GRASP with path-relinking

Celso C. Ribeiro (celso@ic.uff.br)

Universidade Federal Fluminense

Metaheuristics – May 24, 2019

Overview of talk

- Memoryless GRASP
- Elite sets
- Hybridization of GRASP with path-relinking
- Evolutionary path-relinking
- Restart strategies

Introduction

- Path-relinking is a major enhancement to GRASP, adding a long-term memory mechanism to GRASP heuristics.
- GRASP with path-relinking implements long-term memory using an elite set of diverse high-quality solutions found during the search.
- In its most basic implementation, at each iteration the path-relinking operator is applied between the solution found at the end of the local search phase and a randomly-selected solution from the elite set.
- The solution resulting from path-relinking is a candidate for inclusion in the elite set.
- In this presentation we examine elite sets, their integration with GRASP, the basic GRASP with path-relinking procedure, several variants of the basic scheme, including evolutionary path-relinking, and restart strategies for GRASP with path-relinking heuristics.

Memoryless GRASP

- The basic GRASP heuristic, as presented previously, searches the solution space by repeatedly applying independent searches in the solution space graph $\mathcal{G} = (F, M)$, each search starting from a different greedy randomized solution.
- Each independent search uses no information produced by any other search performed at previous iterations.
- The choices of starting solutions for local search are not influenced by information produced during the search.
- However, Reactive GRASP and adaptive memory techniques do make use of information produced during the search.
- Reactive GRASP does so to select the blend of randomness and greediness used in the construction of the starting solutions for local search, while programming with adaptive memory determines the amount of intensification and diversification in the construction phase.

Memoryless GRASP

- The memoryless nature of basic, or pure, GRASP is in contrast with many successful metaheuristics, such as
  - tabu search,
  - genetic algorithms, and
  - ant colony optimization,

  which make extensive use of information gathered during the search process to guide their choice of the region of the solution space to explore.

- In this presentation, we show how path-relinking can be used with any GRASP heuristic to result in a hybrid procedure with a long-term memory mechanism.

- Given the same running time, this hybridization almost always produces better solutions than pure GRASP.

- Alternatively, given a target value, it almost always finds a solution at least as good as this target in less running time than pure GRASP.

Elite sets

- An elite set $\mathcal{E}$ of solutions is a set formed by at most a fixed number $n_{\mathcal{E}}$ of diverse, high-quality solutions found during the run of a heuristic.
- The elite solutions should represent distinct promising regions of the solution space and therefore should not include solutions that are too similar, even if they are of high quality.

- The pseudo-code shows a template to maintain an elite set $\mathcal{E}$ of at most $n_{\mathcal{E}}$ elements for a minimization problem.
- The algorithm is given a candidate solution S and determines if S should be added to $\mathcal{E}$ and, if so, which solution, if any, should be removed from $\mathcal{E}$.
- Let the *symmetric difference* $\Delta(S, S')$ be formed by the ground set elements that belong to either S or S'.

begin UPDATE-ELITE-SET$(S, \mathcal{E})$;
  if $|\mathcal{E}| < n_{\mathcal{E}}$ **then**
    if $\mathcal{E} = \varnothing$ **then**
      $\mathcal{E} \leftarrow \mathcal{E} \cup \{S\}$;
    else
      $\delta \leftarrow \min\{|\Delta(S, S')| \ : \ S' \in \mathcal{E}\}$;
      if $\delta > 0$ **then** $\mathcal{E} \leftarrow \mathcal{E} \cup \{S\}$;
    end-if;
  else
    $f^+ \leftarrow \max\{f(S') \ : \ S' \in \mathcal{E}\}$;
    $\delta \leftarrow \min\{|\Delta(S, S')| \ : \ S' \in \mathcal{E}\}$;
    if $f(S) < f^+$ **and** $\delta > 0$ **then**
      $S^- \leftarrow \mathrm{argmin}\{|\Delta(S, S')| \ : \ S' \in \mathcal{E}$ such that $f(S') \geq f(S)\}$;
      $\mathcal{E} \leftarrow \mathcal{E} \cup \{S\} \setminus \{S^-\}$;
    end-if;
  end-if;
  return $\mathcal{E}$;
end UPDATE-ELITE-SET.

- Our goal is to first improve the average quality of the elite set, and then maximize the diversity of its elements.

- The algorithm can be modified to increase the diversity of the elite set solutions by modifying lines 6 and 11, where condition $\delta > 0$ can be changed to $\delta \geq \underline{\delta}$, where $\underline{\delta} > 0$ is a parameter.

