UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE EDUCAÇÃO SUPERIOR DO ALTO VALE DO ITAJAÍ – CEAVI
ENGENHARIA DE SOFTWARE

GRACIELE RODRIGUES

IMPLEMENTAÇÃO E AVALIAÇÃO DO ALGORITMO DE BUSCA TABU EM INSTÂNCIAS DE INTERAÇÕES COMBINATÓRIAS

IBIRAMA

2024

GRACIELE RODRIGUES

IMPLEMENTAÇÃO E AVALIAÇÃO DO ALGORITMO DE BUSCA TABU EM INSTÂNCIAS DE INTERAÇÕES COMBINATÓRIAS

Relatório apresentado à disciplina de Métodos Quantitativos, do curso de Engenharia de Software do Centro de Educação Superior do Alto Vale do Itajaí (CEAVI), da Universidade do Estado de Santa Catarina (UDESC), como requisito parcial para a aprovação na disciplina.

Orientador: Prof. Marcelo de Souza

IBIRAMA

2024

RESUMO

Este trabalho investiga a aplicação do algoritmo de Busca Tabu para a otimização de problemas combinatórios com interações entre elementos. O objetivo foi avaliar a eficiência do algoritmo na busca de soluções de alta qualidade em instâncias com um grande número de elementos e interações, por meio de múltiplas replicações. Os resultados indicam que a Busca Tabu consegue melhorar as soluções ao longo das iterações, apresentando uma relação entre tempo de execução e qualidade das soluções encontradas. Este estudo contribui para a análise da aplicação de técnicas de busca local em problemas de otimização combinatória.

Palavras-chave: Busca Tabu, Otimização Combinatória, Interações entre Elementos, Busca Local.

ABSTRACT

This work investigates the application of the Tabu Search algorithm for optimizing combinatorial problems with interactions between elements. The goal was to evaluate the algorithm's efficiency in finding high-quality solutions in instances with a large number of elements and interactions through multiple replicates. The results indicate that Tabu Search improves solutions over iterations, showing a relationship between execution time and solution quality. This study contributes to the analysis of the application of local search techniques in combinatorial optimization problems.

Keywords: Tabu Search, Combinatorial Optimization, Interactions between Elements, Local Search.

SUMÁRIO

1 INTRODUÇÃO

A Busca Tabu é uma abordagem heurística amplamente reconhecida e aplicada em problemas de otimização combinatória, especialmente aqueles que envolvem um número elevado de possibilidades e que, devido à sua complexidade computacional, não podem ser resolvidos por métodos exatos. Essa técnica é baseada em uma estratégia de exploração local que evita ciclos repetitivos durante o processo de busca. Para isso, utiliza uma estrutura de memória chamada *tabu list*, que registra as soluções já visitadas, orientando o algoritmo a explorar regiões do espaço de soluções que ainda não foram adequadamente investigadas ou a focar em soluções promissoras que poderiam ter sido negligenciadas por abordagens tradicionais de busca local (GLOVER, 1986).

A principal vantagem da busca tabu é sua capacidade de escapar de ótimos locais, uma limitação comum em outros métodos de otimização, ao permitir movimentos de piora em sua trajetória de busca. Isso torna a busca tabu particularmente eficaz em cenários de otimização onde os espaços de soluções são grandes, não-lineares e altamente complexos (GLOVER, 1989).

Neste estudo, propõe-se o desenvolvimento de um algoritmo de busca tabu voltado para a resolução do problema de seleção de componentes em medicamentos. O objetivo é maximizar a interação benéfica entre os componentes, uma questão que envolve uma grande quantidade de possibilidades de combinação e requer uma abordagem eficiente para encontrar soluções viáveis e otimizadas. A utilização da busca tabu nesse contexto visa não apenas resolver o problema de forma eficaz, mas também explorar potenciais interações entre componentes que poderiam ser negligenciadas por técnicas de otimização convencionais.

2 PROBLEMA

A formulação de medicamentos envolve a combinação de múltiplos componentes químicos, e muitas dessas substâncias podem interagir entre si, afetando a eficácia do medicamento final. Algumas interações podem resultar em benefícios, enquanto outras podem gerar efeitos indesejados.

