Chapter 9
GRASP with path-relinking

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 chapter 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.

9.1 Memoryless GRASP

The basic GRASP heuristic, as presented in Chapter 5, searches the solution space by repeatedly applying independent searches in the solution space graph $\mathscr{G} = (F, M)$, 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 (introduced in Sections 7.1 and 7.6, respectively) 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.

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.

DOI 10.1007/978-1-4939-6530-4_9

In this chapter, 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.

9.2 Elite sets

An elite set $\mathscr{E}$ of solutions is a set formed by at most a fixed number $n_{\mathscr{E}}$ 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.

A basic scheme to maintain an elite set $\mathscr{E}$ for a minimization problem is outlined in the algorithm of Figure 9.1. The algorithm is given a candidate solution S and determines if S should be added to $\mathscr{E}$ and, if so, which solution, if any, should be removed from $\mathscr{E}$.

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

Fig. 9.1 Pseudo-code of a template for the maintenance of the elite set $\mathscr{E}$ of at most $n_{\mathscr{E}}$ elements in the context of a minimization problem.

If line 1 determines that the elite set $\mathscr{E}$ is not full, i.e., if $|\mathscr{E}| < n_{\mathscr{E}}$, then a candidate solution S is always added to $\mathscr{E}$ if it is different from any solution currently in the set. This case is treated in lines 2 to 7 of the pseudo-code. In line 3, S is added to $\mathscr{E}$ if the elite set is empty. Let the *symmetric difference* $\Delta(S,S')$ be formed by the ground set elements that belong to either S or S'. In line 5, the minimum cardinality

δ among the symmetric differences between S and the elements of $\mathscr{E}$ is computed. If S is different from all elite solutions, then it is added to $\mathscr{E}$ in line 6.

Otherwise, if the elite set is full (i.e., if $|\mathscr{E}| = n_{\mathscr{E}}$), then any time a solution is added to the set, another solution must be removed from it, thus maintaining the size of $\mathscr{E}$ equal to $n_{\mathscr{E}}$. Our goal is to first improve the average quality of the elite set, and then maximize the diversity of its elements, which amounts to maximizing the cardinalities of the symmetric differences between all pairs of solutions in the set. This case is treated in lines 9 to 14. In line 9, the cost f^+ of the worst-valued elite set solution is computed, while in line 10 the minimum cardinality δ among the symmetric differences between S and any element of $\mathscr{E}$ is determined. S is added to $\mathscr{E}$ if it is better than the worst solution in the elite set and if it is different from all elite solutions, i.e., if $f(S) < f^+$ and $\delta > 0$ in line 11. This is accomplished in lines 12 and 13. Line 12 determines, among all elite set solutions valued no better than S, one which is most similar to S, i.e., one which minimizes the cardinality of its symmetric difference with respect to S. This solution, S^-, is removed from $\mathscr{E}$ in line 13. The new elite solution S is inserted in the pool as a replacement for S^- at the same line. The updated elite set is returned in line 16.

The algorithm in Figure 9.1 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.

9.3 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. In this section, we show how to hybridize path-relinking with GRASP.

To implement GRASP with path-relinking, we make use of an elite set $\mathscr{E}$, such as the one introduced in Section 9.2, 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 $n_{\mathscr{E}}$ 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 algorithm UPDATE-ELITE-SET of Figure 9.1.

The pseudo-code of Figure 9.2 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 described in Section 8.1.2. The output of the path-relinking operator is a candidate for inclusion in the elite set.