- In this case, instead of requiring that S only be different from all other elite-set solutions, we now require that it be sufficiently different by at least a given number of attributes.

begin UPDATE-ELITE-SET($S, \mathcal{E}$);
1 **if** $|\mathcal{E}| < n_{\mathcal{E}}$ **then**
2   **if** $\mathcal{E} = \varnothing$ **then**
3         $\mathcal{E} \leftarrow \mathcal{E} \cup \{S\}$;
4   **else**
5         $\delta \leftarrow \min\{|\Delta(S, S')| \ : \ S' \in \mathcal{E}\}$;
6   **if** $\delta > 0$ **then** $\mathcal{E} \leftarrow \mathcal{E} \cup \{S\}$;
7   **end-if**;
8 **else**
9     $f^+ \leftarrow \max\{f(S') \ : \ S' \in \mathcal{E}\}$;
10     $\delta \leftarrow \min\{|\Delta(S, S')| \ : \ S' \in \mathcal{E}\}$;
11   **if** $f(S) < f^+$ **and** $\delta > 0$ **then**
12         $S^- \leftarrow \mathrm{argmin}\{|\Delta(S, S')| \ : \ S' \in \mathcal{E}$ such that $f(S') \geq f(S)\}$;
13         $\mathcal{E} \leftarrow \mathcal{E} \cup \{S\} \setminus \{S^-\}$;
14   **end-if**;
15 **end-if**;
16 **return** $\mathcal{E}$;
end UPDATE-ELITE-SET.

Hybridization of GRASP with path-relinking

- Path-relinking is a major enhancement to GRASP, equipping GRASP heuristics with a long-term memory mechanism and enabling search intensification beyond simple local search.
- To implement GRASP with path-relinking, we make use of an elite set $\mathcal{E}$ to collect a diverse set of high-quality solutions found during the search.
- The elite set starts empty and is constrained to have at most $n_{\mathcal{E}}$ solutions.
- Each new locally optimal solution produced by the GRASP local search phase is relinked with one or more solutions from the elite set.
- Each solution resulting from path-relinking is considered as a candidate to be inserted in the elite set according to previous algorithm UPDATE-ELITE-SET.

Hybridization of GRASP with path-relinking

- The pseudo-code outlines the main steps of a GRASP with path-relinking heuristic for minimization.
- This simple variant relinks the locally optimal solution produced in each GRASP iteration with a single, randomly chosen, solution from the elite set, following the forward path-relinking strategy.
- The output of the path-relinking operator is a candidate for inclusion in the elite set.
- The algorithm returns the best-valued elite solution in line 15, after a stopping criterion is met.

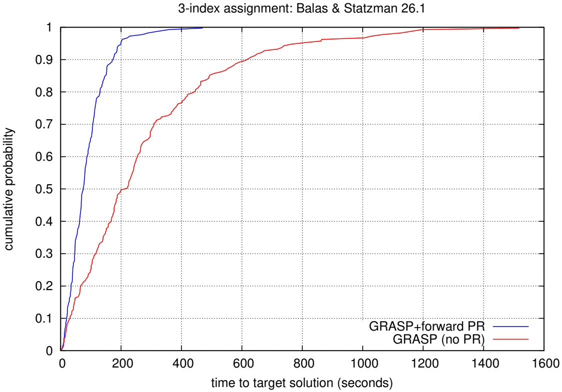
begin GRASP+PR;
1 $\mathcal{E} \leftarrow \varnothing$;
2 $f^* \leftarrow \infty$;
3 **while** stopping criterion not satisfied **do**
4     $S \leftarrow \mathrm{SEMI\text{-}GREEDY}$;
5   **if** S is not feasible **then**
6         $S \leftarrow \texttt{Repair}(S)$;
7   **end-if**;
8     $S \leftarrow \mathrm{LOCAL\text{-}SEARCH}(S)$;
9   **if** $|\mathcal{E}| > 0$ **then**
10   Select an elite solution S' at random from $\mathcal{E}$;
11         $S \leftarrow \mathrm{FORWARD\text{-}PR}(S, S')$;
12   **end-if**;
13   UPDATE-ELITE-SET($S, \mathcal{E}$);
14 **end-while**;
15 **return** $S^* = \mathrm{argmin}\{f(S) \,:\, S \in \mathcal{E}\}$;
end GRASP+PR.

