
HEURÍSTICAS DE REFINAMENTO

Heurísticas de refinamento são também chamadas de busca local e consistem em métodos que buscam por soluções melhores no entorno ou vizinhança de uma dada solução.

Neste trabalho prático, vamos reusar nossas heurísticas construtivas feitas no trabalho prático anterior para gerar as soluções iniciais e depois refiná-las com as buscas locais ou heurísticas de refinamento para o problema da mochila presentes no arquivo em anexo. Neste, temos duas heurísticas de refinamento: troca um valor (1opt ou *swap 1*) e troca dois valores (2opt ou *swap 2*).

O QUE DEVE SER FEITO

1. Você deve montar combinações das heurísticas construtivas com as de refinamento, como, por exemplo, *random* com 1opt, *random* com 2opt, *random* com 1opt e 2opt, e assim sucessivamente, até termos *partial greedy* com 1opt e 2opt.
2. Explique qual combinação apresentou o melhor desempenho. Por que ela obteve esse resultado? Manteve-se a tendência do trabalho prático anterior? Quem ganhou lá ganhou aqui também?
3. Monte agora uma busca local (heurística de refinamento) que seja a combinação do 1-Opt com um procedimento de *backtracking parcial* ou remoção sistemática de itens da mochila e inserção gulosa. Esta busca local é ativado toda vez que a busca local 1-Opt ou 2-Opt atingir um ótimo local. O objetivo é realizar uma exploração sistemática e limitada para encontrar um vizinho longe do ótimo local atual, mas que ainda possa ser promissor. Esta estratégia poderosa combina diversificação com intensificação.

A adicione o procedimento abaixo, Algoritmo 2.0.1, após a chamada do 1-Opt ou 2-Opt e ou 1-Opt com 2-Opt. Repita as análises. O desempenho melhorou?

Algorithm 2.0.1: Procedimento **BacktrackParcial**(S, S^*, k_{max})

Data: S : solução atual
Data: S^* : melhor solução
Data: k_{max} : limite de backtrack
Result: S : solução diversificada (ponto de reinício)

```

1  $k_h \leftarrow \text{RandInt}(1, \min\{k_{max}, |S|\})$  ; // Seleciona aleatoriamente o
   número de itens a serem removidos
2  $\bar{S} = I \setminus S$ ;
3  $N \leftarrow \{\}$ ;
4 for  $k \leftarrow 1$  até  $k_h$  do
5    $i \leftarrow \text{RandSelection}(S \setminus N)$ ;
6    $S \leftarrow S \setminus \{i\}$ ;
7   Shuffle( $\bar{S}$ );
8   for  $j \in \bar{S}$  do
9     if  $f(S \cup \{j\}) > f(S)$  then
10        $N \leftarrow N \cup \{j\}$ ;
11        $S \leftarrow S \cup \{j\}$ ;
12        $\bar{S} \leftarrow \bar{S} \setminus \{j\}$ ;
13       if  $f(S) > f(S^*)$  then
14          $S^* \leftarrow S$ ;
15       end
16     end
17   end
18 end
19 return  $S$ ;

```