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

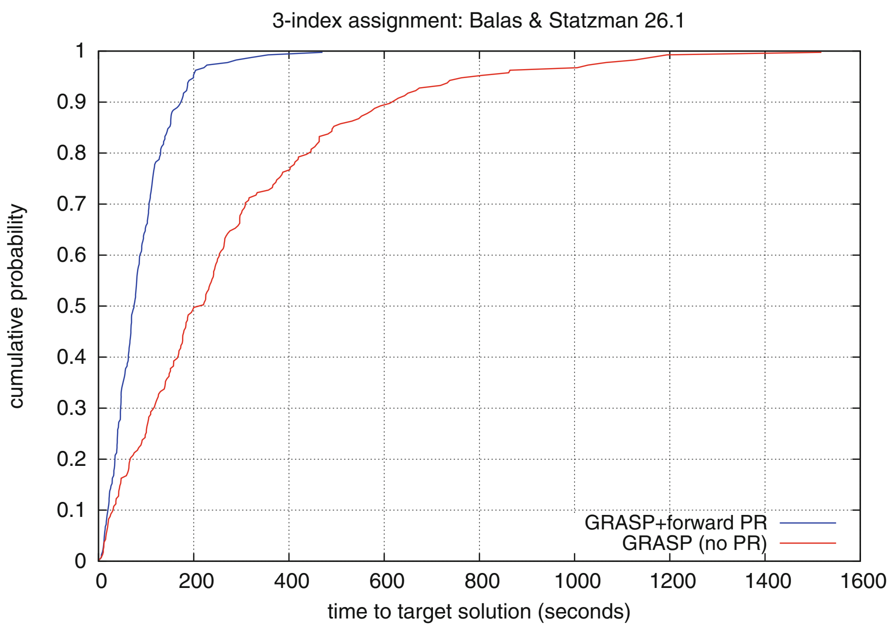
Fig. 9.2 Pseudo-code of a template of a basic GRASP with path-relinking heuristic for minimization.

Line 1 of the pseudo-code initializes the elite set $\mathscr{E}$ as empty. The loop from line 2 to line 13 makes up the steps of GRASP with path-relinking. Lines 3 to 7 correspond to the semi-greedy construction, repair (in case of infeasibility), and local search phases of a basic GRASP heuristic. Forward path-relinking is performed in lines 9 and 10 in case the elite set is not empty: in line 9, an elite set solution S' is selected at random from $\mathscr{E}$ while, in line 10, S' is relinked with the locally optimal solution S produced in line 7. The resulting solution, S, is tested for inclusion in the elite set in line 12, which updates $\mathscr{E}$ by applying algorithm UPDATE-ELITE-SET of Figure 9.1. The algorithm returns the best-valued elite solution in line 14, after a stopping criterion is met.

Enhancing GRASP with path-relinking almost always improves the performance of the heuristic. As an illustration, Figures 9.3 and 9.4 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.

9.4 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 introduced in Section 9.3, 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.