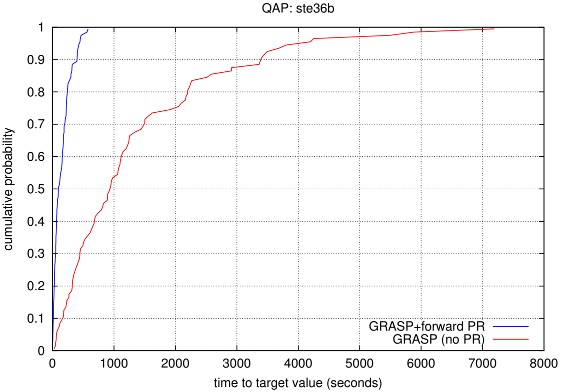
Hybridization of GRASP with path-relinking

- Enhancing GRASP with path-relinking almost always improves the performance of the heuristic.
- As an illustration, the next two slides show time-to-target plots (or runtime distributions) for GRASP with and without path-relinking for four different applications.
- These plots show the empirical cumulative probability distributions of the time-to-target random variable, i.e., the time needed to find a solution at least as good as a given target value.
- For all problems, the plots show that GRASP with path-relinking is able to find target solutions faster than the memoryless basic algorithm.

Hybridization of GRASP with path-relinking

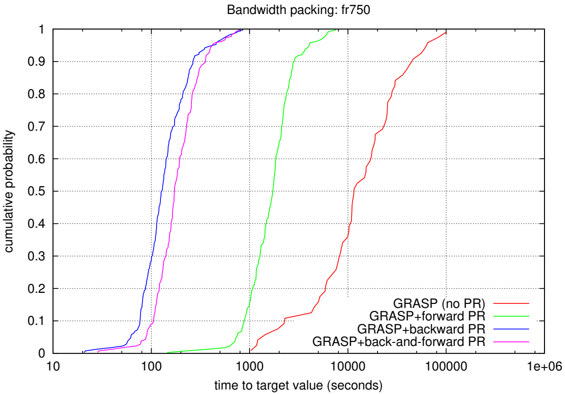
Time-to-target plots comparing running times of GRASP with and without path-relinking on distinct problems: three-index assignment and maximum satisfiability. Forward path-relinking was used in these two examples.

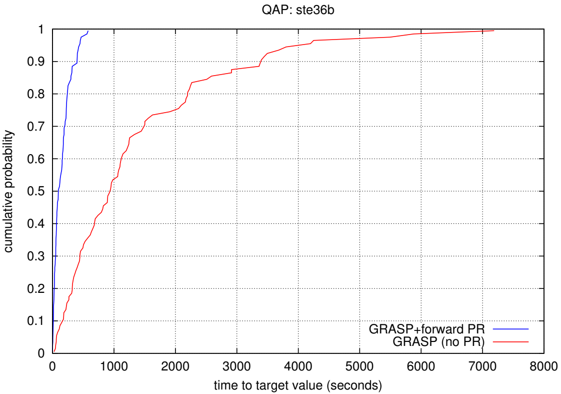




Hybridization of GRASP with path-relinking

Time-to-target plots comparing running times of GRASP with and without path-relinking on distinct problems: bandwidth packing and quadratic assignment. Forward path-relinking was used in these two examples. In addition, on the bandwidth packing example, plots for GRASP with backward and back-and-forward path-relinking are also shown.





Evolutionary path-relinking

- As aforementioned, GRASP with path-relinking heuristics maintain an elite set of high-quality solutions.
- In the variant of GRASP with path-relinking , locally optimal solutions produced by local search are relinked with elite set solutions.
- Path-relinking can also be applied to pairs of elite set solutions to search for new high-quality solutions and to improve the quality of the elite set.
- This procedure, called *evolutionary path-relinking* (EvPR), can be applied as a post-optimization phase of GRASP, after the main heuristic stops, or periodically, when the main heuristic is still running.
- The next two pseudo-codes correspond to the post-processing and periodic variants, respectively.

