Greedy Randomized Adaptive Search Procedure (GRASP)

Vinayak Sareen, Joseph McGeever, Sergiu Harjau, Alicja Osicka

1 GRASP

The GRASP metaheuristic was developed in late 1980's (Feo and Resende, 1989). Since then it was under continuous development, and GRASP was applied to many problem areas such as Travelling Salesman Problem. GRASP is a general, metaheuristic, high level procedure used to find good (ideally optimal) solution to computationally difficult combinatorial optimization problems. (Darby-Dowman and Consoli, 2006) It is an iterative (or multi-start) process, with each iteration consisting of construction phase and local search phase.

Algorithm 1 GRASP metaheuristic

```
                                    ▷ Input: $IntegerMaxIterations$ and $Seed$
▷ Output: $BestSolution$
  for $k \leftarrow 1$ to $MaxIterations$ do
      $Solution \leftarrow GreedyRandomizedConstruction(Seed)$
      $Solution \leftarrow LocalSearch(Solution)$
      $UpdateSolution(Solution, BestSolution)$
  end for
  return $BestSolution$
```

1.1 Construction Phase

In the construction phase an easy and convenient solution is made by choosing the next element randomly in the Restricted Candidate List (RCL). RCL contains r best elements chosen by a greedy function. It allows to obtain a different solution at each iteration.

Algorithm 2 GreedyRandomizedConstruction

```
                                                  ▷ Input: $Seed$
▷ Output: $Solution$
  $Solution \leftarrow \emptyset$
  Evaluate the incremental costs of the candidate elements
  while $Solution$ is not complete $Solution$ do
      Build the restricted candidate list (RCL)
      Select an element $s$ from the RCL at random
      $Solution \leftarrow Solution \cup s$
      Revaluate the incremental costs
  end while
  return $Solution$
```

1.2 Local Search Phase

In the local search phase, the neighbourhood is investigated until a local minimum is found. Effectiveness of local search procedure depends on several aspects, such as the neighbourhood structure (density, regularity), neighbourhood search technique, fast evaluation of the cost function of the neighbours, and the starting solution itself. (Glover and Kochenberger, 2003)

Algorithm 3 Local Search

```
                                                  ▷ Input: $Solution$
▷ Output: $Solution$
  while $Solution$ is not locally optimal do
      Find $s' \in$ neighbourhood $N(Solution)$ with $f(s') < f(Solution)$
      $Solution \leftarrow s'$
  end while
  return $Solution$
```

1.3 Solution

The best overall solution is kept as a result. Solutions provided by the greedy randomized construction are not necessarily optimal, even with respect to simple neighbourhoods. (Glover and Kochenberger, 2003) Local search phase usually improves constructed solution. Local search algorithm

works in iterative fashion, by successively replacing current solution by a better one in its neighbourhood.
GRASP has two main parameters:

- Parameter for stopping iterations.
- Parameter related to the quality of the elements in RCL.

The parameter for the maximum number of iterations needs to be present, to avoid never ending loops, and keeping the cost down. Probability of finding a new better solution then the current best decreases with every iteration, although the quality of the best solution may only improve towards the end. The computation time is pretty much the same for every iteration – computation time is predictable and increases linearly with the number of iterations.

1.4 Time complexity analysis

1. Loop n times: O(n^2)
2. Construction phase O(n^2)
3. Local search phase: O(n^2)
4. Update solution: O(1)
5. Apply m constraints: O(m)
6. Return solution O(1)

Time complexity of GRASP algorithm is O(n^2)m, where n is the number of iterations and m is the number of constraints.