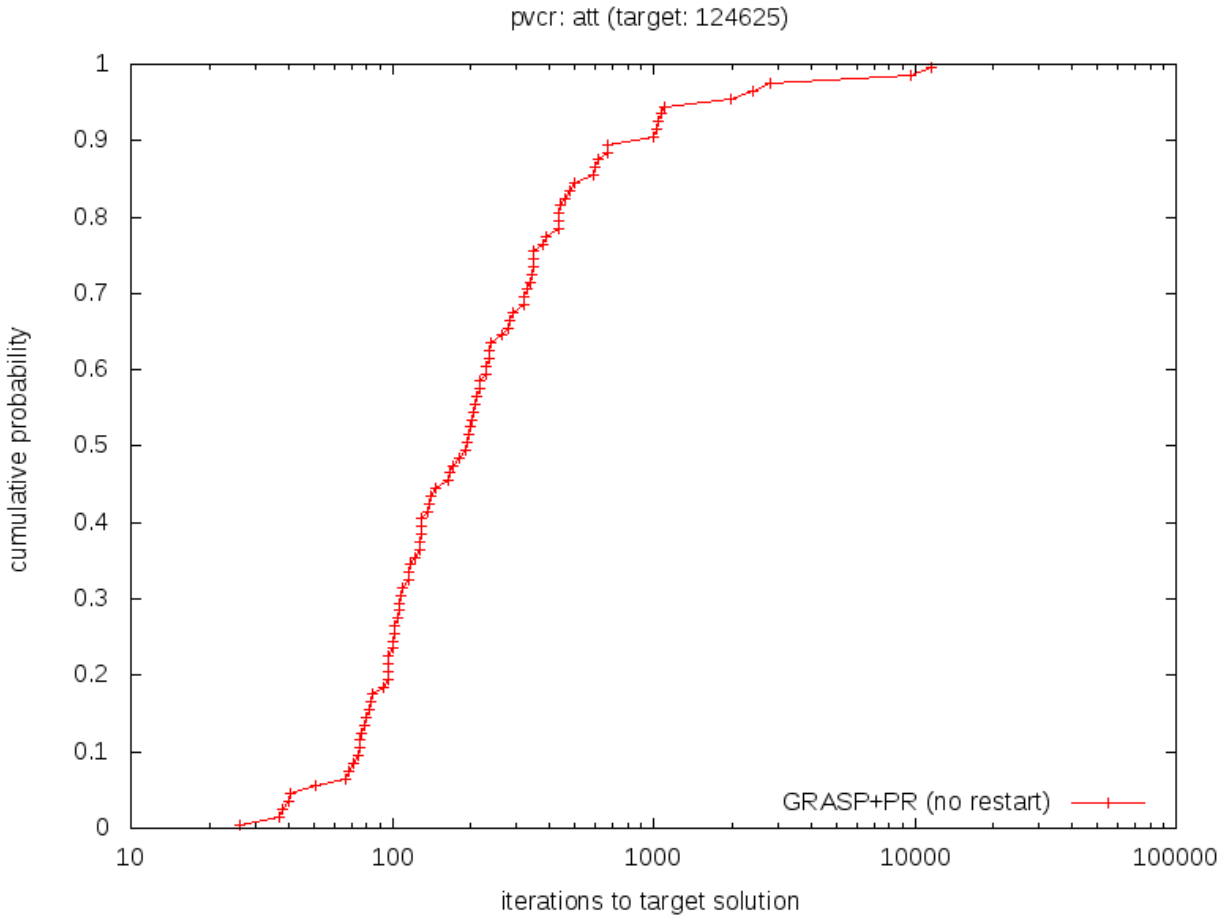
GRASP with path-relinking

- GRASP is a multi-start stochastic search metaheuristic where, in each iteration, local search is performed starting from a greedy randomized solution. A best local minimum found is returned as the solution (Feo & R., 1989, 1995)
- Path-relinking explores paths in the solution space connecting a pair of good-quality solutions, often finding even better solutions in the path (Glover, 1996).
- The hybridization of GRASP with path-relinking adds memory to GRASP and has become the standard way to implement GRASP heuristics (Laguna & Martí, 1999)

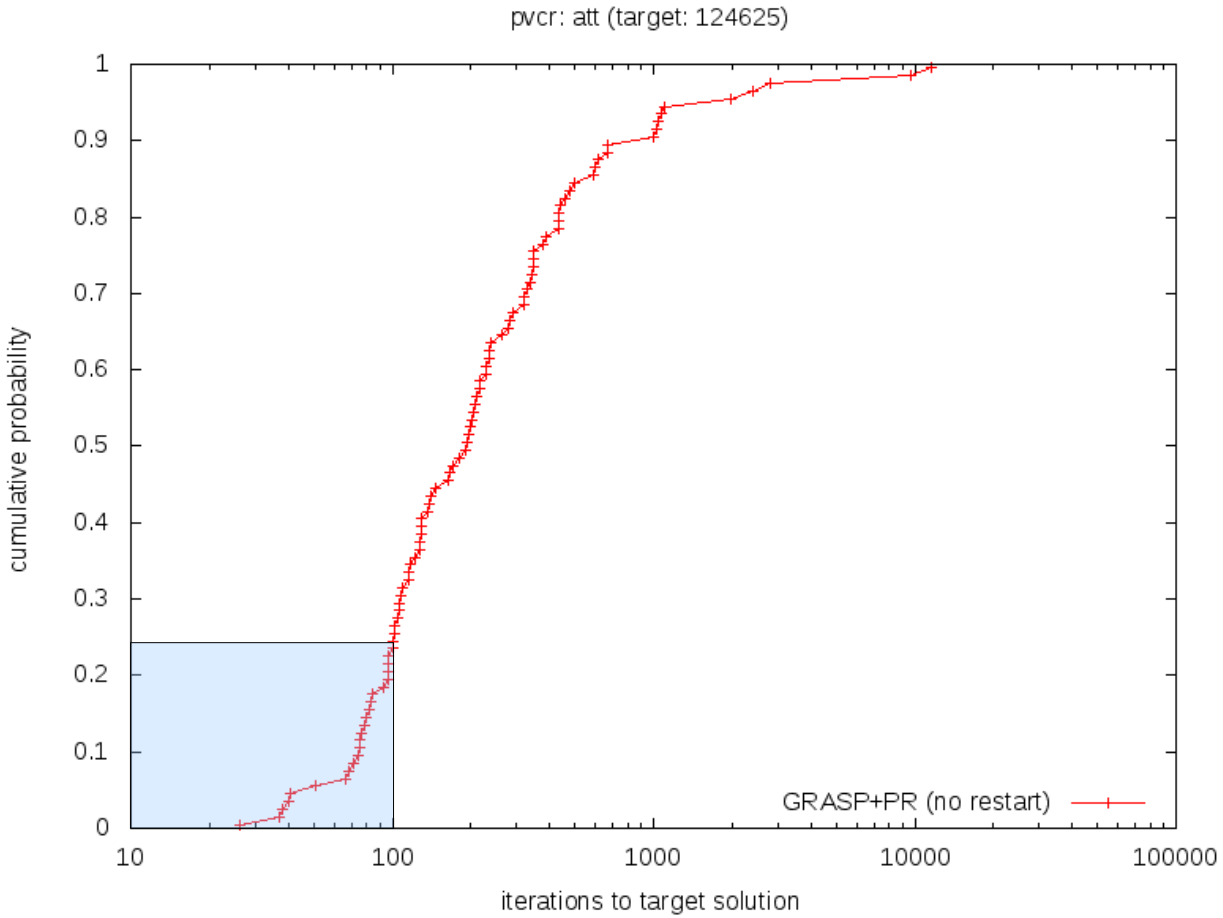


GRASP with path-relinking iteration count distribution:

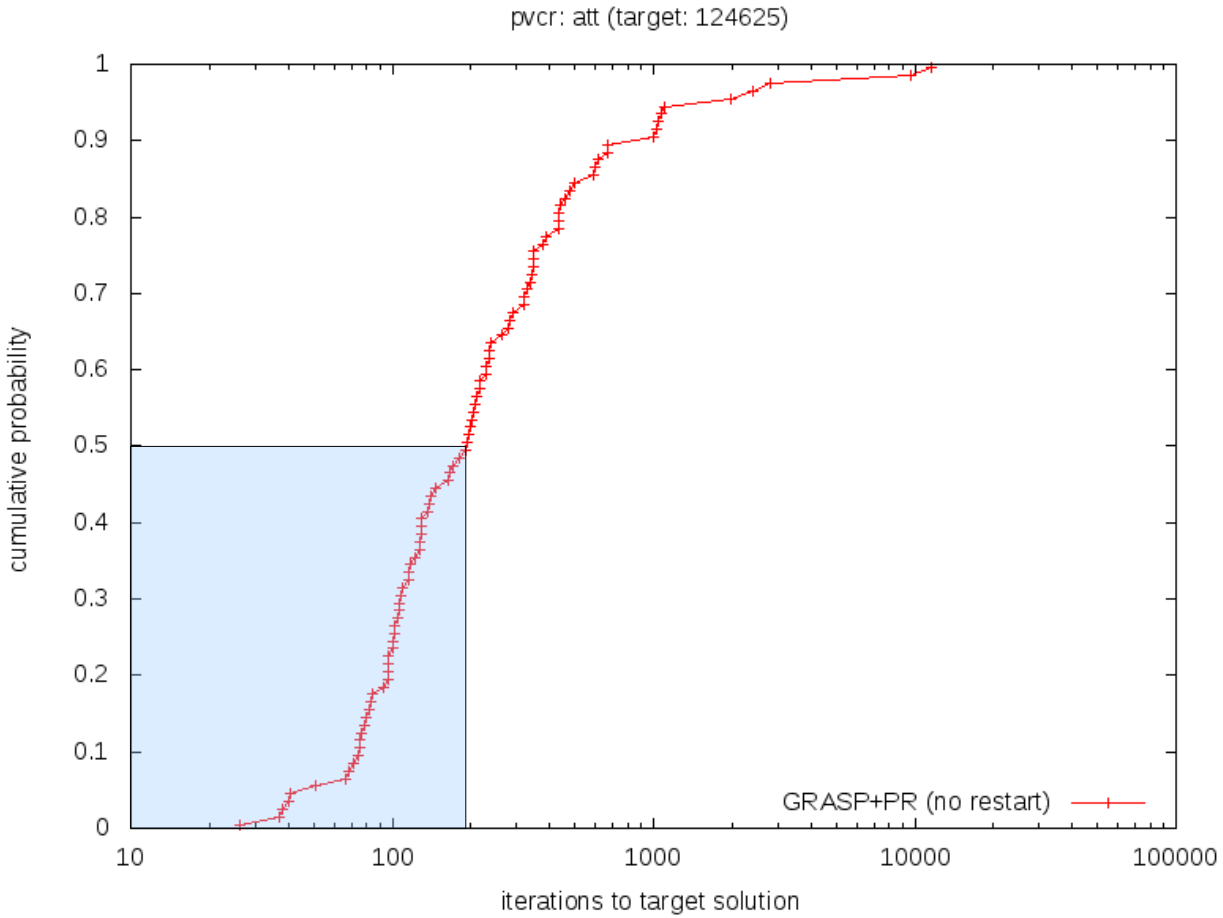
constructed by independently running the algorithm a number of times, each time stopping when the algorithm finds a solution at least as good as a given target.