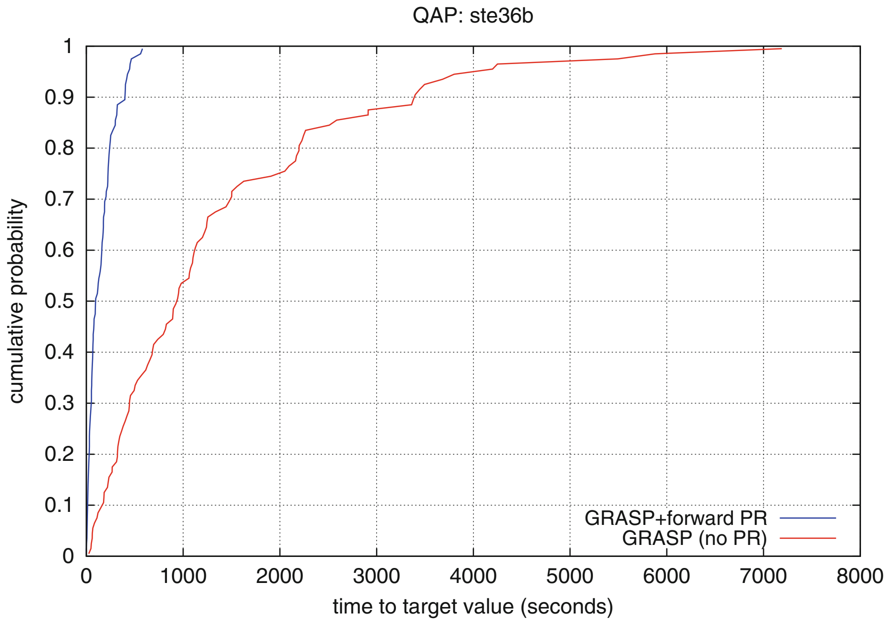
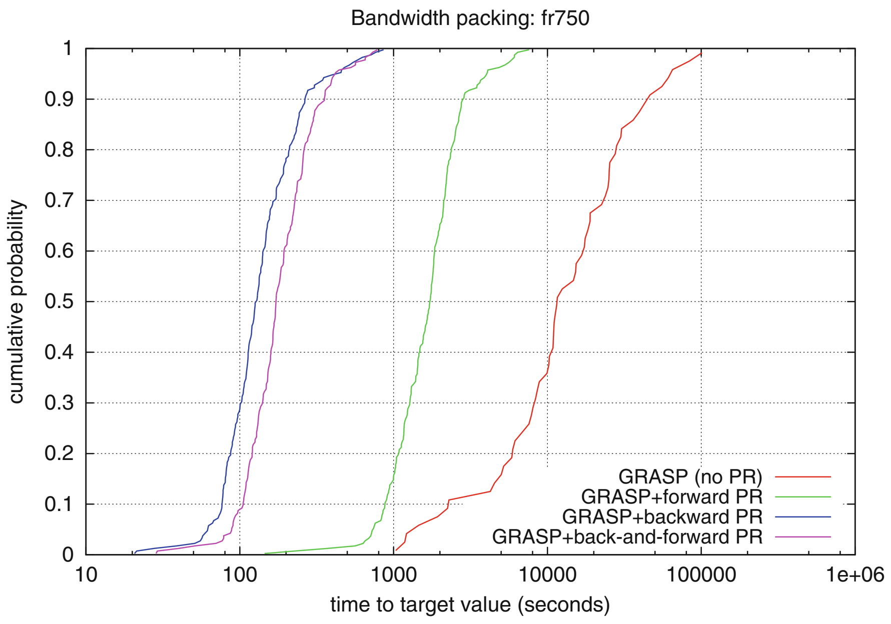


Fig. 9.3 Time-to-target plots comparing running times of GRASP with and without path-relinking on distinct problems: three-index assignment and maximum satisfiability. Forward path-relinking was used in these two examples.



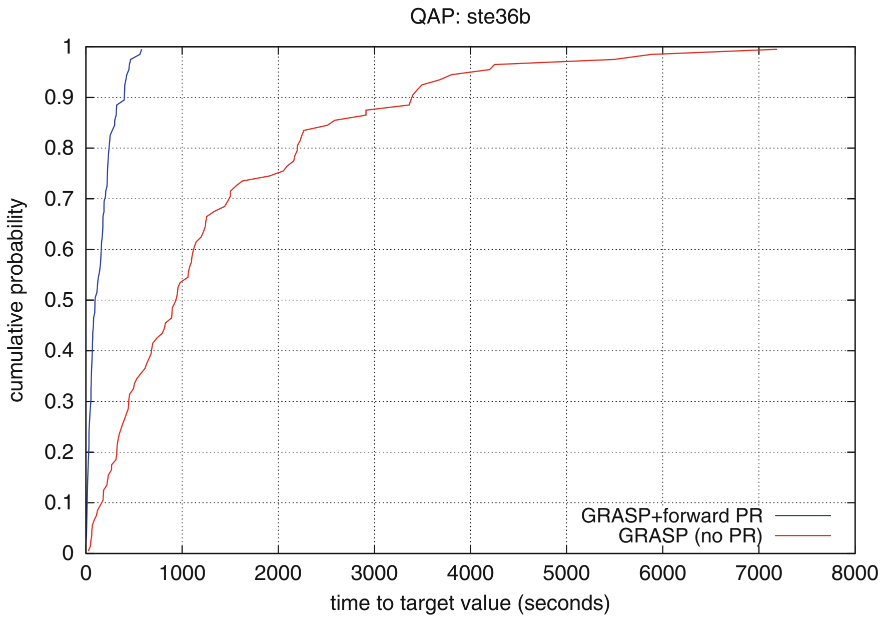


Fig. 9.4 Time-to-target plots comparing running times of GRASP with and without path-relinking on distinct problems: bandwidth packing and quadratic assignment. Forward path-relinking was used in these two examples. In addition, on the bandwidth packing example, plots for GRASP with backward and back-and-forward path-relinking are also shown.

