

Late Acceptance Hill-Climbing

Marcone Jamilson Freitas Souza^{1,2,3}

Puca Huachi Vaz Penna¹

¹ Departamento de Computação

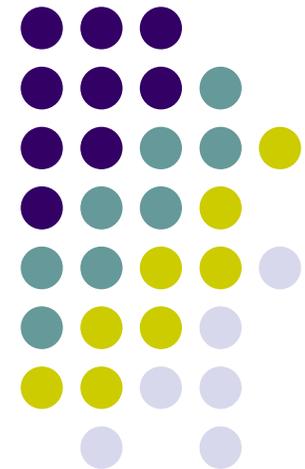
¹ Programa de Pós-Graduação em Ciência da Computação
Universidade Federal de Ouro Preto

² Programa de Pós-graduação em Modelagem Matemática e
Computacional / CEFET-MG

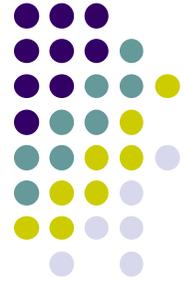
³ Programa de Pós-graduação em Instrumentação, Controle e
Automação de Processos de Mineração / ITV/UFOP

www.decom.ufop.br/prof/marcone, www.decom.ufop.br/puca

E-mail: {marcone,puca}@ufop.edu.br

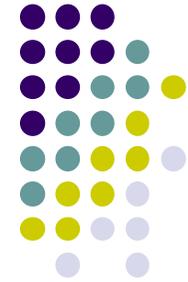


Late Acceptance Hill-Climbing



- *Late Acceptance Hill-Climbing* (LAHC):
 - Método da Subida com Aceitação Tardia
- Metaheurística de busca local
- Proposta por Edmund Burke e Yuri Bykov na *International Conference of Practice and Theory of Automated Timetabling* (PATAT) de 2008
- Publicação em periódico: Burke e Bykov (2017) (<http://www.sciencedirect.com/science/article/pii/S0377221716305495>)

Late Acceptance Hill-Climbing



- Consiste, basicamente, de uma modificação do Método da Subida/Descida Randômica (*Random Uphill/Descent Method*)
- Princípio de funcionamento do método:
 - A cada iteração gerar uma solução vizinha **qualquer** e aceitá-la se:
 - 1) ela for melhor que a solução corrente **ou**
 - 2) ela for melhor que uma das soluções geradas anteriormente, não necessariamente a última (avaliação tardia)
- Desta forma, **soluções piores** que a corrente **podem ser aceitas**

Late Acceptance Hill-Climbing



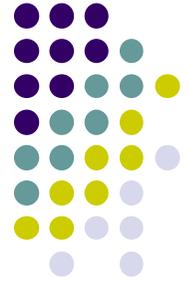
- Trabalha com uma lista F de valores das últimas l soluções geradas:
- $F = \{F_0, F_1, F_2, \dots, F_{l-1}\}$
 - $F_0 = f(s_0), F_1 = f(s_1), F_2 = f(s_2), \dots, F_{l-1} = f(s_{l-1})$
- Uma solução vizinha $s' \in N(s)$ é aceita se:
 - $f(s') \leq f(s)$, isto é, s' é melhor ou igual a s **OU**
 - $f(s') \leq F_p$, isto é, s' é melhor ou igual à solução que está na p -ésima posição da lista F

Late Acceptance Hill-Climbing



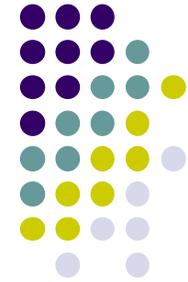
- O índice p é usado para percorrer a lista F e é atualizado como se segue:
 - $p \leftarrow (p + 1) \bmod l$
- Se $l = 20$ e $p = 3$, então $p = (3+1) \bmod 20 = 4$
- Assim, dada uma solução s , o valor de uma solução vizinha s' aleatoriamente gerada, isto é, $f(s')$, é comparado com:
 - 1) o valor da solução s e com
 - 2) o valor da solução armazenada na posição 4 de F , isto é, com F_4

Late Acceptance Hill-Climbing



- No algoritmo LAHC apresentado a seguir:
 - m é o número máximo de rejeições consecutivas
 - r é o contador do número de rejeições consecutivas
 - Cada elemento F_p da lista F é inicializado com o valor da solução s_0 inicial, isto é,
 - $F_p = f(s_0)$ para todo $p = 0, 1, \dots, l$

Late Acceptance Hill-Climbing



Algorithm 1: Late Acceptance Hill-Climbing

Input: Initial solution S , list size l and maximum consecutive rejections m
LAHC(S, l, m)

```
1  foreach  $p \in \{0, \dots, l - 1\}$  do
2     $F_p \leftarrow f(S)$ 
3   $S^* \leftarrow S$ 
4   $p \leftarrow r \leftarrow 0$ 
5  while  $r \leq m$  and time limit is not reached do
6     $N \leftarrow$  selected neighborhood structure
7     $S' \leftarrow$  random neighbor  $N(S)$ 
8    if  $f(S') \leq f(S)$  or  $f(S') \leq F_p$  then
9      if  $f(S') < f(S)$  then
10        $r \leftarrow 0$ 
11        $S \leftarrow S'$ 
12       if  $f(S) < f(S^*)$  then
13          $S^* \leftarrow S$ 
14      $F_p \leftarrow f(S)$ 
15      $p \leftarrow (p + 1) \bmod l$ 
16      $r \leftarrow r + 1$ 
17  return  $S^*$ 
```

Fonte: Toffolo et al., 2018 (<https://doi.org/10.1016/j.cor.2017.08.002>)