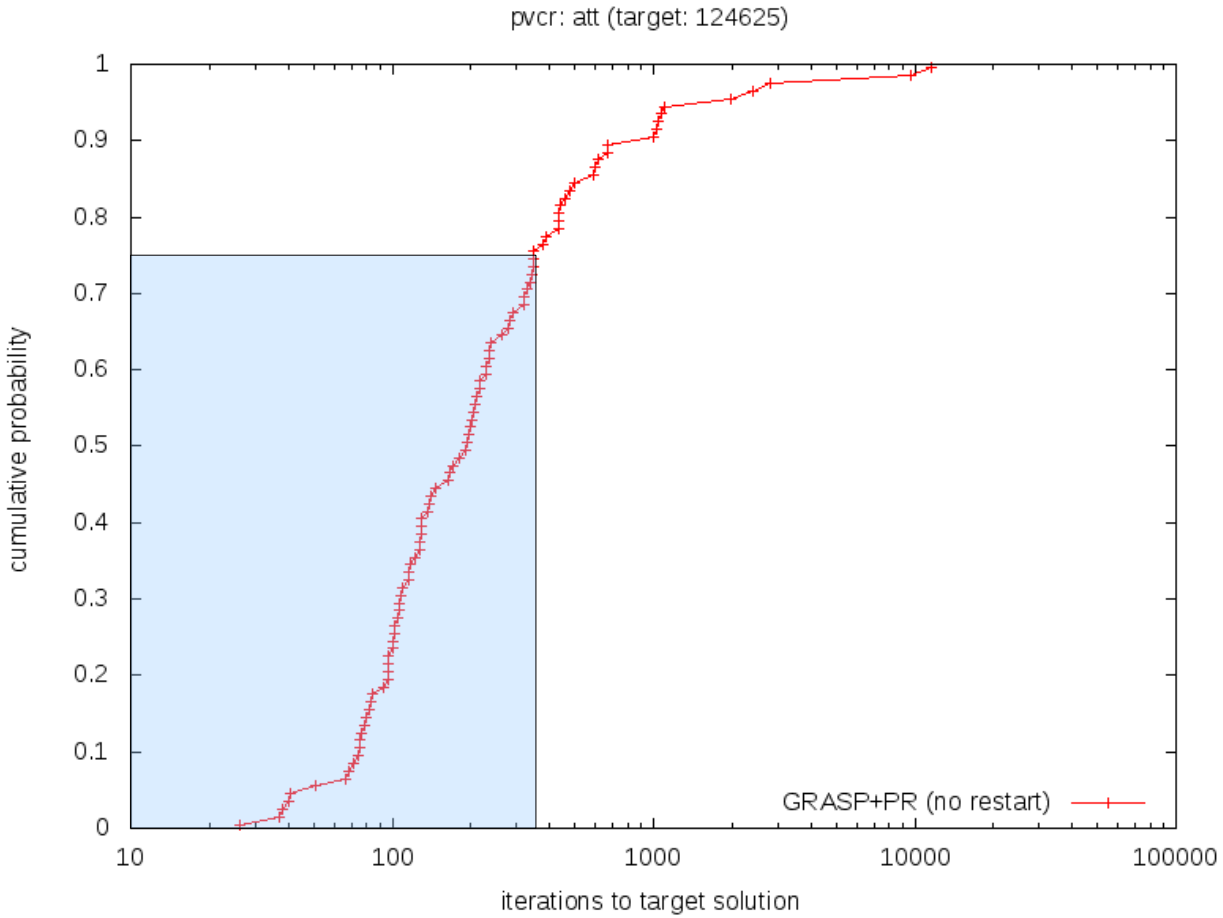
In most of the independent runs, the algorithm finds the target solution in relatively few iterations:



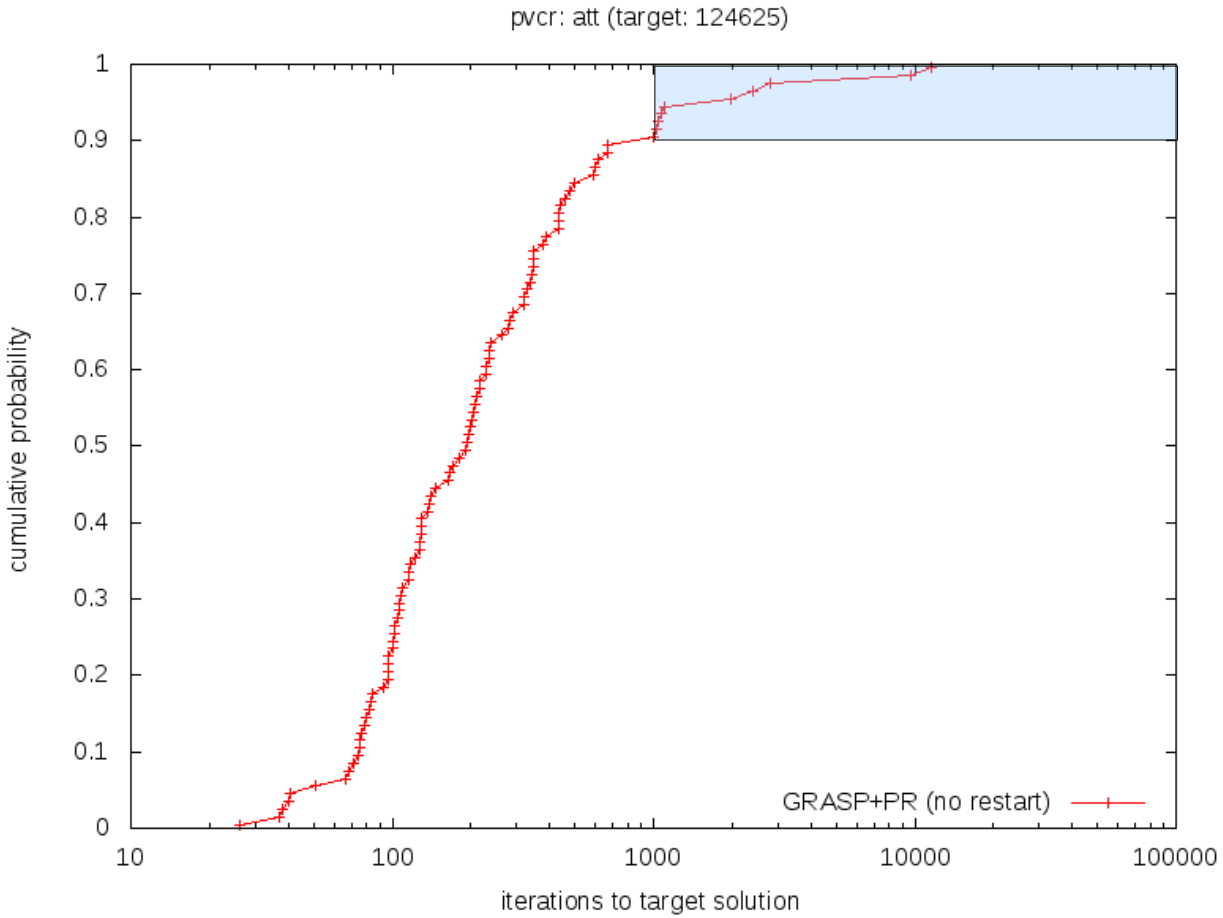
In most of the independent runs, the algorithm finds the target solution in relatively few iterations: 25% of the runs take fewer than 101 iterations



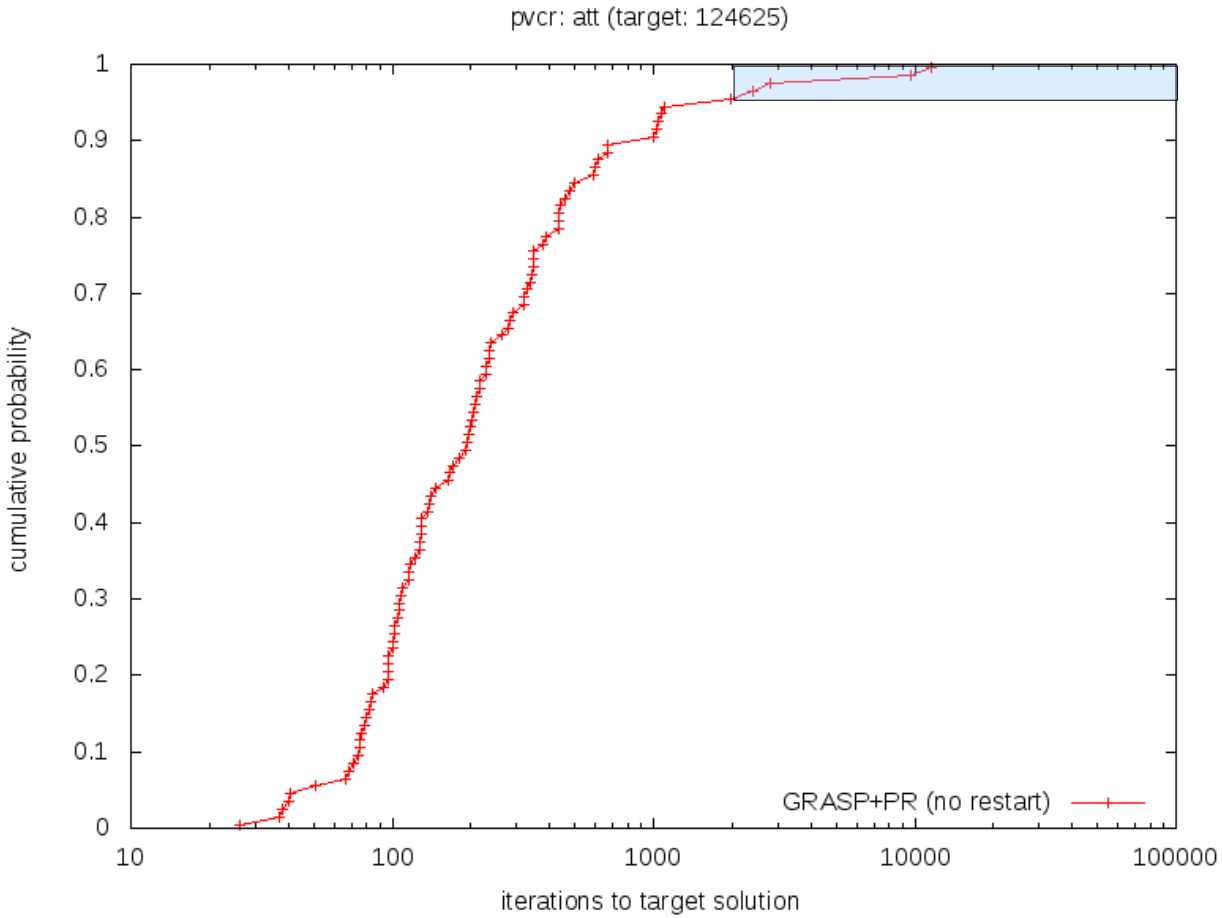
In most of the independent runs, the algorithm finds the target solution in relatively few iterations: 50% of the runs take fewer than 192 iterations



