

Análise e Projeto de Algoritmos

Meta-heurísticas

Prof. Bruno Bruck



Centro de Informática
U F P B

Meta-heurísticas

- Algoritmos que visam encontrar uma solução de qualidade, e eventualmente até mesmo a solução ótima
- Diferentemente de heurísticas convencionais, meta-heurísticas possuem caráter geral
 - ▣ Podem ser aplicadas a qualquer problema de otimização
 - ▣ Possuem mecanismos para sair de ótimos locais

Meta-heurísticas

- Se dividem em duas categorias
 - ▣ Baseadas em busca local
 - Exemplos:
 - GRASP
 - Iterated Local Search (ILS)
 - Busca Tabu
 - Simulated Annealing
 - ▣ Baseadas em busca populacional
 - Exemplos:
 - Algoritmos Genéticos
 - Colônia de Formigas
 - Otimização por Enxame de Partículas



GRASP

GRASP

- GRASP (Greedy Randomized Adaptive Search Procedure)
 - ▣ ou Procedimento de Busca Adaptativa Gulosa e Randômica em Português

- Consiste em duas fases
 - ▣ 1) Fase de construção
 - ▣ 2) Fase de busca local

GRASP

□ 1) Na fase de construção

- Uma solução é iterativamente gerada elemento por elemento
- A cada iteração, os elementos candidatos a "entrarem" para a solução são colocados em uma lista de candidatos
 - Ordenados por meio de um critério guloso que especifica o benefício de se adicionar cada candidato à solução
 - Essa lista é chamada de **Lista de Candidatos Restrita** (LCR)

GRASP

□ 1) Na fase de construção

```
procedimento Construcao( $g(\cdot)$ ,  $\alpha$ ,  $s$ );  
1   $s \leftarrow \emptyset$ ;  
2  Inicialize o conjunto  $C$  de candidatos;  
3  enquanto ( $C \neq \emptyset$ ) faça  
4       $g(t_{min}) = \min\{g(t) \mid t \in C\}$ ;  
5       $g(t_{max}) = \max\{g(t) \mid t \in C\}$ ;  
6       $LCR = \{t \in C \mid g(t) \leq g(t_{min}) + \alpha(g(t_{max}) - g(t_{min}))\}$ ;  
7      Selecione, aleatoriamente, um elemento  $t \in LCR$ ;  
8       $s \leftarrow s \cup \{t\}$ ;  
9      Atualize o conjunto  $C$  de candidatos;  
10 fim-enquanto;  
11 Retorne  $s$ ;  
fim Construcao;
```

GRASP

□ 1) Na fase de construção

- Um candidato é escolhido aleatoriamente dentre os melhores candidatos da LCR
- O parâmetro α controla o grau de aleatoriedade do algoritmo e é um valor no intervalo $[0, 1]$
 - $\alpha = 0$ faz com que o algoritmo seja puramente guloso, ou seja, o melhor candidato sempre é escolhido
 - $\alpha = 1$ faz com que o algoritmo se torne puramente aleatório, ou seja, qualquer candidato pode ser escolhido

GRASP

□ 2) Na fase de busca local

- Heurísticas construtivas geralmente **não** levam a ótimos locais
- Dai a importância de uma fase de busca local para tentar melhorar a solução encontrada
- Nessa fase qualquer algoritmo de busca local pode ser utilizado
 - Exemplo: VND

GRASP

```
procedimento  $GRASP(f(\cdot), g(\cdot), N(\cdot), GRASP_{max}, s)$   
1   $f^* \leftarrow \infty$ ;  
2  para ( $Iter = 1, 2, \dots, GRASP_{max}$ ) faça  
3       $Construcao(g(\cdot), \alpha, s)$ ;  
4       $BuscaLocal(f(\cdot), N(\cdot), s)$ ;  
5      se ( $f(s) < f^*$ ) então  
6           $s^* \leftarrow s$ ;  
7           $f^* \leftarrow f(s)$ ;  
8      fim-se;  
9  fim-para;  
10  $s \leftarrow s^*$ ;  
11 Retorne  $s$ ;  
fim  $GRASP$ 
```



ILS



ILS

- ILS (Iterated Local Search)
 - ▣ Meta-heurística que se baseia na ideia de que um procedimento de busca local pode ser melhorado gerando novas soluções de partida
 - Obtidas por meio de **perturbações** na **solução ótima local**

ILS

procedimento *ILS*

```
1   $s_0 \leftarrow \text{GeraSolucaoInicial}();$   
2   $s \leftarrow \text{BuscaLocal}(s_0);$   
3  enquanto (os critérios de parada não estiverem satisfeitos) faça  
4       $s' \leftarrow \text{Perturbacao}(\text{histórico}, s);$   
5       $s'' \leftarrow \text{BuscaLocal}(s');$   
6       $s \leftarrow \text{CritérioAceitacao}(s, s'', \text{histórico});$   
8  fim-enquanto;  
fim ILS;
```