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.

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

Fig. 9.5 Pseudo-code of a template of a GRASP with evolutionary path-relinking heuristic where evolutionary path-relinking is applied at a post-processing step.

The pseudo-codes in Figures 9.5 and 9.6 correspond to the post-processing and periodic variants, respectively. The pseudo-code in Figure 9.5 is identical to that of the GRASP with path-relinking of Figure 9.2, with an additional step in line 15 where EvPR is applied.

The pseudo-code of Figure 9.6 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.

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 pseudo-code of Figure 9.7. 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

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

Fig. 9.6 Pseudo-code of a template of a GRASP with evolutionary path-relinking heuristic where evolutionary path-relinking is applied periodically during the search.

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

Fig. 9.7 Pseudo-code of a template of the evolutionary path-relinking strategy.

max-min diversity problem. Figure 9.8 shows the evolution of the best solution found by the multistart strategy, pure GRASP, and GRASP with evolutionary path-relinking for a 500-element max-min diversity instance.

9.5 Restart strategies

Figure 9.9 shows a typical iteration count distribution for a GRASP with path-relinking heuristic. Observe in this example that for most of the independent runs whose iteration counts make up the plot, the algorithm finds a target solution in

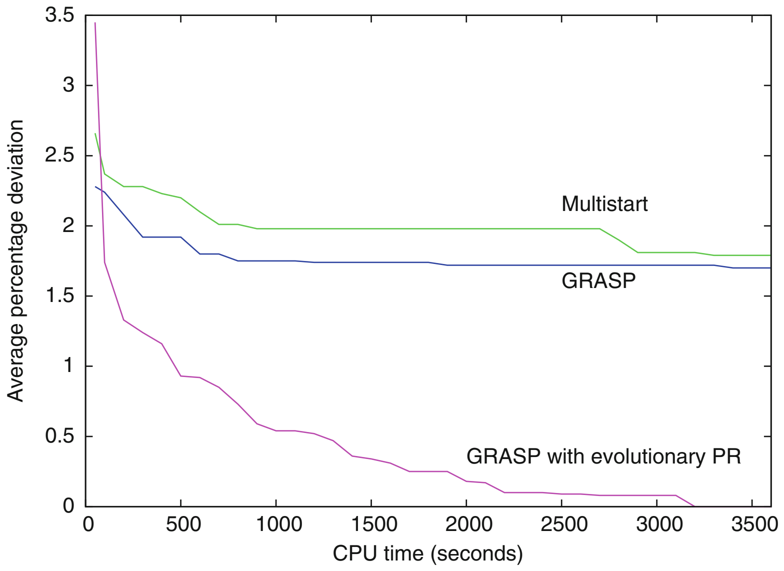


Fig. 9.8 Percent deviation from best known solution value for GRASP with evolutionary path-relinking, pure GRASP, and a multistart algorithm for a 500-element instance of a max-min diversity problem with a time limit of 60 minutes.

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. 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. 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.

Consider again the distribution in Figure 9.9. The distribution 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.