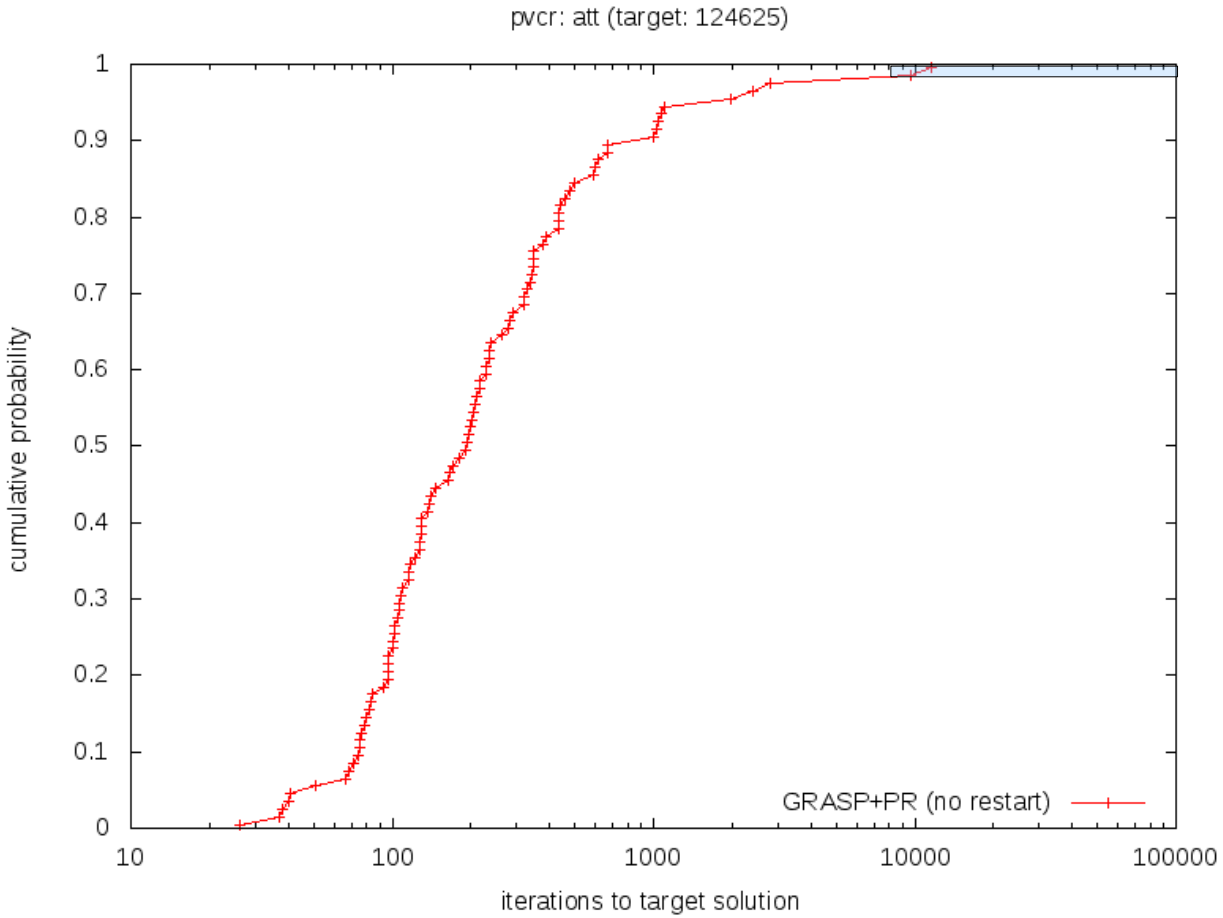
In most of the independent runs, the algorithm finds the target solution in relatively few iterations: 75% of the runs take fewer than 345 iterations



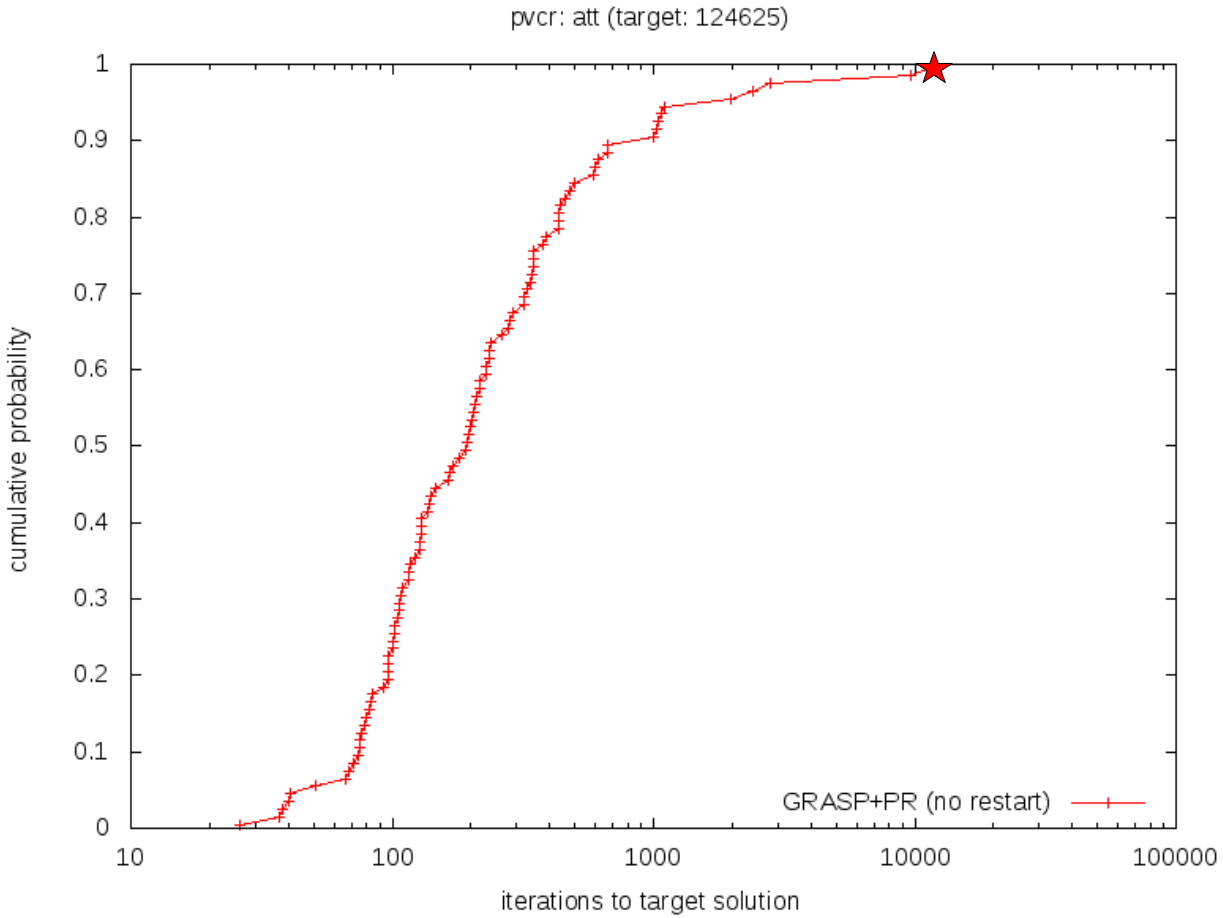
However, some runs take much longer: 10% of the runs take over 1000 iterations



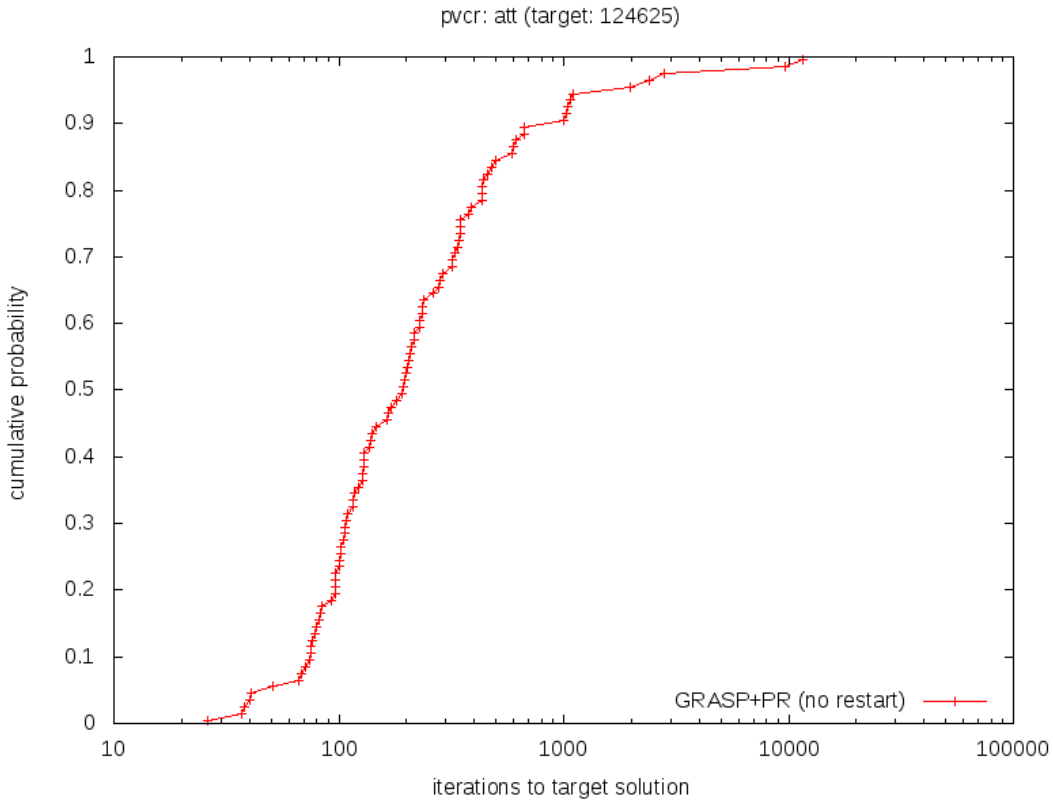
However, some runs take much longer: 5% of the runs take over 2000 iterations



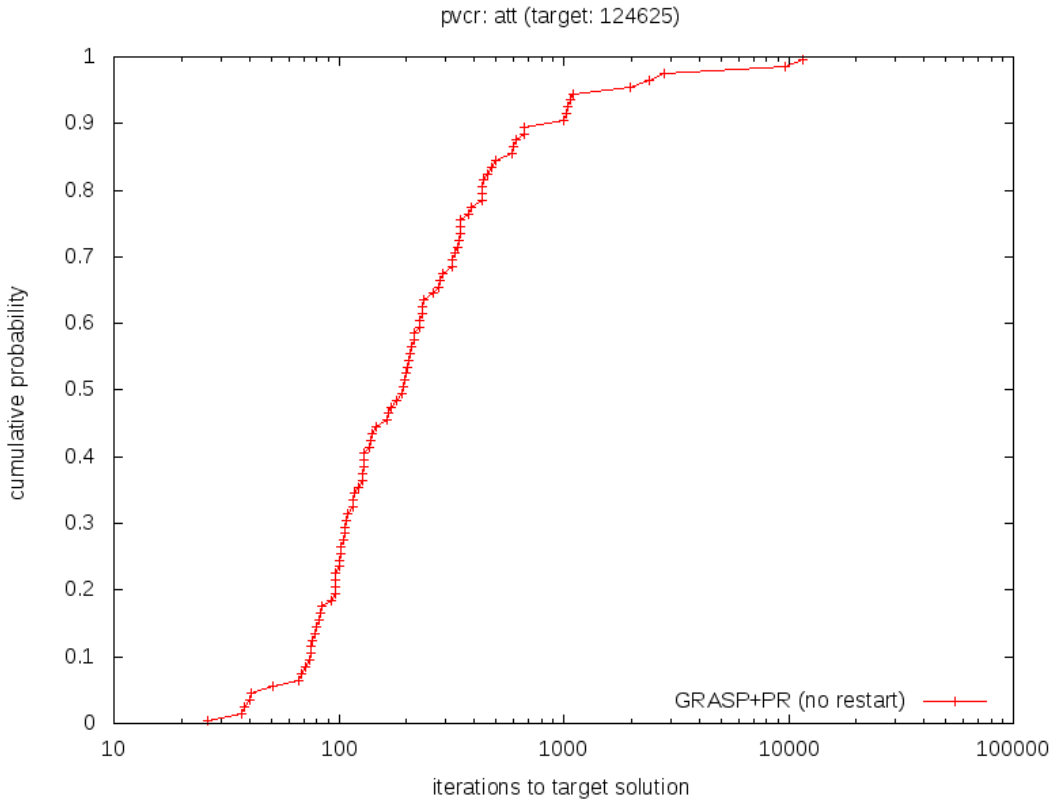
However, some runs take much longer: 2% of the runs take over 9715 iterations



