What is GRASP?

Greedy Randomized Adaptive Search Procedures (GRASP) is an SLS method that tries to construct a large variety of good initial solutions for a local search algorithm.

- predecessors: semi-greedy heuristics
- tries to combine the advantages of random and greedy solution construction

Greedy construction heuristics

- iteratively construct solutions by choosing at each construction step one solution component
  - solution components are rated according to a greedy function
  - the best ranked solutions component is added to the current partial solution
- examples: Kruskal's algorithms for minimum spanning trees, greedy heuristic for the TSP, ...
- advantage: generate good quality solutions; local search runs fast and finds typically better solutions than from random initial solutions
- disadvantage: do not generate many different solutions; difficulty of iterating

Random vs. greedy construction

- random construction
  - high solution quality variance
  - low solution quality
- greedy construction
  - good quality
  - low (no) variance
- goal: exploit advantages of both

Semi-greedy heuristics

- add at each step not necessarily the highest rated solution
- repeat until a full solution is constructed:
  - rate solution components according to a greedy function
  - put high rated solution components into a *restricted candidate list* (RCL)
  - choose one element of the RCL randomly and add it to the partial solution
  - adaptive element: greedy function depends on the partial solution constructed so far

Hart, Shogan, 1987

Generation of the RCL

- mechanisms for generating RCL
  - cardinality based: include the k best rated solution components into RCL
  - value based: include all solution components with greedy values better than a given threshold
- $\min\max\alpha$ based RCL
  - let f_{min} (f_{max}) be greedy values of best (worst) ranked solution component
  - include solution components e with greedy values

$$f(e) \leq f_{min} + \alpha \cdot (f_{max} - f_{min})$$

  $\alpha \in [0, 1]$ is a parameter
  - $\alpha = 0$ corresponds to a greedy construction heuristic
  - $\alpha = 1$ corresponds to a random solution construction

GRASP

- GRASP tries to capture advantages of random and greedy solution construction
- iterate through
  - randomized solution construction exploiting a greedy probabilistic bias to construct feasible solutions
  - apply local search to improve over the constructed solution
- keep track of the best solution found so far and return it at the end

GRASP — local search

- local search from random solutions
  - high variance
  - best solution quality often better than greedy (if not too large instances)
  - average solution quality worse than greedy
  - local search requires many improvement steps
- local search from greedy solutions
  - average solution quality better than random
  - local search typically requires only a few improvement steps
  - low (no) variance

GRASP — Procedural view

procedure *GRASP*
  Initialize Parameter
  while (termination condition not met) **do**
    $s = \textsf{ConstructGreedyRandomizedSolution}()$
    $s' = \textsf{LocalSearch}(s)$
    if $f(s') < f(s_{best})$
      $s_{best} = s'$
  end
  return s_{best}
end *GRASP*

GRASP Example: SAT

- solution components are value assignment to variables
- greedy-Function
  - number of still unsatisfied clauses that would become satisfied by a value assignment
  - Φ_i^+: set of additionally satisfied clauses if $x_i = \texttt{true}$
  - Φ_i^-: set of additionally satisfied clauses if $x_i = \texttt{false}$
- $\min\max\alpha$ based RCL
  - Let $\Phi^* = \max\{|\Phi_i^+|, |\Phi_i^-|\}$ over all free variables x_i
  - $x_i = \texttt{true} \in$ RCL if $|\Phi_i^+| \geq \alpha \cdot \Phi^*$
    $x_i = \texttt{false} \in$ RCL if $|\Phi_i^-| \geq \alpha \cdot \Phi^*$

GRASP Example: SAT

- variable selection
  - if an unsatisfied clause contains only one single still uninstantiated variable, try to satisfy this clause
  - otherwise choose randomly an element from the RCL
- local search
  - best-improvement 1-opt local search (GSAT architecture)
- performance
  - at the time the research was done reasonably good performance
  - however, nowadays by far outperformed by more recent local search algorithms for SAT
  - the same is true for weighted MAX-SAT

GRASP extensions

- convergence of GRASP (not guaranteed if $\alpha \neq 1$)
- introduction of a bias when choosing elements from the RCL
  - different possibilities of using, e.g. ranks (e.g. bias$(r) = 1/r$)
  - choose a solution component with a probability proportional to bias
- reactive GRASP (tuning of α)
- addition of a type of long term memory to bias search
  - path relinking
  - use of previous elite solutions to guide construction
- parallelization of GRASP

GRASP — concluding remarks

- straightforward extension of construction heuristics
- easy to implement
- few parameters
- many different applications available
- several extensions exist
- can be used to generate initial population in population-based methods
- however, as a stand-alone procedure often not state-of-the-art results