O problema central deste estudo é a seleção ótima de componentes químicos para a formulação de um novo medicamento, levando em consideração as interações entre os diferentes componentes químicos e maximizando o benefício total. O objetivo específico é escolher um subconjunto de componentes a partir de 10 bancos de dados distintos, que fornecem informações sobre as interações entre os pares de componentes, de modo a maximizar o benefício resultante dessa combinação. Essas interações são expressas por valores numéricos: valores positivos indicam benefícios terapêuticos (eficácia no combate à doença), valores negativos indicam prejuízos (efeitos colaterais indesejados) e valores nulos ou zero indicam que não há interação significativa.

O desafio consiste em selecionar os componentes químicos que resultem na maior interação positiva global, considerando as limitações de cada banco de dados e as possíveis interações entre os componentes. Dado que o número de combinações possíveis de componentes é extremamente grande, o problema configura-se como um caso típico de otimização combinatória, onde métodos exatos se tornam inviáveis devido à complexidade computacional.

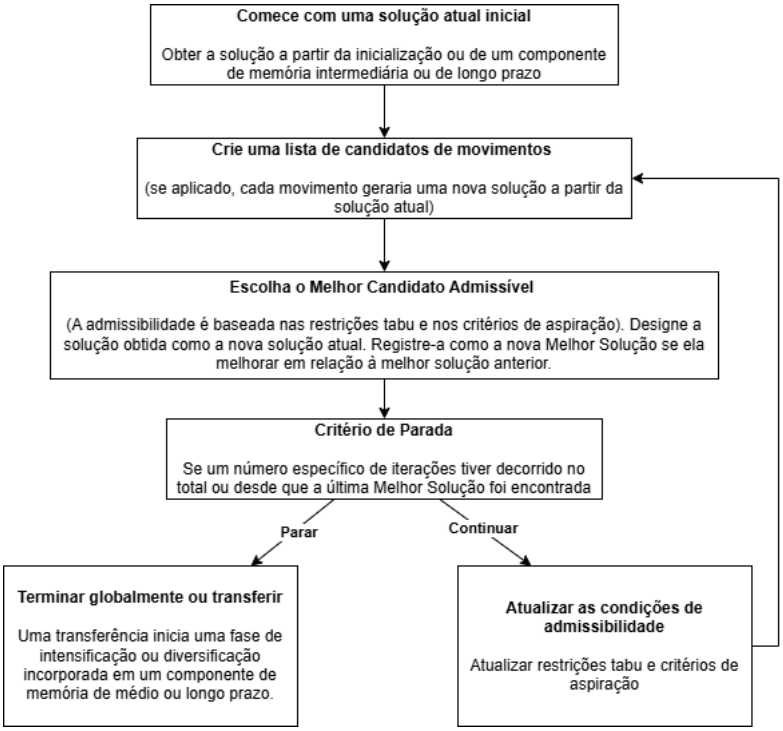
Este estudo fornece os dados da formulação da busca tabu, detalhando os parâmetros utilizados na implementação do algoritmo. Para validação dos resultados, o estudo foi desenvolvido em Python, e os resultados do algoritmo de busca tabu são apresentados, demonstrando a eficácia da abordagem na otimização das interações entre os componentes selecionados.

3 ALGORITMO DE BUSCA TABU

A principal característica da busca tabu é sua capacidade de escapar de ótimos locais, permitindo que o algoritmo realize movimentos que aparentemente pioram a solução, mas que, na verdade, ajudam a explorar novas regiões do espaço de soluções.