However, some runs take much longer: the longest run took 11607 iterations



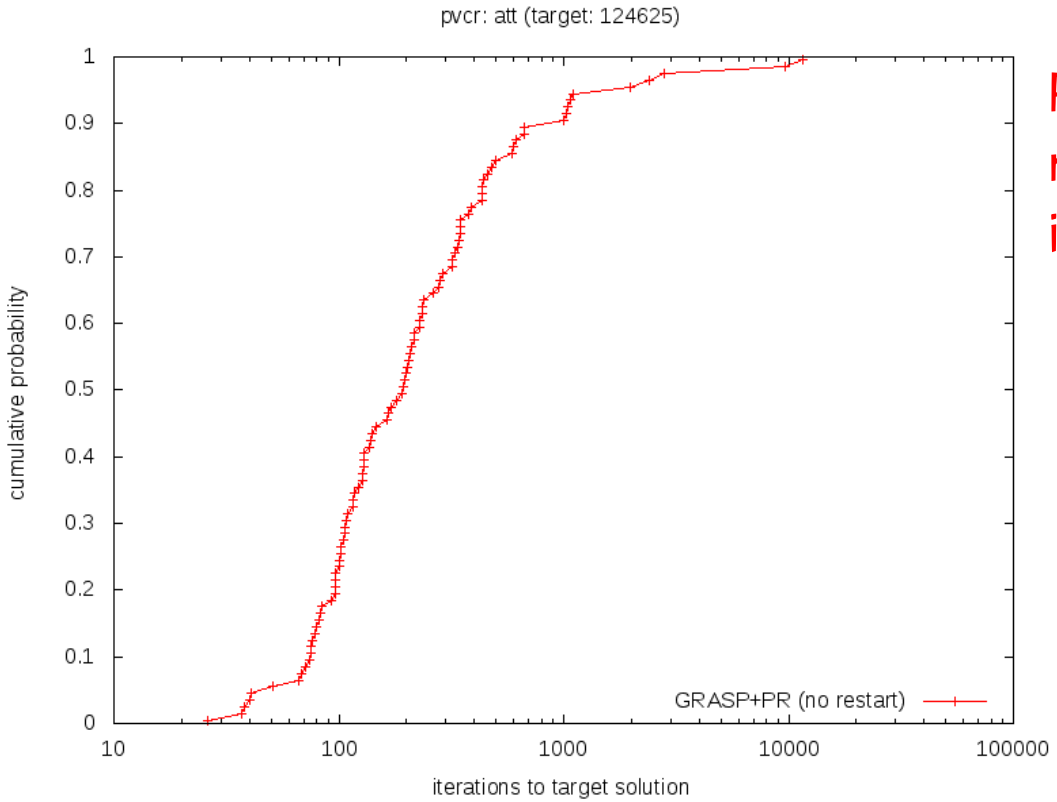
Probability that algorithm will take over 345 iterations: 25% = 1/4



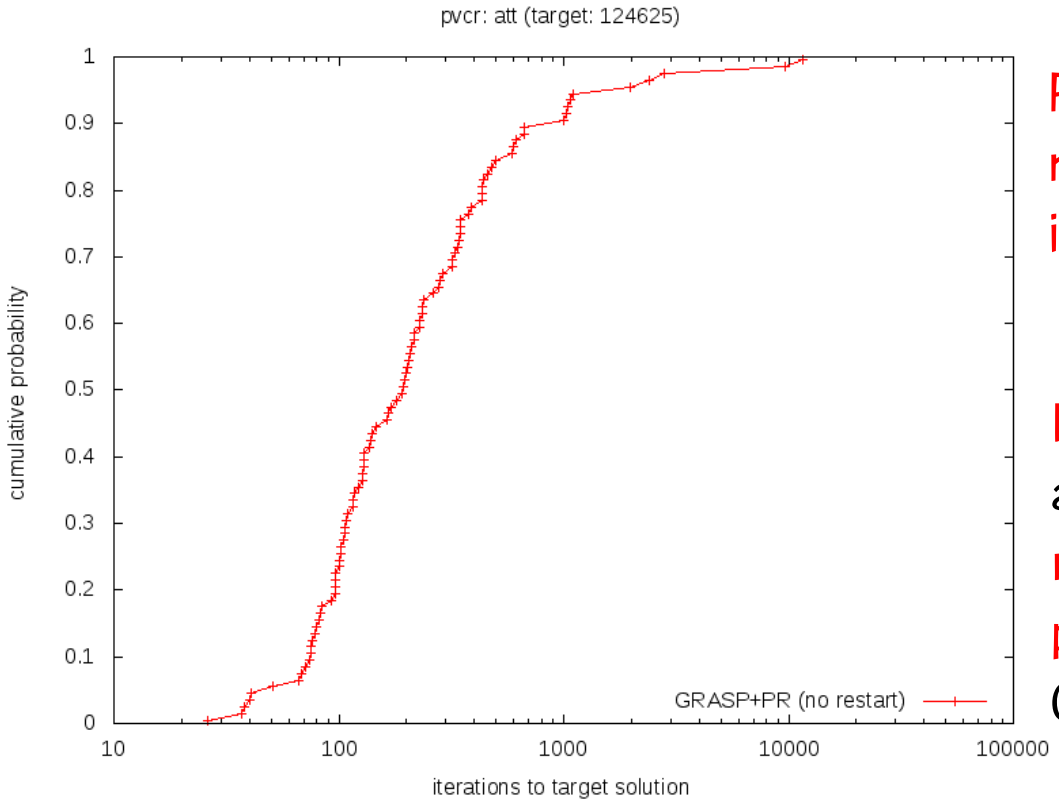
Probability that algorithm will take over 345 iterations: 25% = 1/4

By restarting algorithm after 345 iterations, probability that new run will take over 690 iterations: 25% = 1/4

Probability that algorithm with restart will take over 690 iterations: probability of taking over 345 X probability of taking over 690 iterations given it took over 345 = ¼ x ¼ = $1/4^2$

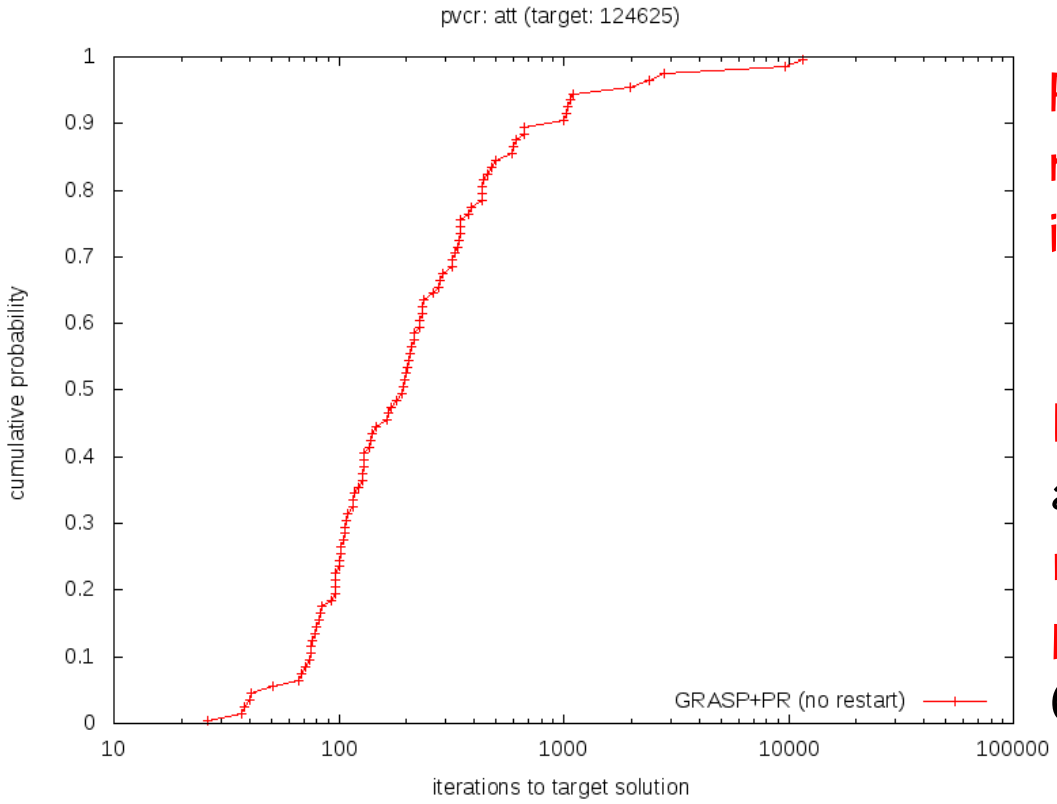


Probability that algorithm will still be running after K periods of 345 iterations: $1/4^K$



Probability that algorithm will still be running after K periods of 345 iterations: $1/4^K$

For example, probability that algorithm with restart will still be running after 1725 iterations (5 periods of 345 iterations): $1/4^5 \cong 0.0977\%$



Probability that algorithm will still be running after K periods of 345 iterations: $1/4^K$

For example, probability that algorithm with restart will still be running after 1725 iterations (5 periods of 345 iterations): $1/4^5 \cong 0.0977\%$

This is much less than the 5% probability that the algorithm without restart will take over 2000 iterations.

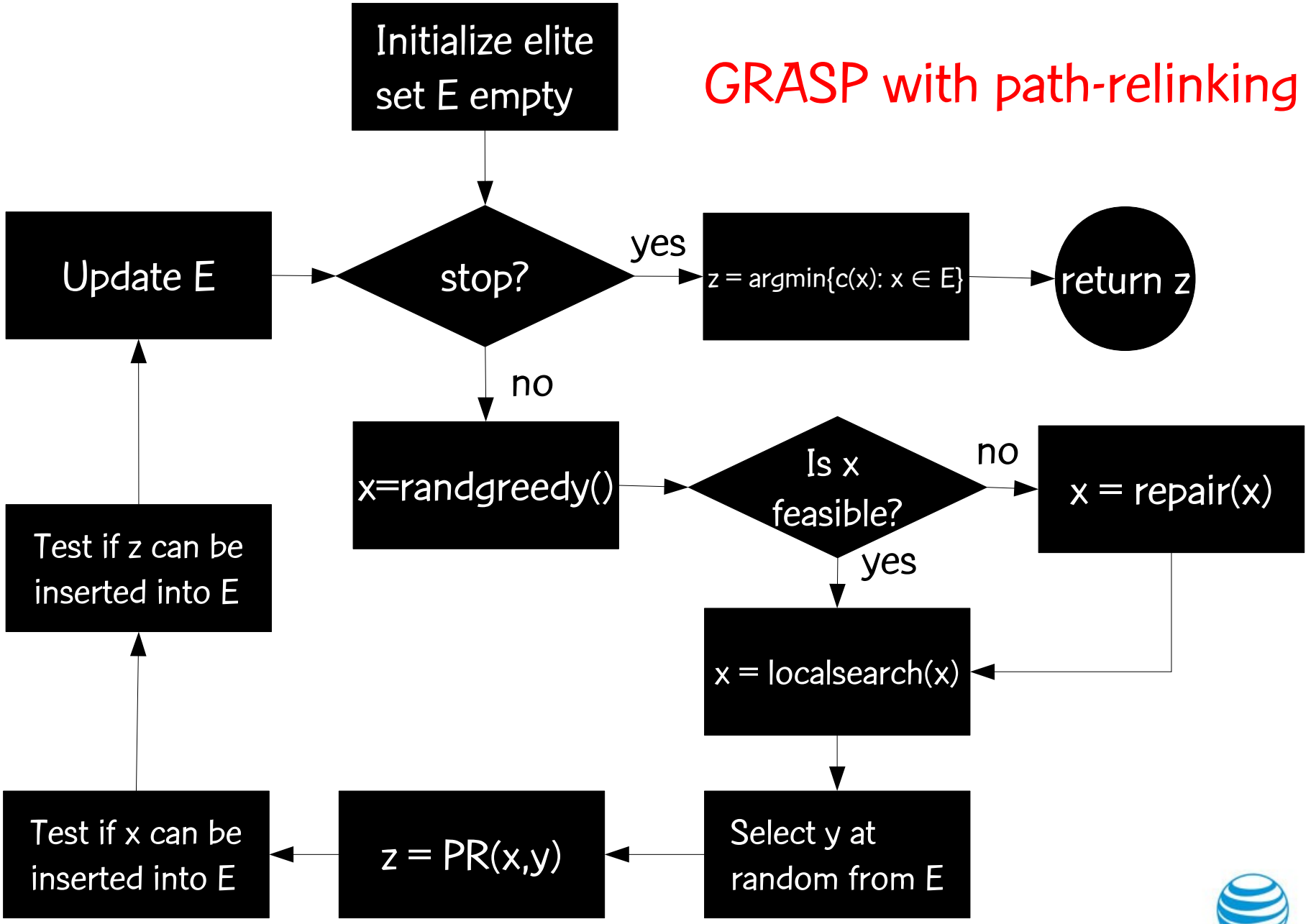
Restart strategies

- First proposed by Luby et al. (1993)
- They define a restart strategy as a finite sequence of time intervals $S = \{\tau_1, \tau_2, \tau_3, \dots\}$ which define epochs τ_1, $\tau_1+\tau_2$, $\tau_1+\tau_2+\tau_3$, ... when the algorithm is restarted from scratch.
- Luby et al. (1993) prove that the optimal restart strategy uses $\tau_1 = \tau_2 = \tau_3 = \cdots = \tau^*$, where τ^* is a constant.