- The pseudo-code shows a template of a GRASP with evolutionary path-relinking heuristic where evolutionary path-relinking is applied at a post-processing step.
- The pseudo-code is identical to that of the GRASP with path-relinking (GRASP+PR), with an additional step in line 15 where EvPR is applied.

begin GRASP+EvPR;
1 $\mathcal{E} \leftarrow \varnothing$;
2 $f^* \leftarrow \infty$;
3 **while** stopping criterion not satisfied **do**
4     $S \leftarrow \text{SEMI-GREEDY}$;
5   **if** S is not feasible **then**
6         $S \leftarrow \texttt{Repair}(S)$;
7   **end-if**;
8     $S \leftarrow \text{LOCAL-SEARCH}(S)$;
9   **if** $|\mathcal{E}| > 0$ **then**
10   Select an elite solution S' at random from $\mathcal{E}$;
11         $S \leftarrow \text{FORWARD-PR}(S, S')$;
12   **end-if**;
13   UPDATE-ELITE-SET$(S, \mathcal{E})$;
14 **end-while**;
15 $\mathcal{E} \leftarrow \text{EvPR}(\mathcal{E})$;
16 **return** $S^* = \text{argmin}\{f(S) \ : \ S \in \mathcal{E}\}$;
end GRASP+EvPR.

- The pseudo-code shows a template of a GRASP with evolutionary path-relinking heuristic where evolutionary path-relinking is applied periodically during the search.
- The pseudo-code adds lines 3 and 15 to 19 to manage the periodic application of EvPR.
- Line 3 initializes `it2evPR`, a counter of iterations to EvPR, with `evPRfreq` being the number of GRASP iterations between consecutive calls to EvPR.
- If `evPRfreq` iterations have passed without the application of EvPR, then in line 16 it is applied and the counter `it2evPR` is reinitialized in line 17.
- Finally, in line 19, `it2evPR` is decreased by one iteration.

begin GRASP+itEvPR(`evPRfreq`);
1 $\mathcal{E} \leftarrow \varnothing$;
2 $f^* \leftarrow \infty$;
3 $\texttt{it2evPR} \leftarrow \texttt{evPRfreq}$;
4 **while** stopping criterion not satisfied **do**
5     $S \leftarrow \text{SEMI-GREEDY}$;
6   **if** S is not feasible **then**
7         $S \leftarrow \texttt{Repair}(S)$;
8   **end-if**;
9     $S \leftarrow \text{LOCAL-SEARCH}(S)$;
10   **if** $|\mathcal{E}| > 0$ **then**
11   Select an elite solution S' at random from $\mathcal{E}$;
12         $S \leftarrow \text{FORWARD-PR}(S, S')$;
13   **end-if**;
14   UPDATE-ELITE-SET$(S, \mathcal{E})$;
15   **if** $\texttt{it2evPR} < 1$ **then**
16         $\mathcal{E} \leftarrow \texttt{EvPR}(\mathcal{E})$;
17         $\texttt{it2evPR} \leftarrow \texttt{evPRfreq} + 1$;
18   **end-if**;
19     $\texttt{it2evPR} \leftarrow \texttt{it2evPR} - 1$;
20 **end-while**;
21 **return** $S^* = \operatorname{argmin}\{f(S) \,:\, S \in \mathcal{E}\}$;
end GRASP+itEvPR.

Evolutionary path-relinking

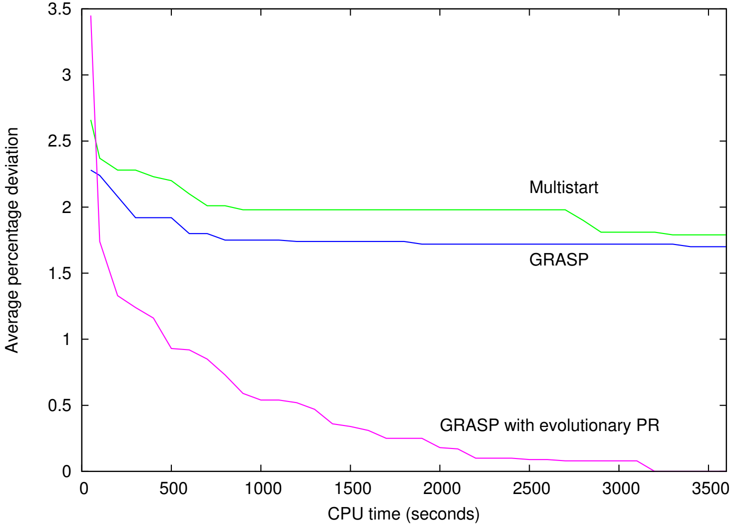
- Evolutionary path-relinking takes as input the elite set and returns either the same elite set or a renewed one with an improved average cost.
- This approach is outlined in the following pseudo-code:

begin `EvPR`($\mathcal{E}$);
  while there exists solutions $S^1, S^2 \in \mathcal{E}$ that have not yet been relinked, with $S^1 \neq S^2$ **do**
    $S \leftarrow \mathrm{FORWARD\text{-}PR}(S^1, S^2)$;
    $\mathrm{UPDATE\text{-}ELITE\text{-}SET}(S, \mathcal{E})$;
  end-while;
  return $\mathcal{E}$;
end `EvPR`.