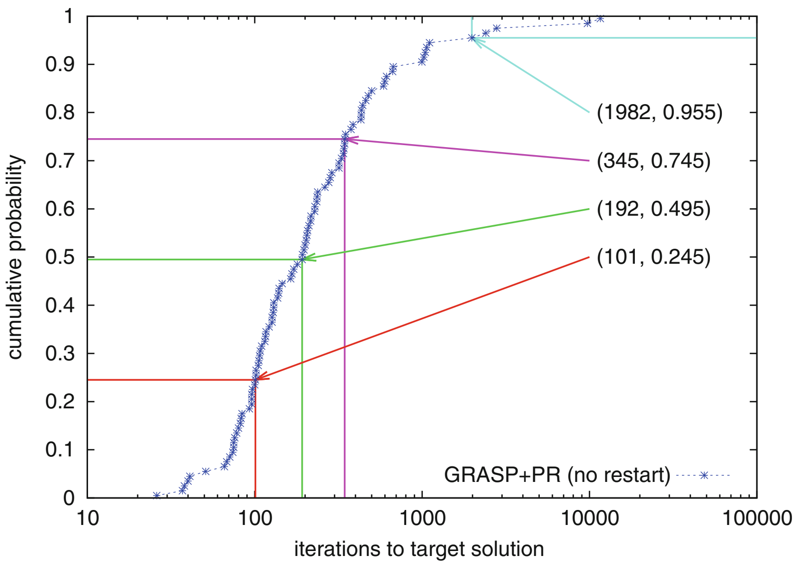


Fig. 9.9 Typical iteration count distribution of GRASP with path-relinking.

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^*. 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. Figure 9.10 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 τ = 6, 9, 12, 18, 24, 30, and 42 seconds. 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. The figure shows that, for some values of τ, the resulting heuristic outperformed its counterpart with no restart by a large margin.

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

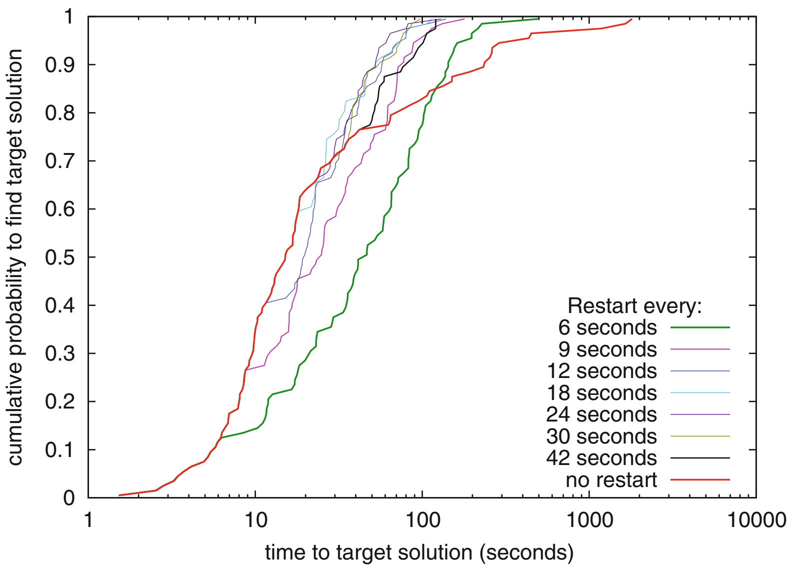


Fig. 9.10 Time-to-target plot for target solution value of 554 for maximum cut instance *G12* using different values of τ.

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 shown in Figure 9.11 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 an improving solution is detected in line 16, then this solution and its cost are saved in lines 17 and 18, respectively, and the iteration of last improvement is set to the current iteration in line 19. 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, emptying out the elite set in line 22 and resetting the iteration of last improvement to the current iteration in line 23. 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.

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. Figures 9.12 and 9.13, as well as Table 9.1, summarize these runs, showing the average time to target solution as a function of the value

```
begin GRASP+PR+RESTARTS;
1  $\mathcal{E} \leftarrow \emptyset$;
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      $S \leftarrow$ SEMI-GREEDY;
8      if $S$ is not feasible then
9          $S \leftarrow$ Repair$(S)$;
10     end-if;
11     $S \leftarrow$ LOCAL-SEARCH$(S)$;
12     if $|\mathcal{E}| > 0$ then
13         Select an elite solution $S'$ at random from $\mathcal{E}$;
14         $S \leftarrow$ 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         $\mathcal{E} \leftarrow \emptyset$;
23         LastImprov $\leftarrow$ CurrentIter;
24     else
25         UPDATE-ELITE-SET$(S, \mathcal{E})$;
26     end-if;
27 end-while;
28 return $S^*$;
end GRASP+PR+RESTARTS.
```

Fig. 9.11 Pseudo-code of a template of a GRASP with path-relinking heuristic with restarts for a minimization problem.

of κ and the time-to-target plots for different values of κ. These 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 (κ = 200, 300, 500, 1000) that result in lower runtimes when compared to the strategy without restarts.