Restart strategies

- Luby et al. (1993)
- Kautz et al. (2002)
- Palubeckis (2004)
- Sergienko et al. (2004)
- Nowicki & Smutnicki (2005)
- D'Apuzzo et al. (2006)
- Shylo et al. (2011a)
- Shylo et al. (2011b)

GRASP with path-relinking

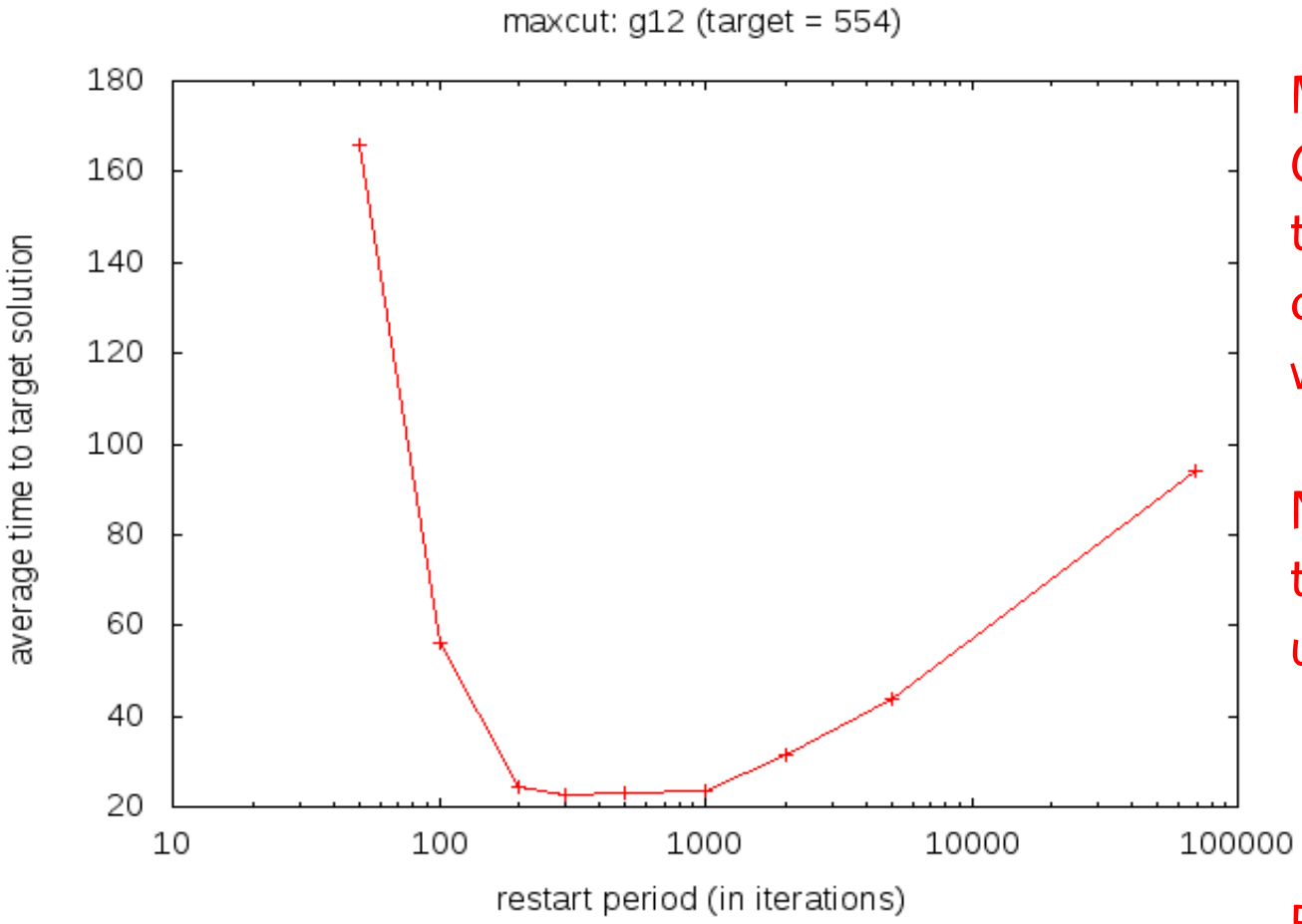


Restart strategy for GRASP with path-relinking

- Recall the restart strategy of Luby et al. where equal time intervals $\tau_1 = \tau_2 = \tau_3 = \cdots = \tau^*$ pass between restarts.
- Strategy requires τ^* as input.
- Since we have no prior information as to the runtime distribution of the heuristic, we run the risk of:
  - choosing τ^* too small: restart variant may take long to converge
  - choosing τ^* too big: restart variant may become like no-restart variant

Restart strategy for GRASP with path-relinking

- We conjecture that number of iterations between improvement of the incumbent (best so far) solution varies less w.r.t. heuristic/ instance/ target than run times.
- We propose the following restart strategy: Keep track of the last iteration when the incumbent improved and restart GRASP with path-relinking if K iterations have gone by without improvement.
- We call this strategy restart(K)



Maxcut instance G12: GRASP+PR was run 100 times stopping when cut of weight 554 or more was found.

Nine restart strategies of the type restart(K) were used.

Best values of K were between 200 and 1000.

Restarting GRASP with path-relinking

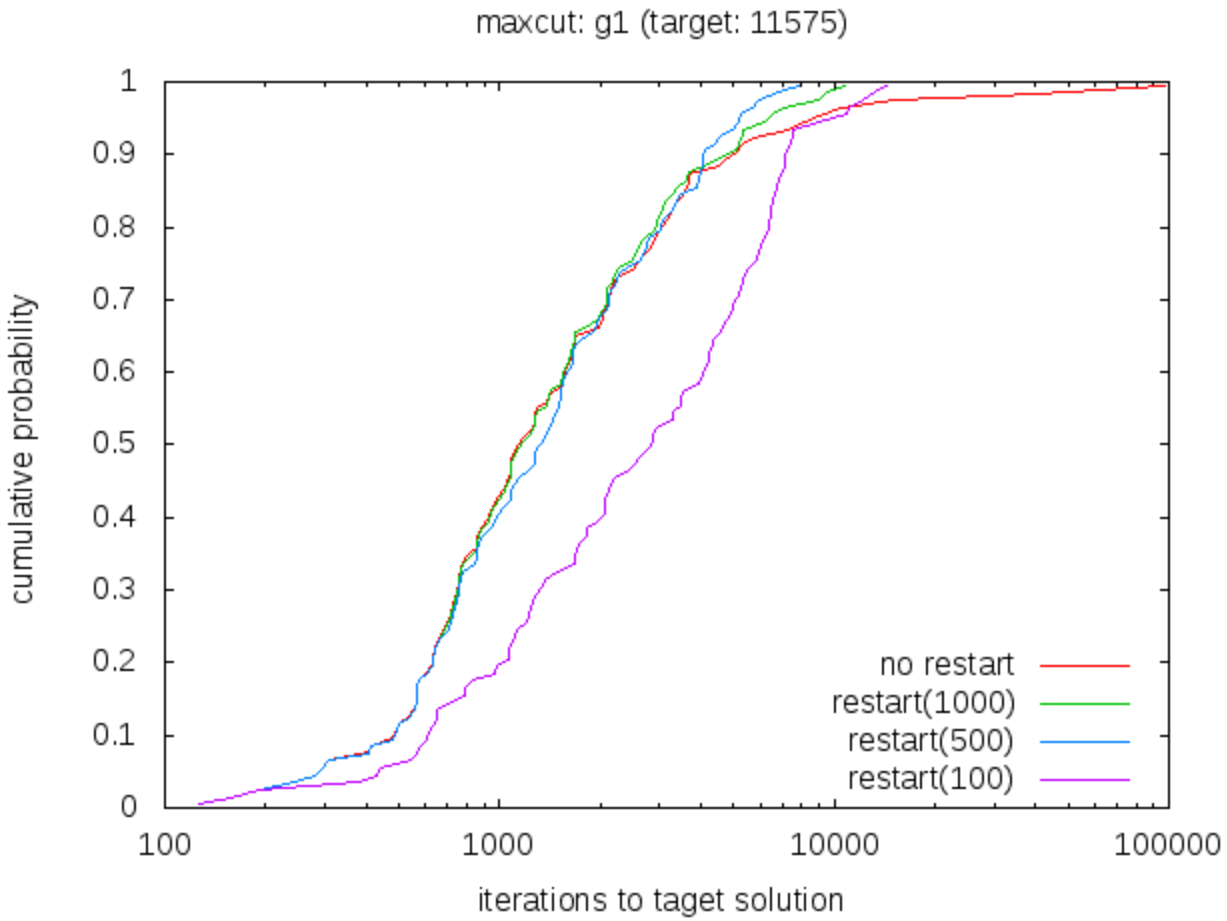
- Empty out elite set
- Discard incumbent
- Start new iteration with new seed for random number generator
- In practice, we could also input a maximum number of restarts and store overall best incumbent
  - We do not do this in the experiments, where runs only complete when a target solution is found

Restarting GRASP with path-relinking

- Strategy restart(K) also requires the parameter K to be input
- We will see in the experiments that even for heuristic/instance/target triplets with significantly different runtime distributions, a limited number of values for K almost always achieves the desired objective:
  - Reduce the average iteration count
  - Reduce standard deviation of the iteration count

Experimental design

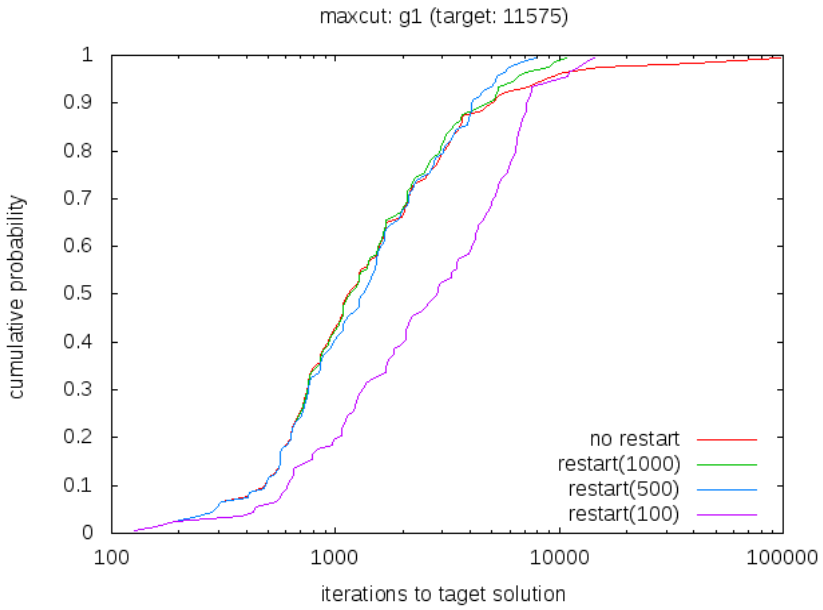
- Three GRASP + PR heuristics:
  - Max-cut (Festa et al., 2002)
  - Max-weighted SAT (Festa et al., 2006)
  - Private virtual circuit routing (R. & Ribeiro, 2003)
- For each heuristic we consider four variants:
  - No restart
  - restart(K), for K = 100, 500, and 1000
- Two instances for each problem:
  - Max-cut: G1 (target: 11575) and G12 (target: 554)
  - Max-weighted SAT: jnh1 (target: 420780) and jnh304 (target: 444125)
  - PVC routing: att (target: 124625) and fr750 (target: 2040000)
- Each heuristic was run 100 times for each instance, stopping when a solution at least as good as the target was found: total of 2400 runs