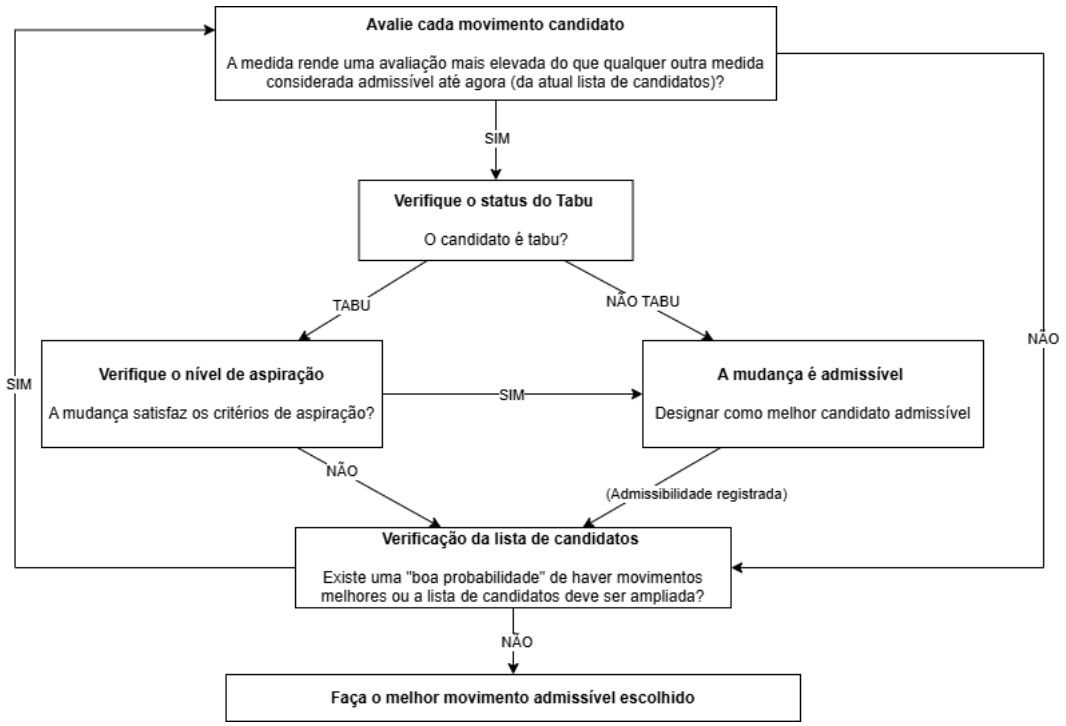
A técnica faz uso de uma memória chamada *tabu list*, que registra as soluções já visitadas, evitando a repetição de movimentos e garantindo que o algoritmo explore áreas inexploradas. A Figura 1 representa o funcionamento da Busca Tabu e a Figura 2 representa a seleção do melhor candidato admissível, conforme elaborado por Glover (1990), mostrando os principais componentes e como o algoritmo evita ciclos e melhora a exploração do espaço de soluções.

Figura 1 – Tutorial Busca Tabu



Fonte: Adaptado de Glover, Fred. "Tabu search: A tutorial."Interfaces 20.4 (1990).

Figura 2 – Selecionar o melhor candidato admissível.



Fonte: Adaptado de Glover, Fred. "Tabu search: A tutorial."Interfaces 20.4 (1990).

A Figura 3 representa um algoritmo tabu clássico. A variável p_{max} controla o número máximo de iterações do algoritmo. A variável q_{max} controla o número de iterações sem melhoria. A linha aspiração permite que um movimento pertencente à lista tabu seja realizado caso resulte na melhoria da melhor solução – $s*$ (GOLDBARG; GOLDBARG; LUNA, 2017).

N(s) representa a vizinhança da configuração s.

f(s) simboliza o valor da configuração s.

Figura 3 – Pseudocódigo da Busca Tabu

Algoritmo 1: Algoritmo de Busca Tabu

Result: Solução s^*

Gerar uma solução inicial s e definir $s^* \leftarrow s$;
Inicializar a Lista Tabu T e os contadores p e q;
while $p < p_{max}$ ***e*** $q < q_{max}$ **do**
    Selecionar o melhor vizinho $s' \in N(s) \backslash T$;
    Selecionar o melhor vizinho $s'' \in N(s) \cap T$;
    if $f(s') < f(s'') \wedge f(s'') < f(s^*)$ **then**
        $s \leftarrow s'$ (*Aspiração*);
    end
    if $f(s') < f(s^*)$ **then**
        $s^* \leftarrow s'$;
        $q \leftarrow 0$;
    end
    if $f(s') < f(s)$ **then**
        Adicionar o movimento inverso (s', s) na lista T e atualizar T;
    end
    $p \leftarrow p + 1$;
    $q \leftarrow q + 1$;
end

Fonte: Adaptado de GOLDBARG, Elizabeth. *Otimização Combinatória e Meta-heurísticas - Algoritmos e Aplicações*. Rio de Janeiro: GEN LTC, 2015. E-book. p. 72. ISBN 9788595154667. Disponível em: <https://app.minhabiblioteca.com.br/reader/books/9788595154667/>. Acesso em: 30 nov. 2024.

3.1 IMPLEMENTAÇÃO

Nesta seção, apresenta-se a implementação do algoritmo de busca tabu para resolver o problema de otimização combinatória proposto. A Figura 4 apresenta o pseudocódigo da implementação realizada.

