

Metaheurísticas

Busca Tabu

Felipe Augusto Lima Reis

felipe.reis@ifmg.edu.br



**INSTITUTO
FEDERAL**
Minas Gerais

Sumário

1 Busca Tabu

BUSCA TABU

Busca Tabu

- A **Busca Tabu** é uma estratégia de Busca Local proposta por Fred Glover em 1986 [Glover, 1986] [Luke, 2013];
- Armazena estados já visitados anteriormente para evitar passar novamente pelos mesmos caminhos [Coppin, 2004]
 - Para isso, mantém uma lista de soluções candidatas visitadas, denominada **Lista Tabu**, e somente visita-as novamente após um determinado intervalo de tempo [Souza, 2011].

[Souza, 2011] indica que o trabalho de Hansen [Hansen, 1986], publicado em 1986 de forma independente do trabalho de Glover, pode ser considerado também Busca Tabu. Tal trabalho usa uma variação do Steepest Ascent Hill Climbing. [Luke, 2013] utiliza esse algoritmo para construção do pseudo-algoritmo da Busca Tabu.

Busca Tabu

- A Busca Tabu explora o espaço de soluções movendo-se de uma solução para outra
 - A busca inicializa com uma solução inicial s ;
 - Em seguida, o algoritmo explora um subconjunto da vizinhança da solução atual, gerando soluções candidatas;
 - A solução candidata s' com o melhor valor na região torna-se a nova solução corrente [Luzia and Rodrigues, 2009];
 - Para evitar que o algoritmo volte ao ponto/solução anterior (ciclagem)¹, é utilizada a **Lista Tabu** [Luke, 2013].

¹Dependendo da aplicação, a solução não volta ao ponto anterior, mas gera uma mesma solução prévia.

Busca Tabu

- A Lista Tabu contém uma série de movimentos ou soluções temporariamente proibidas;
- A lista possui uma quantidade fixa de elementos, permitindo que as visitas sejam “esquecidas com o tempo”²
 - Quando um elemento novo entra na lista, um antigo deve sair, caso a lista esteja cheia;
 - Ao limitar o tamanho possibilita a redução da quantidade de memória necessária para execução do algoritmo [Luzia and Rodrigues, 2009].

²Tal lista é denominada *short-term memory* (memória de curta duração).

Busca Tabu

- O uso da Lista Tabu proíbe o algoritmo de manter-se em um ponto de ótimo local ou retornar a uma solução anterior
 - O algoritmo é forçado, então, a percorrer novos caminhos ou utilizar novas soluções [Luke, 2013];
- Para maior variabilidade, o algoritmo pode optar por buscar soluções em pontos longes da solução atual quando uma solução máxima ou mínima for encontrada
 - Com isso, o algoritmo consegue buscar em uma área maior do espaço de soluções [Luzia and Rodrigues, 2009].

Busca Tabu

- Uma variação da Busca Tabu é denominada **Feature-based Tabu Search**³ [Luke, 2013]
 - Pode ser usada em problemas onde as características (*features*) que compõe a solução podem ser identificadas e separadas;
 - A Lista Tabu é implementada de modo a proibir *features* previamente utilizadas;
 - As *features* podem ser modificadas por um procedimento de modo a gerar soluções distintas [Luke, 2013].

³Tradução livre: Busca Tabu Baseada em Características.

Funções de Aspiração

- A Lista Tabu evita a ciclagem⁴, porém pode limitar soluções ou movimentos que ainda não foram feitos
 - Para solução o problema, pode ser definida uma **Função de Aspiração**, que retira o status de Tabu de uma determinada solução, sob algumas condições;
- Aplicações:
 - Uma aplicação seria considerar uma função $A(f(s)) = f(s^*)$, onde s^* é a melhor solução até o momento;
 - Caso um movimento (ou ação) gere um vizinho melhor que s^* , a solução é permitida;
 - Este tipo é denominado Aspiração por Objetivo⁵ [Souza, 2011].

⁴Durante um período de tempo $|T|$, correspondente à cardinalidade (tamanho) da lista.

⁵Outros tipos de Função de Aspiração, sob outro critérios, também podem ser implementadas.

Parâmetros e Critério de Parada

- Parâmetros (comuns):
 - Cardinalidade da Lista Tabu;
 - Cardinalidade das Soluções Candidatas;
 - Função de Aspiração;
- Critérios de Parada:
 - Quantidade de iterações;
 - Valor objetivo atingido (ou valor próximo);
 - Número máximo de iterações sem melhora no valor da melhor solução [Souza, 2011].