Figure 9.14 further illustrates the behavior 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*. However, in this figure, for each strategy, we plot the number of iterations to the target solution value. It is interesting to note that, 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.

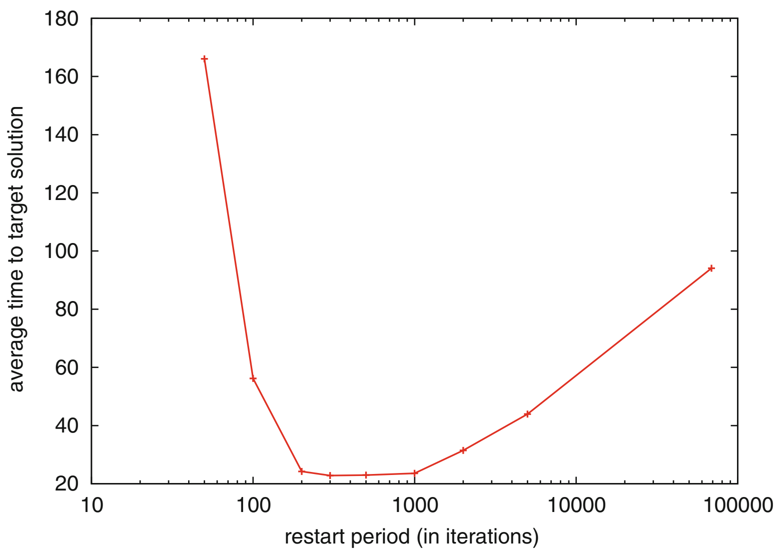


Fig. 9.12 Average time-to-target solution for maximum cut instance *G12* using different values of κ. All runs of all strategies have found a solution at least as good as the target value of 554.

We conclude this chapter with some observations about these experiments. The effect of the restart strategies can be mainly observed in the column corresponding to the fourth quartile of Table 9.1. 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. 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.

Table 9.1 Summary of computational results on maximum cut instance *G12* with four strategies. For each strategy, 100 independent runs were executed, each stopped when a solution as good as the target solution value 554 was found. 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

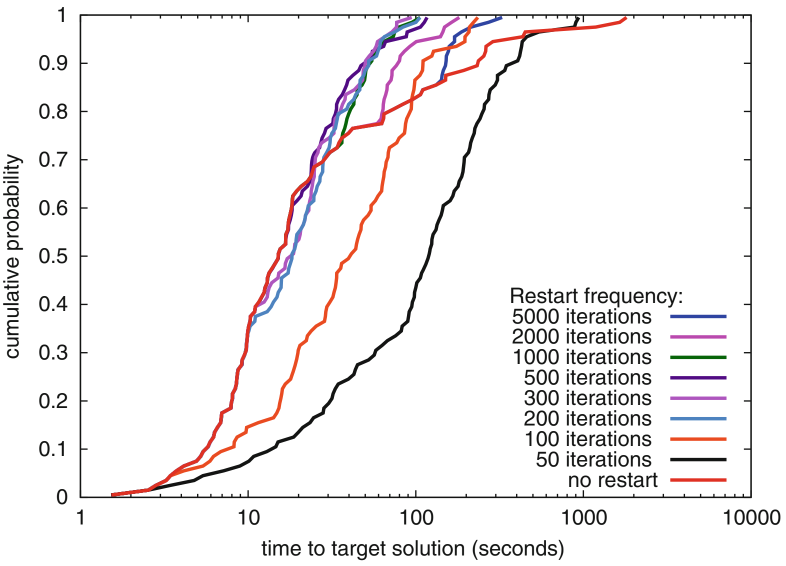


Fig. 9.13 Time-to-target plots for maximum cut instance *G12* using different values of κ. The figure also shows the time-to-target plot for the strategy without restarts. All runs of all strategies found a solution at least as good as the target value of 554.