Figura 4 – Pseudocódigo do algoritmo de Busca Tabu Implementado

Algoritmo 2: Pseudocódigo do algoritmo de Busca Tabu Implementado

```
Result: Melhor solução encontrada s* e o benefício associado f(s*)
Definir parâmetros iniciais:
    MAX_ITERATIONS,
    TABU_TENURE,
    NUM_REPLICATES,
    INSTANCE_DIR,
    OUTPUT_FILE;
Obter arquivos de instância no diretório INSTANCE_DIR;
for cada arquivo de instância inst do
    Ler número de elementos num_elements e interações da instância;
    for cada replicação r de 1 até NUM_REPLICATES do
        Gerar uma solução inicial s aleatória;
        Definir a melhor solução inicial s* ← s;
        Calcular o benefício associado f(s*) ← f(s);
        Inicializar a lista Tabu T;
        for cada iteração t de 1 até MAX_ITERATIONS do
            Gerar um conjunto de soluções vizinhas N(s) alterando um elemento de s;
            Filtrar N(s) para incluir apenas soluções que não estejam na lista Tabu T;
            for cada solução vizinha s' em N(s) do
                Calcular o benefício f(s');
                Adicionar s' à lista de vizinhos válidos se s' ∉ T;
            end
            if não houver vizinhos válidos then
                Encerrar o algoritmo;
            end
            Selecionar a solução vizinha s' com o maior benefício f(s');
            if f(s') > f(s*) then
                Atualizar a melhor solução encontrada: s* ← s';
                Atualizar o benefício: f(s*) ← f(s');
            end
            Adicionar s à lista Tabu T;
            Atualizar a solução atual: s ← s';
            Se necessário, remover os itens mais antigos de T para manter o tamanho
              da lista dentro de TABU_TENURE;
        end
        Armazenar o benefício f(s*) e o tempo de execução da replicação;
    end
    Calcular e armazenar resultados agregados (melhor benefício, média de
      benefícios e tempo médio);
end
Salvar os resultados no arquivo OUTPUT_FILE;
```

Fonte: Elaborado pelo autor, 2024.

3.1.1 Representação da Solução

A solução é representada como um vetor binário de tamanho *num_elements*, onde cada elemento do vetor corresponde à presença ou ausência de um componente químico na fórmula do medicamento. Ou seja, se o valor da posição *i* do vetor for 1, o componente *i* está incluído na solução; caso contrário, ele está excluído. A Figura 5 apresenta um exemplo de solução com 10 componentes possíveis, nesta representação, os componentes 1, 3, 5, 6 e 9 estão incluídos no medicamento, enquanto os outros são excluídos.

Figura 5 – Representação da solução com vetor binário

```
Solução:  [1, 0, 1, 0, 1, 1, 0, 0, 1, 0]
```

Fonte: Elaborado pelo autor, 2024.

3.1.2 Solução inicial

A solução inicial utilizada na Busca Tabu foi gerada de forma aleatória, atribuindo a cada componente químico um valor binário, onde 1 indica que o componente está incluído na formulação do medicamento e 0 significa que ele não está presente. Essa abordagem inicial visa fornecer uma base para a evolução das soluções ao longo das iterações da busca. A partir dessa configuração aleatória, o algoritmo explora o espaço de soluções por meio da troca de componentes, buscando gradualmente a combinação que maximize o benefício das interações entre os componentes selecionados.

3.1.3 Geração da Vizinhança

Para cada iteração do algoritmo, o processo de geração de vizinhos parte da solução atual. A vizinhança é explorada criando novas soluções que diferem da solução atual em apenas um elemento. No código implementado, essa alteração é feita invertendo o valor de um bit da solução: se o valor de um componente é 1, ele é alterado para 0, e vice-versa. Esse processo é repetido para cada componente do vetor de solução, gerando uma nova solução para cada alteração individual. As soluções resultantes dessa operação são chamadas de "vizinhos". Para cada vizinho gerado, é calculado o benefício total da solução, somando as interações entre os componentes que estão presentes na solução. A avaliação leva em consideração o grau de interação entre os componentes, conforme descrito nos bancos de dados. Se a interação entre dois componentes for positiva, ela contribui para o benefício, e se for negativa, diminui o benefício da solução.

