
BUSCA LOCAL ITERADA

A **busca local iterada** ou *iterated local search* (ILS), veja Algoritmo 4.0.1, é uma meta-heurística projetada para resolver problemas de otimização, especialmente aqueles onde métodos de busca local tradicionais tendem a ficar em presos vizinhanças não promissoras, i.e., em ótimos locais.

O princípio básico do ILS é simples e se baseia em uma sequência de iterações:

1. **Busca local** (*local search*): Começa-se com uma solução inicial ou a solução perturbada da iteração anterior e aplica-se um método de busca local, normalmente o **VND**, para melhorá-la até atingir um ótimo local, desejavelmente, melhor do que o corrente.
2. **Perturbação** (*perturbation*): Em seguida, aplica-se uma perturbação aleatória de baixa intensidade, i.e., sem tornar a solução corrente completamente aleatória, para que se escape dessa vizinhança não promissora.
3. **Critério de aceitação** (*acceptance criterion*): A nova solução perturbada é usada como ponto de partida para a próxima busca local. O critério de aceitação decide se a solução resultante dessa nova busca local será a nova *melhor solução atual* para as próximas iterações. O critério de aceitação determina qual das duas soluções S' , o ótimo local anterior, ou S'' , o novo ótimo local, será usada para a próxima perturbação. Este pode ser:
 - Gulosa: Aceita S'' apenas se for melhor que S' .
 - Metáforas (e.g., *Simulated Annealing*): Aceita soluções piores com alguma probabilidade para aumentar a diversificação.

É possível aumentar o nível de perturbação quando a solução corrente não melhora após algumas tentativas. Pode assim tentar explorar melhor a região. Isto dá origem ao *smart iterated local search*

Algorithm 4.0.1: Iterated Local Search (ILS)

Entrada: S : solução inicial
Entrada: h_{\max} : número máximo de iterações
Saída : S^* : melhor solução encontrada

```

1  $S \leftarrow \text{Buscalocal}(S)$  // Encontra um ótimo local inicial
2  $S^* \leftarrow S$ 
3  $h \leftarrow 0$ 
4 while  $h < h_{\max}$  do
5    $S' \leftarrow \text{Perturbação}(S)$  // perturba a solução para escapar do
    ótimo local
6    $S'' \leftarrow \text{BuscaLocal}(S')$  // aplica busca local (VND) na solução
    perturbada
7    $S \leftarrow \text{CritérioAceitação}(S, S'')$ 
8   if  $f(S) > f(S^*)$  then
9      $S^* \leftarrow S$  // atualiza a melhor solução global
10  end
11   $h \leftarrow h + 1$ 
12 end
13 return  $S^*$ 

```

O QUE DEVE SER FEITO

1. Com o código em anexo, você deve montar um conjunto de experimentos que mostrem qual combinação de ideias é a melhor. No código, temos um **SILS** implementado. Nele podemos usar o **VND** ou o **RVND**. Qual foi o melhor?
2. Teste o critério de aceitação usando a estratégia gulosa, usada no código em anexo, e uma que você deverá implementar usando a metáfora do *simulated annealing*, veja Algoritmo 4.0.3. Isto é, vamos aceitar soluções piores desde que o critério do SA seja aceito. Qual combinação de foi melhor?

Algorithm 4.0.2: Smart Iterated Local Search (ILS)

Entrada: S : solução inicial
Entrada: h_{\max} : número máximo de iterações
Saída : S^* : melhor solução encontrada

```

1  $S \leftarrow \text{Buscalocal}(S)$  // Encontra um ótimo local inicial
2  $S^* \leftarrow S$ 
3  $h \leftarrow 0$ 
4  $p \leftarrow 1$ 
   // nível de perturbação
5 while  $h < h_{\max}$  do
6    $k \leftarrow 0$ 
7   while  $k < k_{\max}$  do
8      $S' \leftarrow \text{Perturbação}(S, p)$  // perturba a solução para escapar do
       ótimo local
9      $S'' \leftarrow \text{BuscaLocal}(S')$  // aplica busca local (VND) na solução
       perturbada
10     $S \leftarrow \text{CritérioAceitação}(S, S'')$ 
11    if  $f(S) > f(S^*)$  then
12       $S^* \leftarrow S$  // atualiza a melhor solução global
13       $h \leftarrow 1$ 
14       $k \leftarrow 0$ 
15       $p \leftarrow 1$ 
16    else
17       $h \leftarrow h + 1$ 
18    end
19     $k \leftarrow k + 1$ 
20  end
21   $p \leftarrow p + 1$ 
22   $h \leftarrow h + 1$ 
23 end
24 return  $S^*$ 

```

Algorithm 4.0.3: Critério de aspiração usando *Simulated Annealing*

```
1  $T \leftarrow T^0$  // no início do algoritmo configuramos a temperatura
   inicial
   // no critério de aceitação fazemos
2  $\Delta \leftarrow f(S'') - f(S)$ 
3  $\delta \leftarrow \text{Rand}(0,1)$  // valor aleatório entre 0 e 1
4 if  $\delta < e^{-\frac{\Delta}{T}}$  then
5   |  $S \leftarrow S''$ 
6 end
   // No final o laço principal atualizamos a temperatura
7  $T \leftarrow \alpha T$ 
```