Evolutionary path-relinking

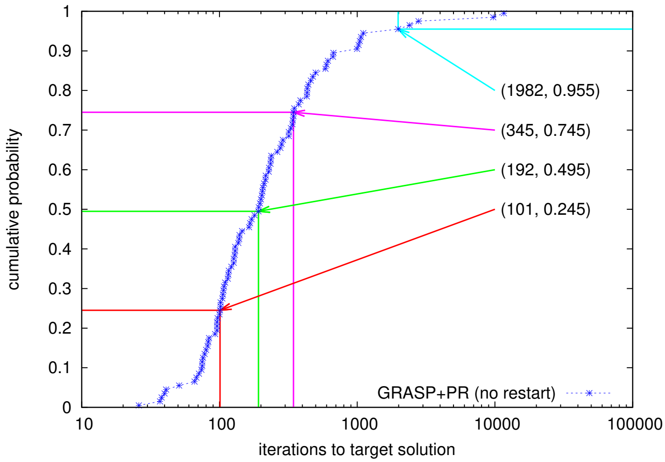
- While there exists a pair of solutions in the elite set for which path-relinking has not yet been applied, the two solutions are combined with path-relinking and the resulting solution is tested for membership in the elite set.
- If it is accepted, it then replaces the elite solution most similar to it among all solutions having worse cost.
- To explore more than one path connecting two solutions, evolutionary path-relinking can apply greedy randomized adaptive path-relinking a fixed number of times between each pair of elite solutions.
- This strategy outperformed several other heuristics using GRASP with path-relinking, simulated annealing, tabu search, and a multistart strategy for the max-min diversity problem.

The figure shows the evolution of the percent deviation from best known solution value found by the multistart strategy, pure GRASP, and GRASP with evolutionary path-relinking for a 500-element instance of a max-min diversity problem with a time limit of 60 minutes.



Restart strategies

The figure shows a typical iteration count distribution for a GRASP with path-relinking heuristic.

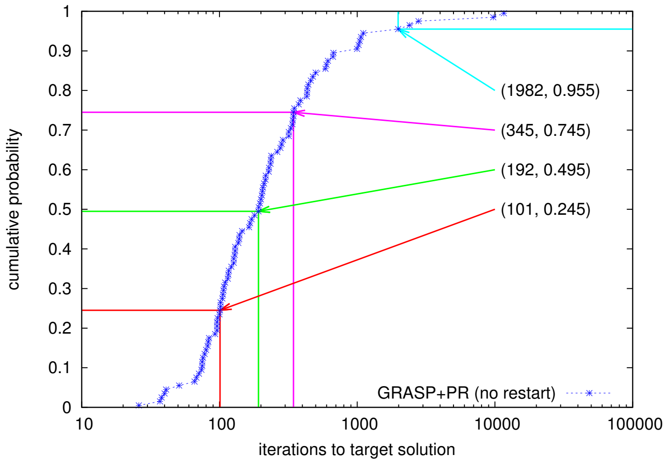


Observations:

- For most of the independent runs whose iteration counts make up the plot, the algorithm finds a target solution in relatively few iterations:
  - about 25% of the runs take at most 101 iterations;
  - about 50% take at most 192 iterations; and
  - about 75% take at most 345.

Restart strategies

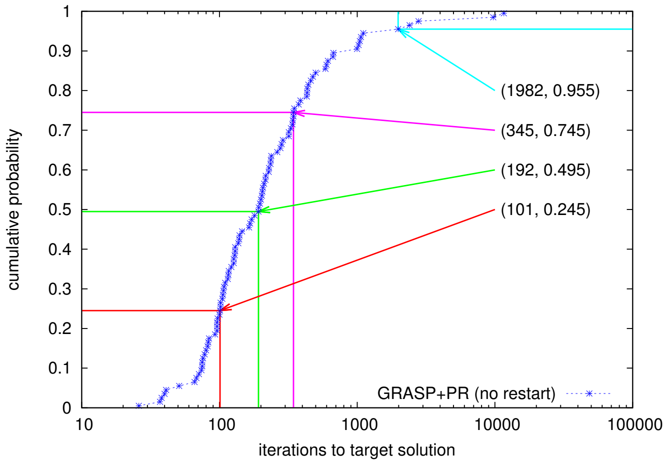
The figure shows a typical iteration count distribution for a GRASP with path-relinking heuristic.