Antes de adicionar um vizinho à lista de candidatos, o algoritmo verifica se essa solução já foi explorada recentemente, utilizando uma lista chamada Tabu List. Essa lista armazena as soluções visitadas recentemente, de modo a evitar que o algoritmo fique preso em um ciclo de soluções já exploradas. Se uma solução não estiver na lista Tabu, ela é considerada para possível aceitação. No entanto, se uma solução tabu gera um benefício maior do que a melhor solução encontrada até o momento, o algoritmo permite a aspiração, ou seja, aceita essa solução mesmo que ela esteja na lista Tabu.

Após gerar todos os vizinhos e avaliá-los, o algoritmo escolhe o vizinho com o maior benefício, ou seja, aquele que proporciona a melhor solução possível naquele momento. A melhor solução encontrada é então atualizada e a solução atual passa a ser esse melhor vizinho.

Esse processo de geração de vizinhos e escolha do melhor vizinho é repetido por um número determinado de iterações (*MAX_ITERATIONS*). Durante o processo, a lista Tabu é atualizada com a nova solução escolhida, para garantir que o algoritmo não repita soluções recentes.

3.1.4 Escolha da melhor solução

Ao final do processo (após o número máximo de iterações ou quando o algoritmo não encontra mais vizinhos válidos), a melhor solução é aquela que obteve o maior benefício durante toda a execução do algoritmo.

4 RESULTADOS EXPERIMENTAIS

4.1 DESCRIÇÃO DAS INSTÂNCIAS

As instâncias utilizadas no experimento consistem em bancos de dados contendo informações sobre 500 componentes químicos. Cada interação é caracterizada por três valores: os dois componentes químicos envolvidos e o grau de benefício (ou prejuízo) gerado pela combinação. Os valores de benefício podem ser positivos (indicando benefício na combinação) ou negativos (indicando prejuízo, como efeitos colaterais).

Esses dados foram organizados em arquivos no formato .sparse, onde cada linha contém a identificação de dois componentes e o grau de interação entre eles. Foram fornecidos 10 diferentes arquivos de instâncias, oriundos de fornecedores distintos.

4.2 CONFIGURAÇÃO EXPERIMENTAL

Para teste do algorítimo, foram utilizadas 10 instâncias que representam diferentes bancos de dados contendo informações sobre 500 componentes químicos e suas interações. Para o problema em questão, onde a média de interações fornecidadas em cada instância é de 12.402, foi adotado 15% desse valor como tamanho da lista tabu, resultando em aproximadamente 1.860 elementos. Esse valor visa equilibrar o histórico de soluções armazenadas, permitindo ao algoritmo evitar ciclos e repetições, ao mesmo tempo que mantém a flexibilidade necessária para explorar novas soluções. A busca Tabu foi configurada com um número máximo de iterações de 1000 e foi realizado 10 replicações por instância, a fim de garantir robustez nos resultados.

4.3 RESULTADOS DAS REPLICAÇÕES

Os resultados das replicações para cada instância foram obtidos após a execução da busca Tabu. Em cada replicação, a solução inicial foi gerada aleatoriamente, e a busca visou maximizar o benefício total da interação entre os componentes selecionados. O algoritmo executou 1000 iterações, durante as quais diferentes soluções foram avaliadas e ajustadas com base nas interações.

Os melhores resultados (benefício total) para cada replicação foram registrados, e o tempo de execução também foi medido. A Tabela 1, apresenta os resultados agregados por instância, incluindo o melhor valor benéfico encontrado, a média dos valores e o tempo médio de execução em segundos.

Tabela 1 – Resultados da Busca Tabu.

Instância	Melhor Valor	Valor Médio	Tempo Médio (s)
BD1	60653.00	60415.60	1001.8255
BD2	58259.00	58088.10	999.9868
BD3	55305.00	54998.40	976.3782
BD4	50554.00	50317.30	935.4778
BD5	61828.00	61617.10	1103.1685
BD6	58129.00	57683.30	1140.8564
BD7	62726.00	62563.20	2208.5917
BD8	50288.00	49942.80	1407.2610
BD9	56178.00	55932.40	1165.5909
BD10	57343.00	57043.10	988.0383

Fonte: Elaborado pelo autor, 2024.

4.4 ANÁLISE DOS RESULTADOS

4.4.1 Análise dos Melhores Valores Obtidos

Os melhores valores variaram entre 50.288 (BD8) e 62.726 (BD7), refletindo diferenças significativas entre as instâncias. O maior valor obtido no BD7 sugere um espaço de busca mais favorável, com maior número de interações benéficas de alta intensidade. Por outro lado, o BD8 apresentou o menor valor, possivelmente devido à ausência de interações positivas significativas.