9.6 Bibliographical notes

GRASP with path-relinking as proposed in Section 9.3 was first introduced by Laguna and Martí (1999), where a forward path-relinking operator from the solution found by local search to a randomly selected elite solution was applied. This was followed by a number of applications of GRASP with path-relinking, e.g., to maximum cut (Festa et al., 2002), 2-path network design (Ribeiro and Rosseti, 2002), Steiner problem in graphs (Ribeiro et al., 2002), job-shop scheduling (Aiex et al., 2003), private virtual circuit routing (Resende and Ribeiro, 2003a), p-median (Resende and Werneck, 2004), quadratic assignment (Oliveira et al., 2004), set packing (Delorme et al., 2004), three-index assignment (Aiex et al., 2005), p-hub median (Pérez et al., 2005), uncapacitated facility location (Resende and Werneck, 2006), project scheduling (Alvarez-Valdes et al., 2008a), maximum weighted satisfiability (Festa et al., 2006), maximum diversity (Silva et al., 2007), network migration scheduling (Andrade and Resende, 2007a), capacitated arc routing (Labadi et al., 2008; Usberti et al., 2013), disassembly sequencing (Adenso-Díaz et al., 2008), flowshop scheduling (Ronconi and Henriques, 2009), multi-plant capacitated lot sizing (Nascimento et al., 2010), workover rig scheduling (Pacheco et al., 2010), max-min diversity (Resende et al., 2010a), biobjective orienteering (Martí et al., 2015), biobjective path dissimilarity (Martí et al., 2015), generalized quadratic assignment (Mateus et al., 2011), antibandwidth (Duarte et al., 2011), capacitated clustering (Deng and Bard, 2011), linear ordering (Chaovalitwongse et al., 2011), data

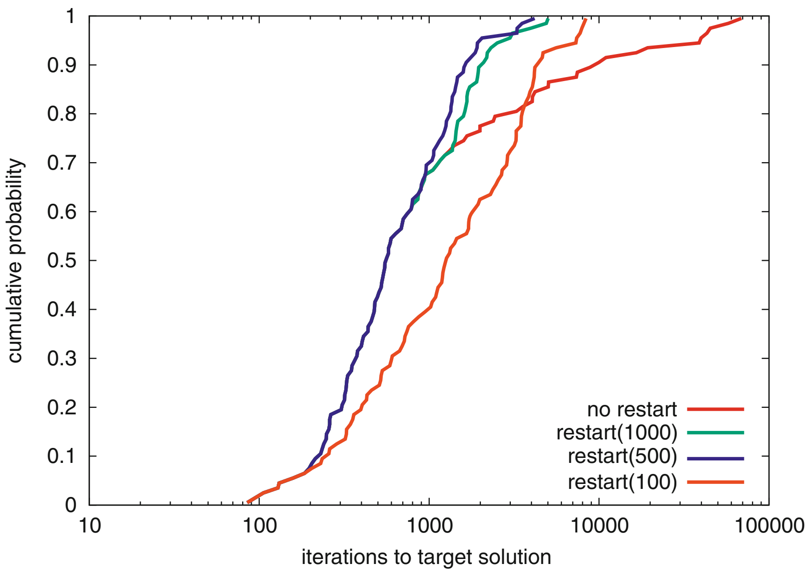


Fig. 9.14 Comparison of the iterations-to-target plots for maximum cut instance *G12* using strategies restart(100), restart(500), and restart(1000). The figure also shows the iterations-to-target plot for the strategy without restarts. All runs of all strategies found a solution at least as good as the target value of 554.