Problem: maxcut

Instance: G1

Target: 11575



Problem: maxcut
Instance: G1
Target: 11575

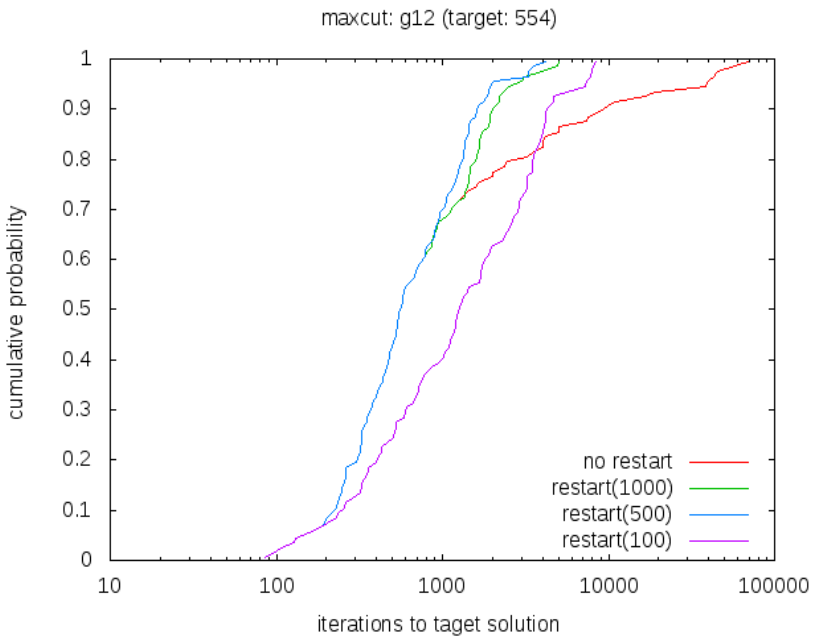
	restart strategy			
	no restart	1000	500	100
1st quartile	708	687	708	1120
2nd quartile	1145	1145	1292	2775
3rd quartile	2610	2270	2404	5557
4st quartile	96763	10753	7849	14343
average	3332	1944	1850	3673
std. dev.	10449	20212	1592	3054

Max # of iterations taken by all 100 runs in quartile.

Problem: maxcut

Instance: G12

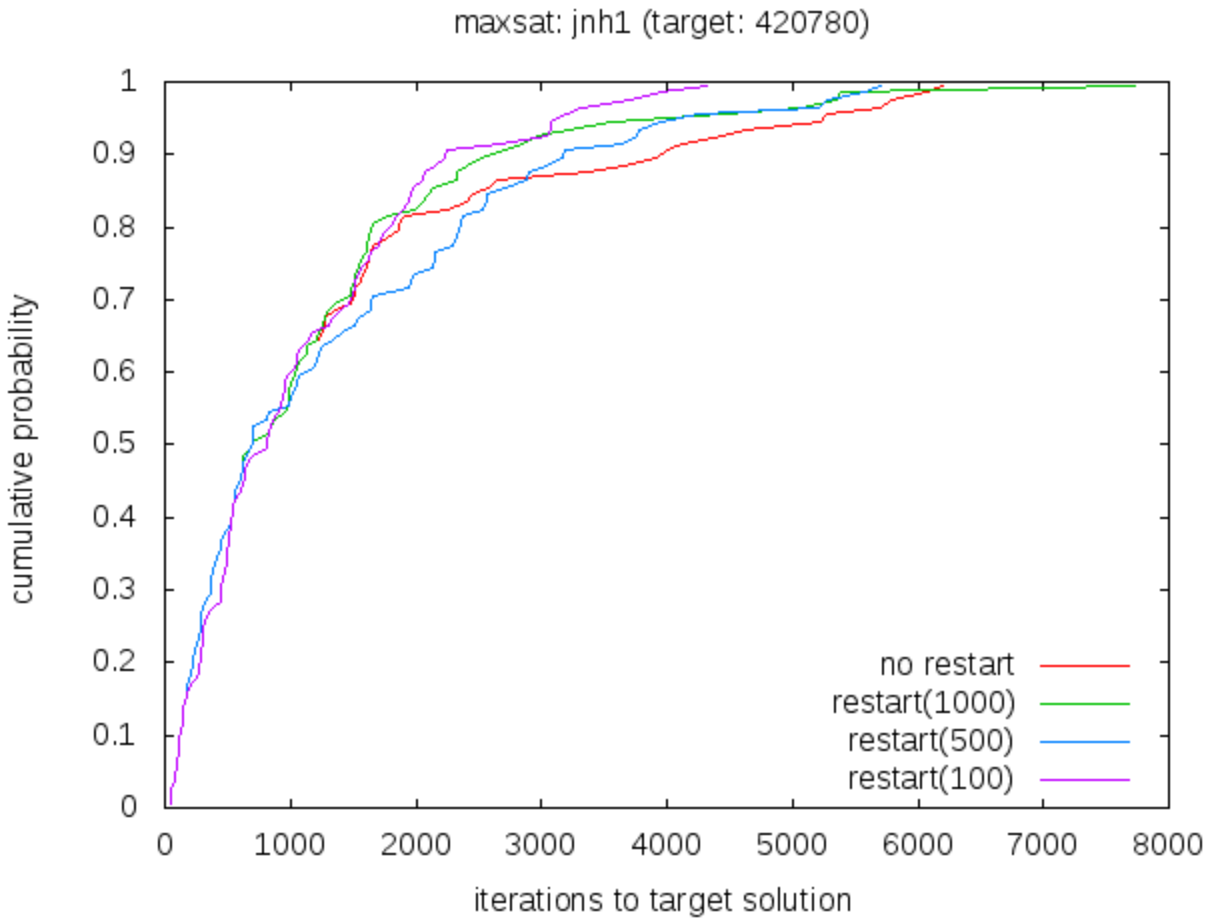
Target: 554



Problem: maxcut
Instance: G12
Target: 554

	restart strategy			
	no restart	1000	500	100
1st quartile	326	326	326	509
2nd quartile	550	550	550	1243
3rd quartile	1596	1423	1152	3247
4st quartile	68813	5014	4178	8382
average	4525	953	835	2055
std. dev.	11927	942	746	2006

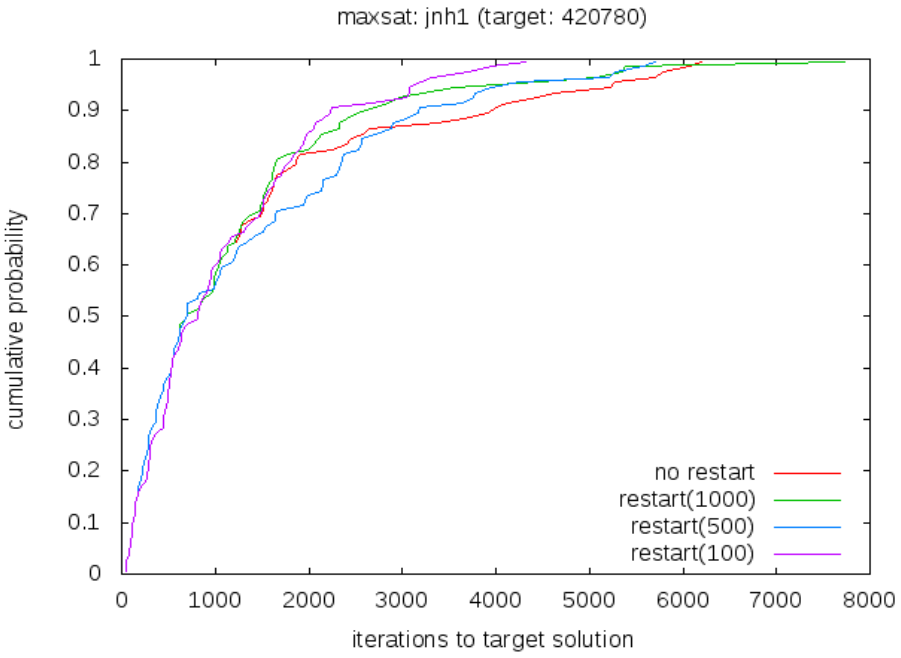
Max # of iterations taken by all 100 runs in quartile.



Problem: maxsat

Instance: jnh1

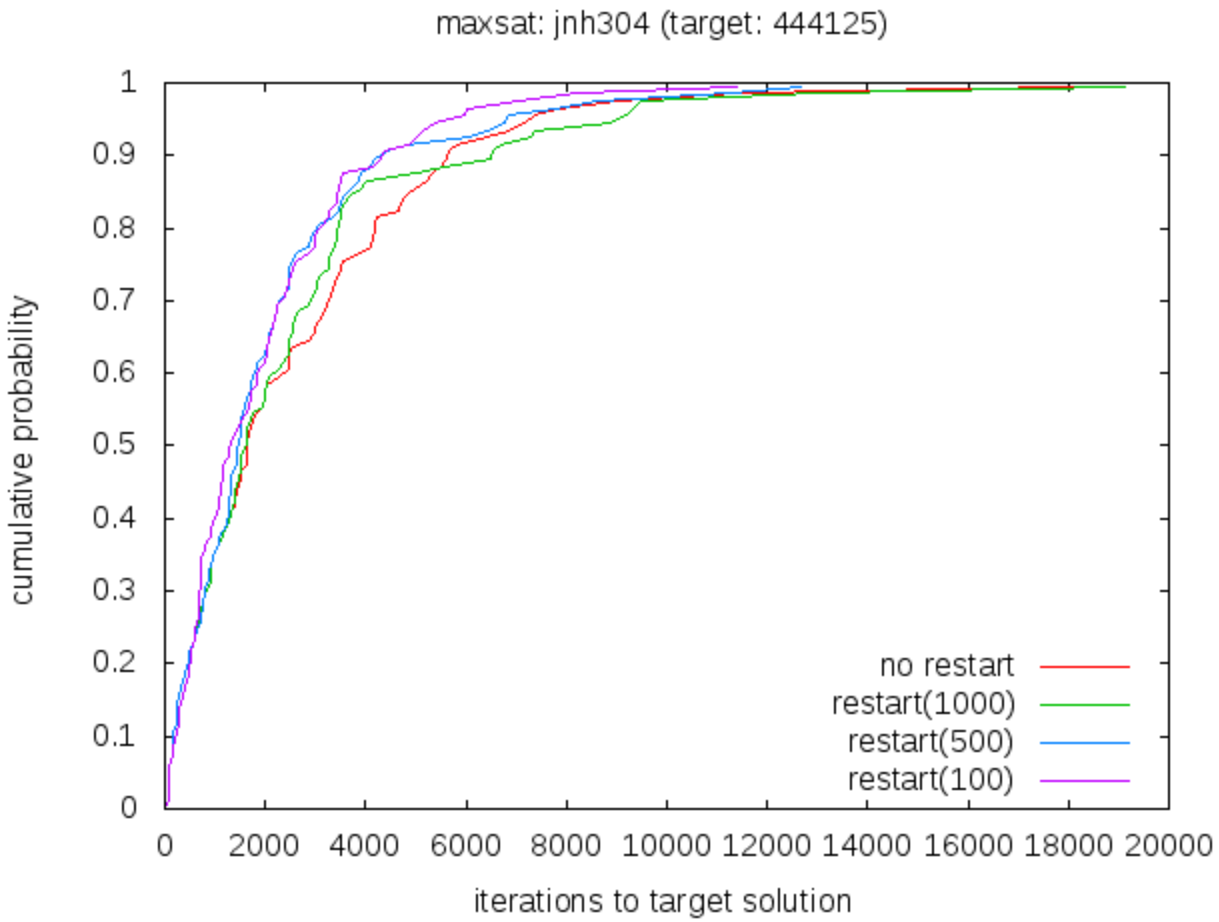
Target: 420780



Problem: maxsat
Instance: jnh1
Target: 420780

	restart strategy			
	no restart	1000	500	100
1st quartile	281	281	281	308
2nd quartile	684	684	684	308
3rd quartile	1611	1547	2142	1562
4st quartile	6206	7737	5708	4323
average	1320	1171	1309	1071
std. dev.	1522	1317	1364	961