Busca Tabu em Espaços Contínuos

- A Busca Tabu funciona adequadamente, sem alterações, apenas em espaços/problemas discretos;
- Em espaços de valores reais (contínuos), a utilização de uma Lista Tabu pode necessitar de uma comparação flexível
 - Tal comparação teria uma margem de tolerância, fazendo que valores próximos fossem considerados iguais;
 - Tal procedimento evitaria que pequenas diferenças de valores fossem considerados resultados distintos [Luke, 2013].

INTENSIFICAÇÃO E DIVERSIFICAÇÃO

Intensificação

- **Intensificação** é um termo utilizado para estratégias que buscam concentrar a pesquisa em regiões mais promissoras do espaço de soluções [Souza, 2011]
 - A intensificação pode fazer o algoritmo voltar a uma região já visitada para explorar mais efetivamente as soluções;
 - A intensificação pode também incorporar atributos de outras boas soluções, gerando novos resultados [Souza, 2011].

Diversificação

- **Estratégias de Diversificação** são procedimentos que buscam variar consideravelmente os atributos de novas soluções em relação às melhores soluções obtidas até o momento [Souza, 2011]
 - Essas estratégias podem utilizar uma memória de longo prazo para direcionar a pesquisa para regiões pouco exploradas;
 - A diversificação pode ser aplicada apenas em condições especiais, como em regiões onde a busca já se exauriu;
 - Um número N de iterações sem melhoria da solução pode ativar essas estratégias [Souza, 2011].

ALGORITMO

Algoritmo

- O pseudo-algoritmo da Busca Tabu pode ser visto abaixo:

```
01: n ← número de vizinhos a serem gerados
02: S ← solução candidata inicial qualquer
03: Melhor ← S
04: L ← lista tabu vazia
05: insira S em L
06: repita
07:   R ← GerarVizinho( S )
08:   repita n - 1 vezes
09:     W ← GerarVizinho( S )
10:     se W não está em L e (Qualidade(W) > Qualidade(R) ou R está em L) então
11:       R ← W
12:   se R não está em L e Qualidade(R) > Qualidade(S) então
13:     S ← R
14:     insira R em L
15:   se Qualidade(S) > Qualidade(Melhor) então
16:     Melhor ← S
17: até que Melhor seja a solução ideal ou o tempo tenha esgotado
18: devolva Melhor
```

Tabu Search

Fonte: [Luzia and Rodrigues, 2009]

VANTAGENS E DESVANTAGENS

Vantagens e Desvantagens

- Vantagens:
 - Capacidade de lidar com problemas combinatoriais difíceis;
 - Capacidade de lidar com restrições complexas [Luzia and Rodrigues, 2009].
- Desvantagens:
 - Funciona adequadamente apenas em espaços discretos;
 - Se o espaço de busca for muito grande, a busca poderá ficar presa em uma mesma vizinhança;
 - Aumentar Lista Tabu pode reduzir esse problema, porém requer mais memória, além de aumentar as comparações;
 - Em problemas extremamente grandes, essa situação pode se agravar devido a limitações de tamanho da Lista Tabu [Luzia and Rodrigues, 2009] [Luke, 2013].

Referências I



Coppin, B. (2004).

Artificial intelligence illuminated.

Jones and Bartlett illuminated series. Jones and Bartlett Publishers, 1 edition.



Glover, F. (1986).

Future paths for integer programming and links to artificial intelligence.

Computers & Operations Research, 13(5):533–549.

Applications of Integer Programming.



Hansen, P. (1986).

The steepest ascent mildest descent heuristic for combinatorial programming.

In Congress on numerical methods in combinatorial optimization, Capri, Italy, pages 70–145.



Luke, S. (2013).

Essentials of Metaheuristics (Second Edition).

lulu.com, 2 edition.

[Online]; Disponível em: <https://cs.gmu.edu/~sean/book/metaheuristics/>.



Luzia, L. F. and Rodrigues, M. C. (2009).

Estudo sobre as metaheurísticas.

[Online]; acessado em 22 de Setembro de 2020. Disponível em:

<https://www.ime.usp.br/~gold/cursos/2009/mac5758/LeandroMauricioHeuristica.pdf>.

Referências II



Souza, M. J. F. (2011).

Inteligência computacional para otimização.

[Online]; acessado em 12 de Maio de 2021. Disponível em: <http://www.decom.ufop.br/prof/marcone/Disciplinas/InteligenciaComputacional/InteligenciaComputacional.pdf>.