4.4.2 Consistência dos Resultados

A diferença entre o melhor valor e o valor médio foi pequena, indicando consistência no desempenho do algoritmo. Por exemplo, no BD1, o melhor valor foi 60.653, enquanto o valor médio foi 60.415,6, com uma diferença relativa de apenas 0,39%. Isso demonstra que a Busca Tabu foi capaz de explorar o espaço de soluções de forma robusta.

4.4.3 Tempo Médio de Execução

Os tempos médios de execução variaram de 935,48 segundos (BD4) a 2208,59 segundos (BD7). Essa variação pode ser atribuída à complexidade do espaço de busca em cada instância. O maior tempo médio no BD7 reflete a necessidade de maior esforço computacional para explorar interações complexas. Instâncias com menor complexidade, como BD4 e BD10, apresentaram tempos significativamente menores.

4.4.4 Desempenho Geral do Algoritmo

A Busca Tabu demonstrou ser eficaz para o problema proposto, encontrando soluções de alta qualidade e apresentando consistência em diferentes cenários. Entretanto, o tempo de execução em instâncias mais complexas sugere a necessidade de ajustes nos parâmetros do algoritmo para melhorar a eficiência.

4.5 CONCLUSÃO

O algoritmo de Busca Tabu mostrou-se eficaz na maximização das interações benéficas entre componentes químicos, com resultados consistentes em diferentes instâncias. Entretanto, o aumento do tempo de execução em instâncias mais complexas destaca a necessidade de melhorias na eficiência do método. Estudos futuros podem explorar ajustes nos parâmetros do algoritmo e abordagens híbridas para alcançar melhor desempenho computacional.

4.6 DISPONIBILIDADE DE CÓDIGO E DADOS

O código-fonte utilizado para a implementação do algoritmo de Busca Tabu, assim como os bancos de dados simulados, está disponível publicamente no seguinte repositório GitHub:

- <https://github.com/GracieleRodrigues-dev/tabu-search-algorithm/>

5 CONSIDERAÇÕES FINAIS

O presente estudo demonstrou a eficácia do algoritmo de Busca Tabu na resolução de problemas de otimização relacionados à seleção de componentes químicos para o desenvolvimento de medicamentos. Os resultados indicaram que o método foi capaz de maximizar as interações benéficas entre os componentes, apresentando soluções de alta qualidade e consistência, mesmo em instâncias com diferentes graus de complexidade.

Observou-se que, embora o algoritmo tenha produzido resultados satisfatórios em termos de qualidade das soluções, o tempo de execução variou significativamente entre as instâncias. Essa variação está diretamente associada à complexidade do espaço de busca, indicando que ajustes nos parâmetros do algoritmo e estratégias para redução de tempo computacional poderiam ser explorados em estudos futuros.

O estudo reforça o potencial de métodos heurísticos, como a Busca Tabu, na solução de problemas complexos em áreas críticas, como a indústria farmacêutica, onde a otimização de recursos e a eficácia dos medicamentos é essencial. Pesquisas futuras podem incluir comparações com outros métodos, como algoritmos genéticos ou híbridos, e a aplicação em problemas com restrições adicionais ou bancos de dados reais. Além disso, explorar técnicas como o aprendizado de máquina para guiar a busca pode ser um caminho promissor para alcançar resultados ainda mais robustos e eficientes.

REFERÊNCIAS

GLOVER, F. Future paths for integer programming and links to artificial intelligence. **Computers & operations research**, Elsevier, v. 13, n. 5, p. 533–549, 1986.

GLOVER, F. Tabu search—part i. **ORSA Journal on computing**, Informs, v. 1, n. 3, p. 190–206, 1989.

GLOVER, F. Tabu search: A tutorial. **Interfaces**, INFORMS, v. 20, n. 4, p. 74–94, 1990.

GOLDBARG, E.; GOLDBARG, M.; LUNA, H. **Otimização combinatória e metaheurísticas: algoritmos e apliacações**. [S.l.]: Elsevier Brasil, 2017.

RODRIGUES, Graciele. **Tabu Search Algorithm for Combinatorial Optimization**. Disponível em: <https://github.com/GracieleRodrigues-dev/tabu-search-algorithm/>.