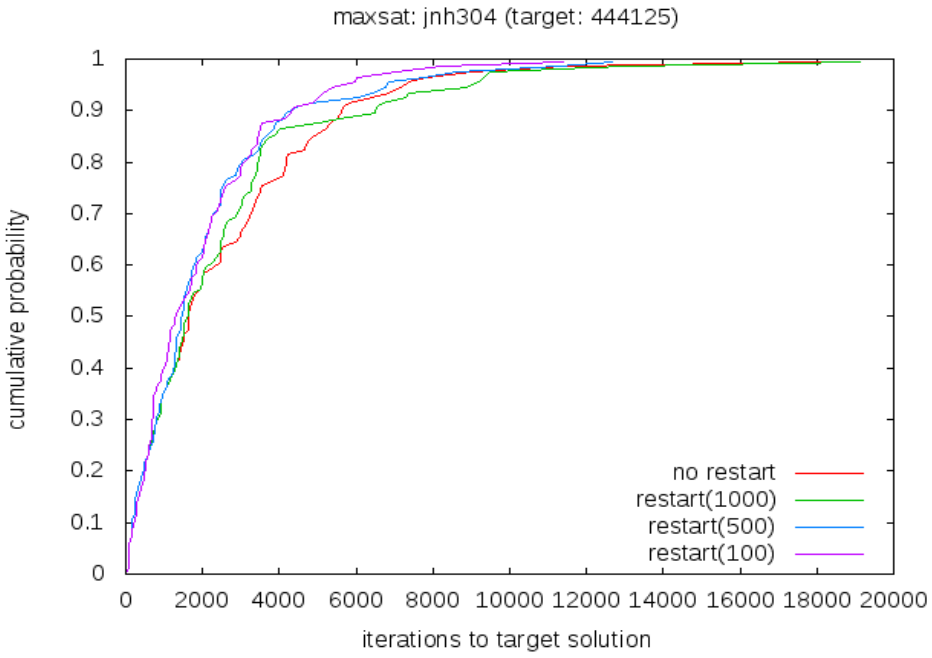
Max # of iterations taken by all 100 runs in quartile.



Problem: maxsat

Instance: jnh304

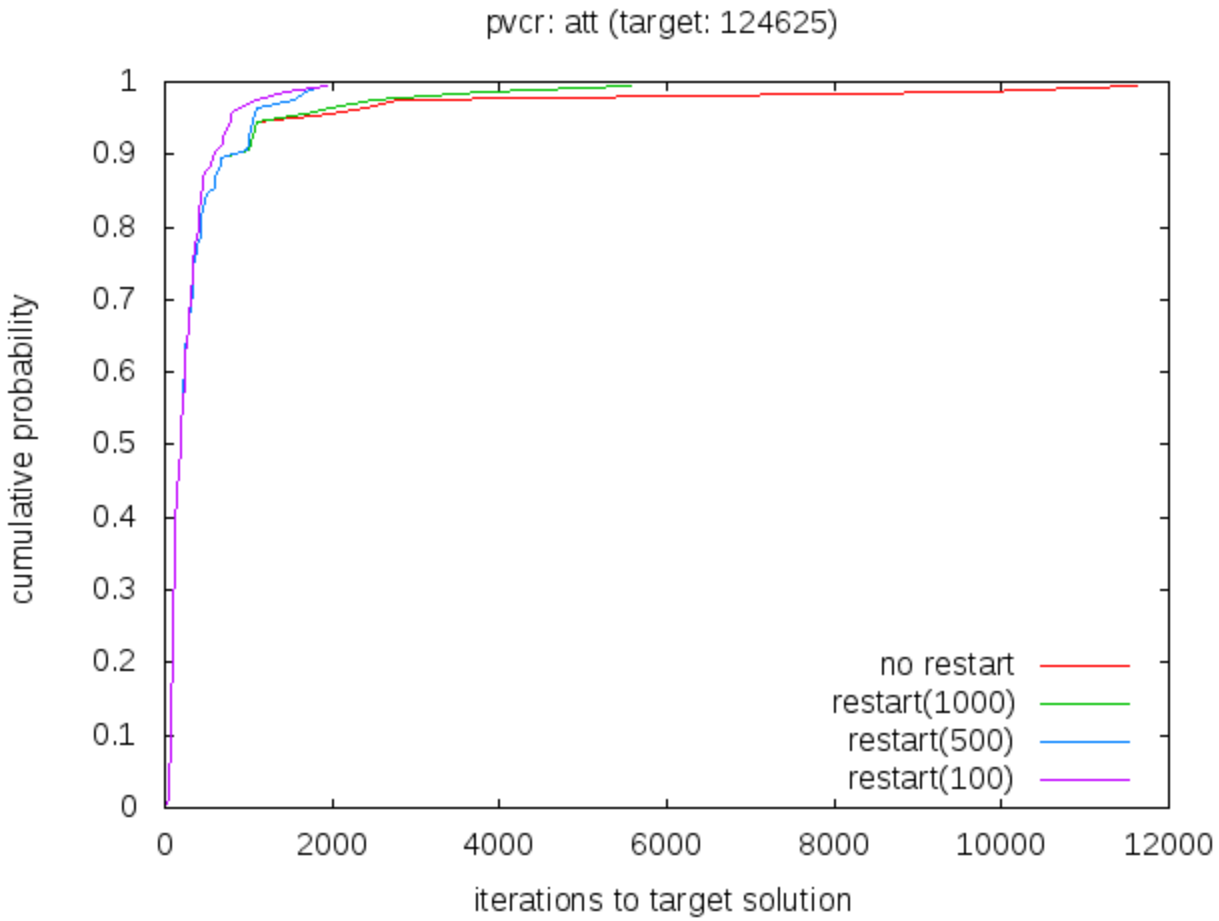
Target: 444125



Problem: maxsat
Instance: jnh304
Target: 444125

	restart strategy			
	no restart	1000	500	100
1st quartile	657	657	657	605
2nd quartile	1621	1610	1432	1266
3rd quartile	3488	3255	2483	2558
4st quartile	18095	19124	12651	11390
average	2546	2508	2092	1930
std. dev.	2738	2957	2247	1908

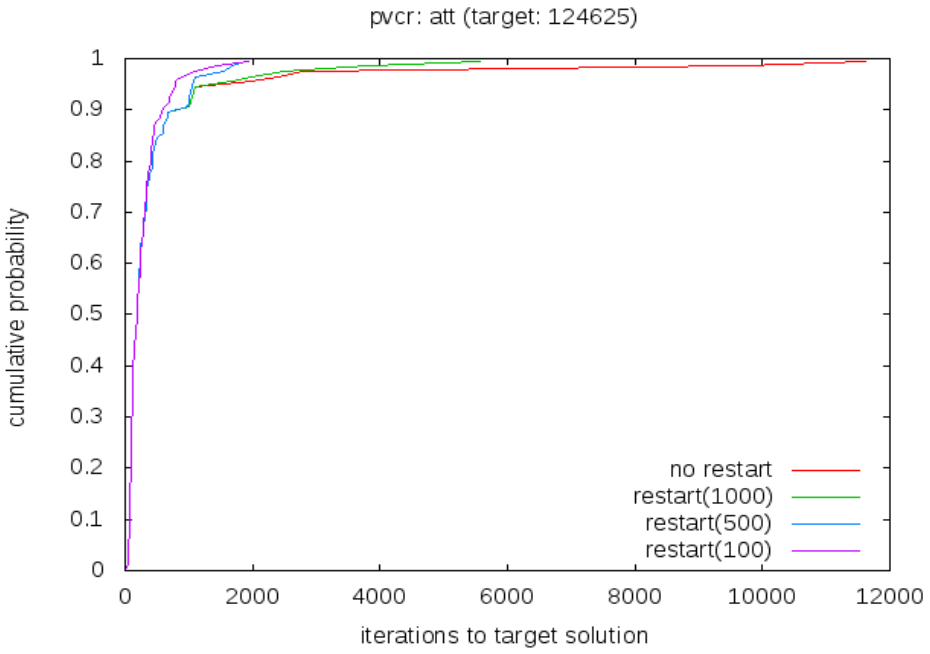
Max # of iterations taken by all 100 runs in quartile.



Problem: pvcr

Instance: att

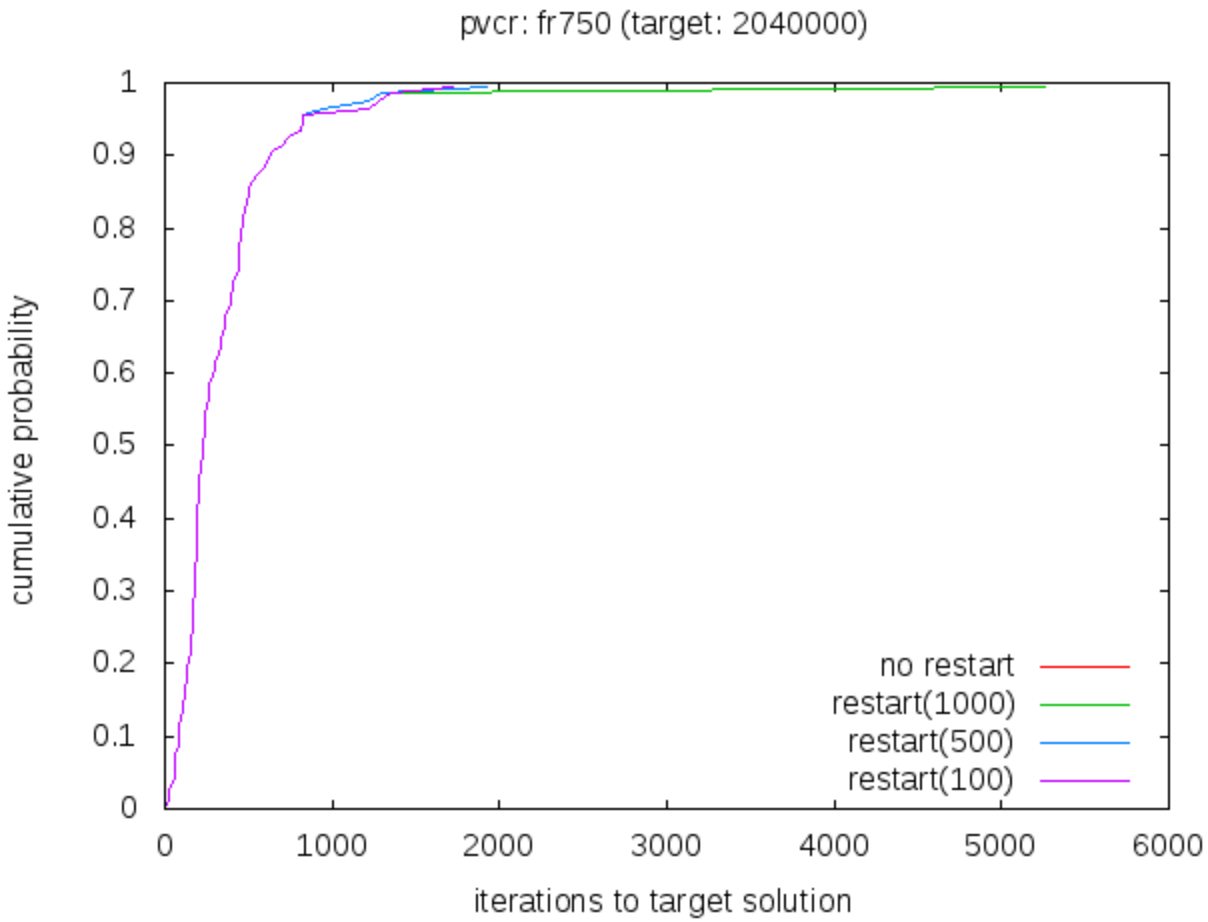
Target: 124625



Problem: pvcr
Instance: att
Target: 124625

	restart strategy			
	no restart	1000	500	100
1st quartile	101	101	101	101
2nd quartile	192	192	192	192
3rd quartile	345	345	345	345
4st quartile	11607	5567	1891	1948
average	527	398	314	278
std. dev.	1518	738	356	292