Observations:

- However, some runs take much longer:
  - 10% take over 1000 iterations;
  - 5% over 2000; and
  - 2% over 9715 iterations.
- The longest run took 11607 iterations to find a solution at least as good as the target.

The figure shows a typical iteration count distribution for a GRASP with path-relinking heuristic.



Observations:

- These long tails contribute to a large average iteration count as well as to a high standard deviation.
- This section proposes strategies to reduce the tail of the distribution, consequently reducing the average iteration count and its standard deviation.

Restart strategies

- The distribution in previous figure shows that each run will take over 345 iterations with about 25% probability.
- Therefore, any time the algorithm is restarted, the probability that the new run will take over 345 iterations is also about 25%.
- By restarting the algorithm after 345 iterations, the new run will take more than 345 iterations with probability of also about 25%.
- Therefore, the probability that the algorithm will be still running after $345 + 345 = 690$ iterations is the probability that it takes more than 345 iterations multiplied by the probability that it takes more than 690 iterations given that it took more than 345 iterations, i.e., about $(1/4) \times (1/4) = (1/4)^2$.
- It follows by induction that the probability that the algorithm will still be running after k periods of 345 iterations is $1/(4^k)$.
- In this example, the probability that the algorithm will be running after 1725 iterations will be about 0.1%, i.e., much less than the 5% probability that the algorithm will take over 2000 iterations without restart.

Restart strategies

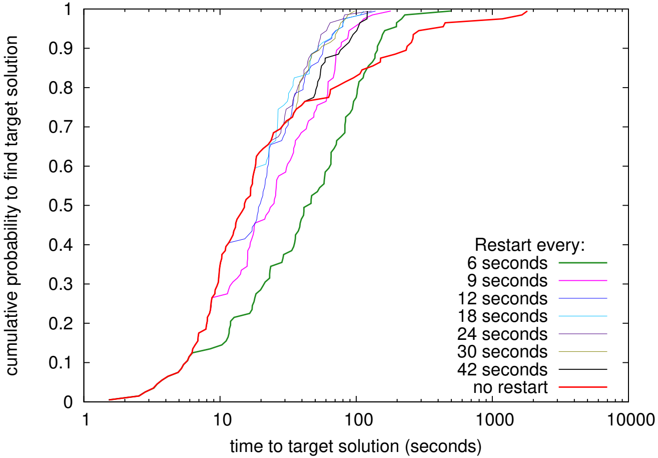
- A *restart strategy* is defined as an infinite sequence of time intervals $\tau_1, \tau_2, \tau_3, \ldots$ which define epochs $\tau_1, \tau_1 + \tau_2, \tau_1 + \tau_2 + \tau_3, \ldots$ when the algorithm is restarted from scratch.
- It can be shown that the optimal restart strategy uses $\tau_1 = \tau_2 = \cdots = \tau^*$, where τ^* is some (unknown) constant.
- Implementing the optimal strategy may be difficult in practice because it requires inputting the constant value τ^*.

Restart strategies

- Runtimes can vary greatly for different combinations of algorithm, instance, and solution quality sought.
- Since usually one has no prior information about the runtime distribution of the stochastic search algorithm for the optimization problem under consideration, one runs the risk of choosing a value of τ^* that is either too small or too large.
  - On the one hand, a value that is too small can cause the restart variant of the algorithm to take much longer to converge than a no-restart variant.
  - On the other hand, a value that is too large may never lead to a restart, causing the restart-variant of the algorithm to take as long to converge as the no-restart variant.

Restart strategies

The figure illustrates the restart strategies with time-to-target plots for the maximum cut instance *G12* on an 800-node graph with edge density of 0.63% with target solution value 554 for different values of τ.



- For each value of τ, 100 independent runs of a GRASP with path-relinking heuristic with restarts were performed.
- The variant with $\tau = \infty$ corresponds to the heuristic without restart.
- **Observation**: For some values of τ, the resulting heuristic outperformed its counterpart with no restart by a large margin.