clustering (Frinhani et al., 2011), two-echelon location routing (Nguyen et al., 2012), image registration (Santamaría et al., 2012), drawing proportional symbols in maps (Cano et al., 2013), family traveling salesperson (Morán-Mirabal et al., 2014), handover minimization in mobility networks (Morán-Mirabal et al., 2013b), facility layout (Silva et al., 2013b), survivable network design (Pedrola et al., 2013), equitable dispersion (Martí and Sandoya, 2013), 2D and 3D bin packing (Alvarez-Valdes et al., 2013), microarray data analysis (Cordone and Lulli, 2013), community detection (Nascimento and Pitsoulis, 2013), set k-covering (Pessoa et al., 2013), network load balancing (Santos et al., 2013), power optimization in ad hoc networks (Moraes and Ribeiro, 2013), capacitated vehicle routing (Sörensen and Schittekat, 2013), and symmetric Euclidean clustered traveling salesman (Mestria et al., 2013).

Surveys on GRASP with path-relinking can be found in Resende and Ribeiro (2005a), Aiex and Resende (2005), Resende (2008), Resende et al. (2010b), Resende and Ribeiro (2010), Ribeiro and Resende (2012), and Festa and Resende (2013). A special issue of *Computers & Operations Research* (Martí et al., 2013b) was dedicated to GRASP with path-relinking.

Section 9.4 discussed evolutionary path-relinking that was originally proposed by Resende and Werneck (2004), where it was used as a post-processing phase for a GRASP with path-relinking for the p-median problem. Andrade and Resende (2007a) were the first to apply evolutionary path-relinking periodically during the search. The term evolutionary path-relinking was introduced by Andrade and Resende (2007b). This was followed by a number of applications of GRASP with evolutionary path-relinking, e.g., to uncapacitated facility location (Resende and Werneck, 2006), max-min diversity (Resende et al., 2010a), image registration (Santamaría et al., 2010; Santamaría et al., 2012), power transmission network expansion planning (Rahmani et al., 2010), vehicle routing with trailers (Villegas, 2010), antibandwidth minimization (Duarte et al., 2011), truck and trailer routing (Villegas et al., 2011), parallel machine scheduling (Rodriguez et al., 2012), linear ordering (Duarte et al., 2012), family traveling salesperson (Morán-Mirabal et al., 2014), handover minimization in mobility networks (Morán-Mirabal et al., 2013b), set covering (Morán-Mirabal et al., 2013a), maximum cut (Morán-Mirabal et al., 2013a), node capacitated graph partitioning (Morán-Mirabal et al., 2013a), capacitated arc routing (Usberti et al., 2013), and 2D and 3D bin packing (Alvarez-Valdes et al., 2013),

Figures 9.3 and 9.4 show time-to-target plots comparing pure GRASP and GRASP with path-relinking implementations on instances of the three-index assignment problem (Aiex et al., 2005), maximum satisfiability (Festa et al., 2006), bandwidth packing (Resende and Ribeiro, 2003a), and the quadratic assignment problem (Oliveira et al., 2004).

Figure 9.8 shows results from Resende et al. (2010a), where a GRASP and GRASP with evolutionary path-relinking for max-min diversity were proposed. The simulated annealing and multistart algorithms were the ones described in Kincaid (1992) and Ghosh (1996), respectively.

The restart(κ) strategy for GRASP with path-relinking discussed in Section 9.5 was proposed by Resende and Ribeiro (2011). Besides the experiments presented in this chapter for the maximum cut instance *G12*, that paper also considered five other instances of maximum cut, maximum weighted satisfiability, and bandwidth packing. Strategies for speeding up stochastic local search algorithms using restarts were first proposed by Luby et al. (1993), where they proved the result for an optimal restart strategy. Restart strategies in metaheuristics have been addressed in D'Apuzzo et al. (2006), Kautz et al. (2002), Nowicki and Smutnicki (2005), Palubeckis (2004), and Sergienko et al. (2004). Further work on restart strategies can be found in Shylo et al. (2011a) and Shylo et al. (2011b).