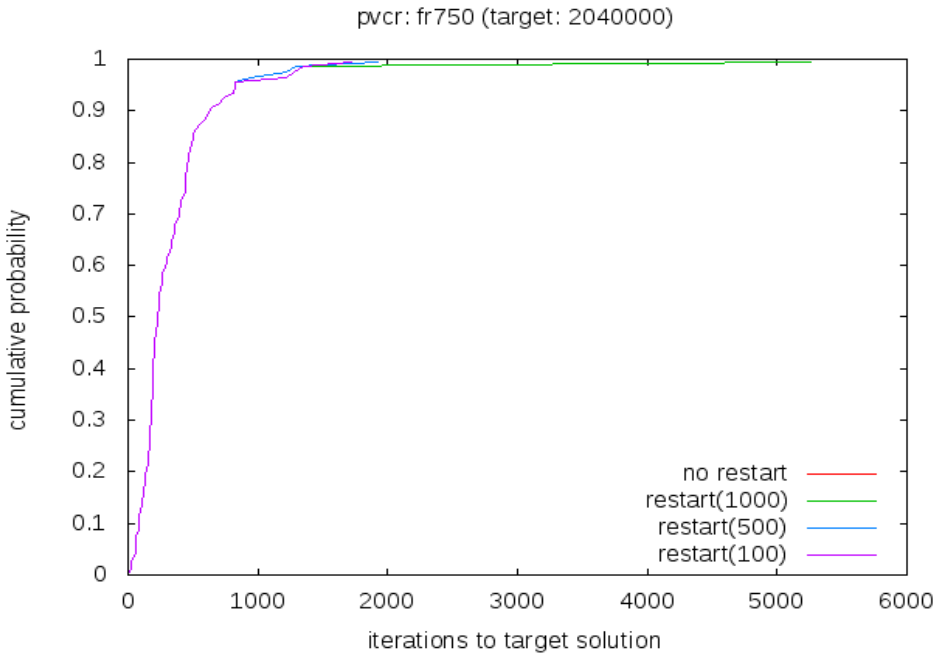
Max # of iterations taken by all 100 runs in quartile.



Problem: pvcr

Instance: fr750

Target: 2040000



Problem: pvcr
Instance: fr750
Target: 2040000

	restart strategy			
	no restart	1000	500	100
1st quartile	186	163	163	163
2nd quartile	223	223	223	223
3rd quartile	438	438	438	438
4st quartile	5260	5260	1924	1717
average	359	359	326	287
std. dev.	547	547	288	288

Max # of iterations taken by all 100 runs in quartile.

Analysis of experiments

- Main effect of restart strategies is observed in 4th quartile. Little effect in other quartiles.
- Compared to no restart, for all instances at least one restart strategy reduced the maximum number of iterations, the average number of iterations, and the standard deviation of number of iterations.

Analysis of experiments

- In only three (strategy/instance) pairs was a the maximum number of iterations not less than that taken by the no restart variant:

Analysis of experiments

- In only three (strategy/instance) pairs was a the maximum number of iterations not less than that taken by the no restart variant:
  - Restart(1000) on maxsat instance jnh1

Analysis of experiments

- In only three (strategy/instance) pairs was a the maximum number of iterations not less than that taken by the no restart variant:
  - Restart(1000) on maxsat instance jnh1 (avg and std. dev. were reduced)

Analysis of experiments

- In only three (strategy/instance) pairs was a the maximum number of iterations not less than that taken by the no restart variant:
  - Restart(1000) on maxsat instance jnh1(avg and std. dev. were reduced)
  - Restart(1000) on maxsat instance jnh304

Analysis of experiments

- In only three (strategy/instance) pairs was a the maximum number of iterations not less than that taken by the no restart variant:
  - Restart(1000) on maxsat instance jnh1(avg and std. dev. were reduced)
  - Restart(1000) on maxsat instance jnh304 (avg and std. dev. were reduced)

Analysis of experiments

- In only three (strategy/instance) pairs was a the maximum number of iterations not less than that taken by the no restart variant:
  - Restart(1000) on maxsat instance jnh1(avg and std. dev. were reduced)
  - Restart(1000) on maxsat instance jnh304 (avg and std. dev. were reduced)
  - Restart(1000) on pvcr instance fr750

Analysis of experiments

- In only three (strategy/instance) pairs was a the maximum number of iterations not less than that taken by the no restart variant:
  - Restart(1000) on maxsat instance jnh1(avg and std. dev. were reduced)
  - Restart(1000) on maxsat instance jnh304 (avg and std. dev. were reduced)
  - Restart(1000) on pvcr instance fr750 (no restart was done since the no-restart variant never took over 1000 iterations)

Analysis of experiments

- In only a single (strategy/instance) pair was the average number of iterations greater than that of the no-restart strategy:

Analysis of experiments

- In only a single (strategy/instance) pair was the average number of iterations greater than that of the no-restart strategy:
  - Restart(100) on maxcut instance G1

Analysis of experiments

- In only a single (strategy/instance) pair was the average number of iterations greater than that of the no-restart strategy:
  - Restart(100) on maxcut instance G1 (about 10% more iterations)

Analysis of experiments

- In only a single (strategy/instance) pair was the standard deviation of number of iterations greater than that of the no-restart strategy:

Analysis of experiments

- In only a single (strategy/instance) pair was the standard deviation of number of iterations greater than that of the no-restart strategy:
  - Restart(1000) on maxsat instance jnh304

Analysis of experiments

- In only a single (strategy/instance) pair was the standard deviation of number of iterations greater than that of the no-restart strategy:
  - Restart(1000) on maxsat instance jnh304 (about 8% greater)

Analysis of experiments

- In only a single (strategy/instance) pair was the standard deviation of number of iterations greater than that of the no-restart strategy:
  - Restart(1000) on maxsat instance jnh304 (about 8% greater)

Analysis of experiments

- Compared to the no-restart strategy, restart(1000):

Analysis of experiments

- Compared to the no-restart strategy, restart(1000):
  - Reduced the max, average, and std. dev. of the iteration count for maxcut instances G1 and G12, and pvcr instance att.

Analysis of experiments

- Compared to the no-restart strategy, restart(1000):
  - Reduced the max, average, and std. dev. of the iteration count for maxcut instances G1 and G12, and pvcr instance att.
  - Increased the max number of iterations on maxsat instrances jnh1 and jnh304.

Analysis of experiments

- Compared to the no-restart strategy, restart(1000):
  - Reduced the max, average, and std. dev. of the iteration count for maxcut instances G1 and G12, and pvcr instance att.
  - Increased the max number of iterations on maxsat instrances jnh1 and jnh304.
  - Increased the std. dev. of the number of iterations for maxsat instance jnh304.

Analysis of experiments

- Compared to the no-restart strategy, restart(1000):
  - Reduced the max, average, and std. dev. of the iteration count for maxcut instances G1 and G12, and pvcr instance att.
  - Increased the max number of iterations on maxsat instrances jnh1 and jnh304.
  - Increased the std. dev. of the number of iterations for maxsat instance jnh304.
  - On pvcr instance fr750, restart was not activated a single time.

Analysis of experiments

- Compared to the no-restart strategy, restart(500):

Analysis of experiments

- Compared to the no-restart strategy, restart(500):
  - Reduced the max, average, and std. dev. of number of iterations for all instances.

Analysis of experiments

- Compared to the no-restart strategy, restart(500):
  - Reduced the max, average, and std. dev. of number of iterations for all instances.
- Restart(100) did so, too, for all but one instance (maxcut instance G1) where it had a larger average number of iterations.

Analysis of experiments

- Restart(500) was clearly the best strategy for maxcut instance G1 and G12

Analysis of experiments

- Restart(500) was clearly the best strategy for maxcut instance G1 and G12
- Restart(100) was best for maxsat instances jnh1 and jnh304

Analysis of experiments

- Restart(500) was clearly the best strategy for maxcut instance G1 and G12
- Restart(100) was best for maxsat instances jnh1 and jnh304
- On both pvcr instances restart(100) and restart(500) were better than restart(1000).

Analysis of experiments

- Restart(500) was clearly the best strategy for maxcut instance G1 and G12
- Restart(100) was best for maxsat instances jnh1 and jnh304
- On both pvcr instances restart(100) and restart(500) were better than restart(1000).
  - Restart(500) reduced the max number of iterations more than restart(100)

Analysis of experiments

- Restart(500) was clearly the best strategy for maxcut instance G1 and G12
- Restart(100) was best for maxsat instances jnh1 and jnh304
- On both pvcr instances restart(100) and restart(500) were better than restart(1000).
  - Restart(500) reduced the max number of iterations more than restart(100)
  - Restart(100) reduced the average and std. Dev. More than restart(500)

Concluding remarks

Concluding remarks

- We proposed new restart strategies for GRASP with path-relinking heuristics based on the number of iterations without improvement of the incumbent solution.

Concluding remarks

- We proposed new restart strategies for GRASP with path-relinking heuristics based on the number of iterations without improvement of the incumbent solution.
- Tested the strategies on instances for which the average and maximum number of iterations of the no-restart strategy varied from 359 to 4525 and 5260 to 96763, respectively.

Concluding remarks

- While no restart strategy increased all three performance measures (max, average, std. dev.) for a single instance,

Concluding remarks

- While no restart strategy increased all three performance measures (max, average, std. dev.) for a single instance,
  - Restart(500) decreased all three measures for all instances

Concluding remarks

- While no restart strategy increased all three performance measures (max, average, std. dev.) for a single instance,
  - Restart(500) decreased all three measures for all instances
  - Restart(100) increased a single measure for a single instance.

Concluding remarks

- While no restart strategy increased all three performance measures (max, average, std. dev.) for a single instance,
  - Restart(500) decreased all three measures for all instances
  - Restart(100) increased a single measure for a single instance.
- Overall restart(500) was the best restart strategy.

Concluding remarks

- These results are valid for the implementations, instances, and target values used in the experiments.

Concluding remarks

- These results are valid for the implementations, instances, and target values used in the experiments.
- Though we conjecture they will be valid for other problems, instances, and target values, further testing is needed.

Concluding remarks

- These results are valid for the implementations, instances, and target values used in the experiments.
- Though we conjecture they will be valid for other problems, instances, and target values, further testing is needed.
- We plan to extend our study to GRASP+PR heurisitics for the generalized quadratic assignment problem and the antibandwidth problem.

My coauthor (Celso Ribeiro) with Brazil's President Lula in 2006.

The End

These slides and all papers cited in this talk
can be downloaded from my homepage:
http://mauricioresende.com