Metaheuristic methods

GRASP: Greedy Randomized Adaptive Search Procedure

Juan Antonio Díaz García

GRASP

1. Semi-Greedy algorithms
2. Random multistart procedures
3. Semi-greedy multistart procedures
4. GRASP Metaheuristic

GRASP

Greedy
Randomized
Adaptive
Search
Procedure

Semi-Greedy algorithms

The idea of a *semi-greedy* algorithm is to add randomization to a greedy or to an adaptive greedy algorithm. A greedy heuristic can also be randomized by selecting at random one element of a *Restricted Candidate List*, denoted by *RCL*, which contains the best candidate elements (with respect to their greedy function value) from the set of feasible elements $\mathscr{F}$. This can be done with the following two schemes:

- *Cardinality-based*: Select one of the $k = \min\{K, |\mathscr{F}|\}$ elements with the lowest values of the greedy function, where K is the maximum size of the *RCL*.
- *Quality-based*: Let $g_{min} = \min\{g(i) : i \in \mathscr{F}\}$ and $g_{max} = \max\{g(i) : i \in \mathscr{F}\}$. Let also $0 < \alpha < 1$ be a parameter that controls the greediness or randomness of the semi-greedy heuristic. The *RCL* can be constructed in such a way that it contains the elements whose greedy function value satisfies $g_{min} \leq g(i) \leq g_{min} + \alpha\,(g_{max} - g_{min})$.

Semi-greedy pseudocode

Semi-greedy pseudocode (minimization case)

$S \leftarrow \emptyset$
$f(S) \leftarrow 0$
$\mathscr{F} \leftarrow \{i \in E : S \cup \{i\}\text{is not infeasible}\}$
while $\mathscr{F} \neq \emptyset$ **do**
    $RCL \leftarrow \textsc{ConstructRCL}(parameter^{\dagger})$
    Select $s^{\star}$ at random from RCL
    $S \leftarrow S \cup \{s^{\star}\}$
    $f(S) \leftarrow f(S) + c_{s^{\star}}$
    $\mathscr{F} \leftarrow \{s \in \mathscr{F} \setminus \{s^{\star}\} : S \cup \{s\}\text{is not infeasible}\}$
end while
return S

† The *parameter* is the maximum cardinality K of the RCL or α to control the greediness/randomness of the greedy procedure.

Random multistart procedures

A *random multistart procedure* is an iterative method. At each iteration an initial solution is built with a random construction procedure. The multistart procedure outputs the best solution found in all iterations.
The construction procedure is carried out under different conditions at each iteration to allow obtaining different solutions. In a similar way to greedy procedures, a random construction procedure selects one element from the ground set E at a time and adds it to the solution set until a solution is obtained.

Random multistart pseudocode

Random multistart pseudocode (minimization case)

$f_{best} \leftarrow \infty$
$Stop \leftarrow$ **false**
while (**not** $Stop$) **do**
    $S \leftarrow$ RandomSolution()
    if $(f(S) < f_{best})$ **then**
        $S^{\star} \leftarrow S$
        $f_{best} \leftarrow f(S^{\star})$
    end if
    UpdateStopingCriterion†
end while
return $S^{\star}$

† This procedure must allow to change the value of the variable *Stop* to **true** at some iteration of the random multistart procedure.

Semi-greedy multistart procedures

A *semi-greedy multistart procedure* is an iterative method. At each iteration an initial solution is built with an adaptive greedy heuristic. The multistart procedure outputs the best solution found in all iterations.
Since at each iteration of the construction procedure of the adaptive greedy heuristic there may exists more than one element of the ground set of elements E that minimizes the greedy function, random tie-breaking must be provided in order to have chances of obtaining different solutions. Otherwise the embedding of the adaptive greedy solution in a multistart procedure will be useless since the same solution will be obtained at each iteration.

Semi-greedy multistart pseudocode

Semi-greedy multistart pseudocode (minimization case)

$f_{best} \leftarrow \infty$
$Stop \leftarrow$ **false**
while (**not** $Stop$) **do**
    $S \leftarrow$ SemiGredy()
    if $(f(S) < f_{best})$ **then**
        $S^{\star} \leftarrow S$
        $f_{best} \leftarrow f(S^{\star})$
    end if
    UpdateStopingCriterion†
end while
return $S^{\star}$

† This procedure must allow to change the value of the variable *Stop* to **true** at some iteration of the random multistart procedure.

GRASP

A greedy randomized adaptive search procedure (GRASP) is the hybridization of a semi-greedy algorithm with a local search method embedded in a multistart framework. The method consists of multiple applications of local search, each starting from a solution generated with a semi-greedy construction procedure. The best local optimum, over all GRASP iterations, is returned as the solution provided by the algorithm.[1]

[1] Resende, M.G.C., and Ribeiro, C.C. Optimization by GRASP: Greedy Randomized Adaptive Search Procedures, Springer, 2013

Greedy Randomized Adaptive Search Procedure

Let S be a solution, f a cost function, *BestS* the best solution found, and $f^\star$ the objective function value of the best solution found.

$f^\star \leftarrow \infty$
StopCriterion $\leftarrow$ **false**
while not *StopCriterion* **do**
    $S \leftarrow$ *RandomizedGreedyConstruction*(α)
    $S \leftarrow$ *LocalSearch*(S)
    if $(f(S) < f^\star)$ **then**
        BestS $\leftarrow S$
        $f^\star \leftarrow f(S)$
    end if
    Update Stop Criterion
end while
return *BestS* and $f^\star$

Randomized Greedy Construction

Randomized Greedy Procedure

function *RandomizedGreedyConstruction*(α)
Let S be the ground set of elements in the solution, c_i the cost of element $i \in E$, g a greedy function, and μ a threshold value.
$S \leftarrow \emptyset$
$f(S) \leftarrow 0$
$\mathscr{F} \leftarrow \{i \in E : S \cup \{i\}$ is not infeasible$\}$
Compute the greedy choice function $g(i)$ for all $i \in \mathscr{F}$
while $(\mathscr{F} \neq \emptyset)$ **do**
    $g_{min} \leftarrow \min_{i \in \mathscr{F}} \{g(i)\}$
    $g_{max} \leftarrow \max_{i \in \mathscr{F}} \{g(i)\}$
    $\mu \leftarrow g_{min} + \alpha\,(g_{max} - g_{min})$
    $RCL \leftarrow \{i \in \mathscr{F} : g(i) \leq \mu\}$
    Select randomly an element $i^\star$ from RCL
    $S \leftarrow S \cup \{i^\star\}$
    $f(S) \leftarrow f(S) + c(i^\star)$
    $\mathscr{F} \leftarrow \{i \in \mathscr{F} \setminus \{i^\star\} : S \cup \{i\}$ is not infeasible$\}$
    Compute the greedy choice function $g(i)$ for all $i \in \mathscr{F}$
end while
return $f(S)$