Restart strategies

- In GRASP with path-relinking, the number of iterations between improvements of the incumbent (or best so far) solution tends to vary less than the runtimes for different combinations of instance and solution quality sought.
- If one takes this into account, a simple and effective restart strategy for GRASP with path-relinking is to keep track of the last iteration when the incumbent solution was improved and restart the GRASP with path-relinking heuristic if κ iterations have gone by without improvement.
- We shall call such a strategy restart(κ).
- A restart consists in saving the incumbent and emptying out the elite set.

The pseudo-code summarizes the steps of a GRASP with path-relinking heuristic using the restart(κ) strategy for a minimization problem.

- The algorithm keeps track of the current iteration (`CurrentIter`), as well as of the last iteration when an improving solution was found (`LastImprov`).
- If, in line 21, it is determined that more than κ iterations have gone by since the last improvement of the incumbent, then a restart is triggered.
- If restart is not triggered, then in line 25 the current solution S is tested for inclusion in the elite set and the set is updated if S is accepted.
- The best overall solution found S^* is returned in line 28 after the stopping criterion is satisfied.

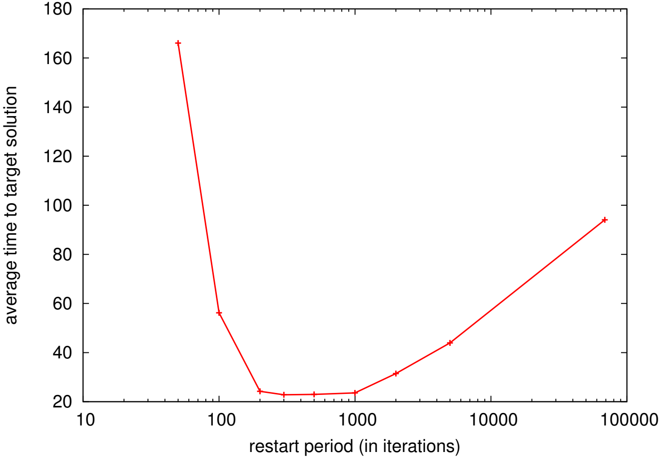
begin GRASP+PR+RESTARTS;
1 $\mathcal{E} \leftarrow \varnothing$;
2 $f^* \leftarrow \infty$;
3 `LastImprov` $\leftarrow 0$;
4 `CurrentIter` $\leftarrow 0$;
5 **while** stopping criterion not satisfied **do**
6   `CurrentIter` $\leftarrow$ `CurrentIter` $+ 1$;
7     $S \leftarrow \text{SEMI-GREEDY}$;
8   **if** S is not feasible **then**
9         $S \leftarrow$ `Repair`(S);
10   **end-if**;
11     $S \leftarrow \text{LOCAL-SEARCH}(S)$;
12   **if** $|\mathcal{E}| > 0$ **then**
13   Select an elite solution S' at random from $\mathcal{E}$;
14         $S \leftarrow \text{FORWARD-PR}(S, S')$;
15   **end-if;**
16   **if** $f(S) < f^*$ **then**
17         $S^* \leftarrow S$;
18         $f^* \leftarrow f(S)$;
19   `LastImprov` $\leftarrow$ `CurrentIter`;
20   **end-if;**
21   **if** `CurrentIter` $-$ `LastImprov` $> \kappa$ **then**
22         $\mathcal{E} \leftarrow \varnothing$;
23   `LastImprov` $\leftarrow$ `CurrentIter`;
24   **else**
25         $\text{UPDATE-ELITE-SET}(S, \mathcal{E})$;
26   **end-if**;
27 **end-while**;
28 **return** S^*;

Restart strategies

- As an illustration of the use of the restart(κ) strategy within a GRASP with path-relinking heuristic, consider the maximum cut instance *G12*.
- For the values $\kappa = 50$, 100, 200, 300, 500, 1000, 2000, and 5000, the heuristic was run independently 100 times, stopping when a cut of weight 554 or higher was found.
- A strategy without restarts was also implemented.
- The next two figures, as well as the following table, summarize these runs, showing the average time to target solution as a function of the value of κ and the time-to-target plots for different values of κ.

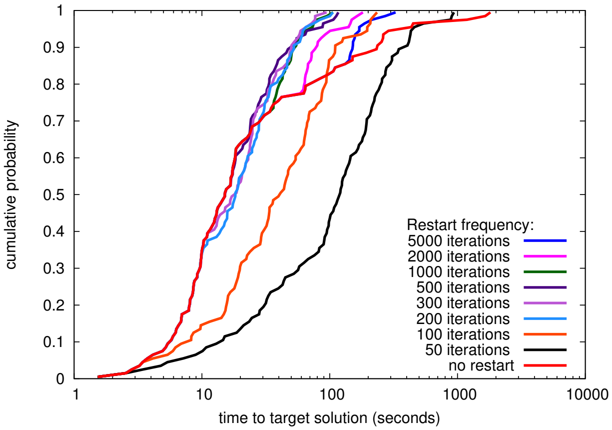
Restart strategies

Average time to target solution for maximum cut instance *G12* using different values of κ. All runs of all strategies have found a solution at least as good as the target value of 554.



Restart strategies

Time-to-target plots for maximum cut instance *G12* using different values of κ. The figure also shows the time-to-target plot for the strategy without restarts. All runs of all strategies found a solution at least as good as the target value of 554.



Restart strategies

Observations:

- The two previous figures illustrate well the effect on running time of selecting a value of κ that is either too small ($\kappa = 50, 100$) or too large ($\kappa = 2000, 5000$).
- They further show that there is a wide range of κ values ($\kappa =$ 200, 300, 500, 1000) that result in lower runtimes when compared to the strategy without restarts.

Restart strategies

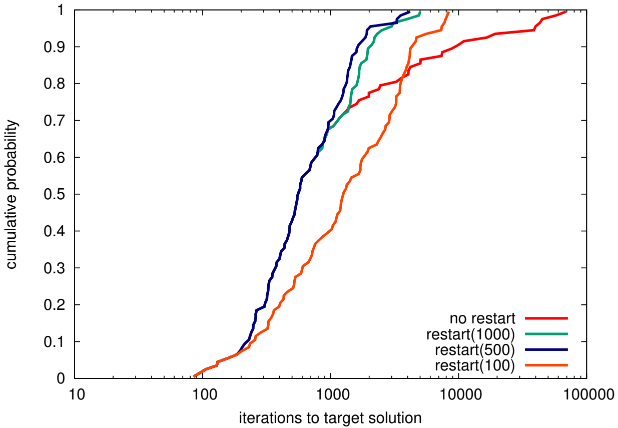
Summary of computational results on maximum cut instance *G12* with four strategies:

- For each strategy, the table shows the distribution of the number of iterations by quartile.
- For each quartile, the table gives the maximum number of iterations taken by all runs in that quartile, i.e., the slowest of the fastest 25% (1st), 50% (2nd), 75% (3rd), and 100% (4th) of the runs.
- The average number of iterations over the 100 runs and the standard deviation (st.dev.) are also given for each strategy.

	Iterations in quartile					
Strategy	1st	2nd	3rd	4th	Average	st.dev.
Without restarts	326	550	1596	68813	4525.1	11927.0
restart(1000)	326	550	1423	5014	953.2	942.1
restart(500)	326	550	1152	4178	835.0	746.1
restart(100)	509	1243	3247	8382	2055.0	2005.9

Restart strategies

The figure illustrates the iterations-to-target plots of the restart(100), restart(500), and restart(1000) strategies for the previous example, when compared with the strategy without restarts on the same maximum cut instance *G12*.



Onservation:

- As expected, each strategy restart(κ) behaves exactly like the strategy without restarts for the κ first iterations, for $\kappa = 100, 500, 1000$.
- After this point, each trajectory deviates from that of the strategy without restarts.
- Among these strategies, restart(500) is the one with the best performance.

Restart strategies

Concluding observations:

- The effect of the restart strategies can be mainly observed in the column corresponding to the fourth quartile of previous table.
- Entries in this quartile correspond to those in the heavy tails of the distributions.
- The restart strategies in general did not affect the other quartiles of the distributions, which is a desirable characteristic.
- Compared to the no-restart strategy, at least one restart strategy was always able to reduce the maximum number of iterations, the average number of iterations, and the standard deviation of the number of iterations.

Restart strategies

Concluding observations:

- Compared to the no-restart strategy, restart strategies restart(500) and restart(1000) were able to reduce the maximum number of iterations, as well as the average and the standard deviation.
- Strategy restart(100) did so, too, but not as much as restart(500) and restart(1000).
- Restart strategies restart(500) and restart(1000) were clearly the best strategies of those tested.

Concluding remarks

The material in this talk is taken from

- Chapter 9 – GRASP with path-relinking

of our book, *Optimization by GRASP: Greedy Randomized Adaptive Search Procedures* (Resende & Ribeiro, Springer. 2016).