

SCATTER SEARCH

Trabalho de Algoritmos Bioinspirados

Gustavo Henrique da Cunha

Departamento de Ciência da Computação - DCOMP
Coordenadoria do Curso de Ciência da Computação - CCOMP
Universidade Federal de São João del Rei - UFSJ

Setembro de 2024

- 1 Introdução
 - Contextualização
- 2 O Algoritmo
 - Projeto Básico
 - Fluxograma
 - Pseudocódigo
- 3 Problema do Caixeiro-Viajante
- 4 SS para o PCV
 - Método de Geração Diversificado e Método de Refinamento
 - Método de Atualização do Conjunto de Referência
 - Método de Geração de Subconjuntos e Método de Combinação de Soluções
 - Parâmetros

- 1 Introdução
 - Contextualização
- 2 O Algoritmo
 - Projeto Básico
 - Fluxograma
 - Pseudocódigo
- 3 Problema do Caixeiro-Viajante
- 4 SS para o PCV
 - Método de Geração Diversificado e Método de Refinamento
 - Método de Atualização do Conjunto de Referência
 - Método de Geração de Subconjuntos e Método de Combinação de Soluções
 - Parâmetros

- O *Scatter Search* (SS), ou Busca Dispersa, é uma metaheurística populacional que trabalha com a recombinação de soluções num conjunto de referência, com o objetivo de gerar novas soluções.
- Com origem no artigo de F. Glover em 1977, e posteriormente popularizada por Manuel Laguna e Rafael Martí em 2003, o SS obteve bons resultados em diversos problemas de otimização combinatória [1].

- A ideia básica do algoritmo é a construção e atualização de um conjunto de referência que contém as melhores soluções geradas por um método de geração de soluções diversificadas.
- É importante notar que o critério que define uma solução como "melhor" neste caso não se baseia apenas no valor da função objetivo, mas também numa métrica de diversificação.

- 1 Introdução
 - Contextualização
- 2 O Algoritmo
 - Projeto Básico
 - Fluxograma
 - Pseudocódigo
- 3 Problema do Caixeiro-Viajante
- 4 SS para o PCV
 - Método de Geração Diversificado e Método de Refinamento
 - Método de Atualização do Conjunto de Referência
 - Método de Geração de Subconjuntos e Método de Combinação de Soluções
 - Parâmetros

- A construção básica do algoritmo pode ser definida em cinco métodos:
 - 1 **Método de Geração Diversificada:** Cria uma coleção de soluções diversificadas, a partir de critérios de qualidade e diversidade. O número de soluções geradas é grande, geralmente pelo menos 10 vezes maior que o conjunto de referência.
 - 2 **Método de Refinamento:** Transforma cada solução gerada em uma ou mais soluções melhoradas.
 - 3 **Método de Atualização do Conjunto de Referência:** Cria e mantém o conjunto de referência, que contém as b melhores soluções encontradas, onde b é um valor pequeno. As soluções podem entrar neste conjunto com base nos seus valores na função objetivo ou nos valores de diversidade.

- A construção básica do algoritmo pode ser definida em cinco métodos:
 - ④ **Método de Geração de Subconjuntos:** A partir do conjunto de referência, gera diversos subconjuntos que serão usados para serem recombinaados em novas soluções.
 - ⑤ **Método de Combinação de Soluções:** A partir de um subconjunto, cria uma ou mais soluções.

Fluxograma

- A Figura 1 [2] mostra o fluxograma que representa a construção básica do algoritmo.

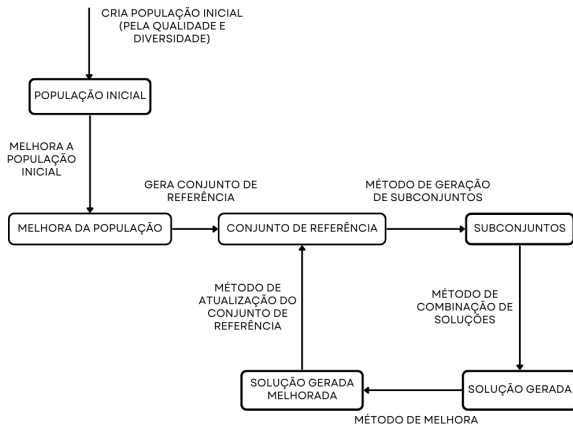


Figura 1: Fluxograma de um SS básico.

procedimento *ScatterSearch*

```
1   $P \leftarrow \emptyset$ ;  
   Use o procedimento Diversificação para construir uma solução  $x$ ;  
   Se  $x \notin P$  então adicione  $x$  a  $P$ , isto é,  $P \leftarrow P \cup \{x\}$ ; caso contrário, descarte  $x$ ;  
   Repita este procedimento até que  $|P| = PSize$ ;  
   Construa  $RefSet = \{x^1, \dots, x^{nbest}\}$ , com  $nbest$  soluções diversificadas de  $P$ ;  
2  Avalie as soluções em  $RefSet$  e ordene-as de acordo com a função de avaliação;  
   (Considere  $x^1$  a melhor solução e  $x^{nbest}$  a pior)  
    $NewSolutions \leftarrow TRUE$ ;  
   enquanto ( $NewSolutions$ ) faça  
3     Gere  $NewSubsets$ , isto é, todos os pares de soluções de  $RefSet$   
       desde que haja pelo menos uma nova solução, isto é, que  $NewSolutions = TRUE$ ;  
        $NewSolutions \leftarrow FALSE$ ;  
       enquanto ( $NewSubsets \neq \emptyset$ ) faça  
4         Selecione o próximo subconjunto  $s$  em  $NewSubsets$ ;  
5         Aplique Combinação de Soluções a  $s$  para obter uma ou mais soluções  $x$ ;  
         se ( $x \notin RefSet$  e  $f(x) < f(x^{nbest})$ ) então  
6            $x^{nbest} \leftarrow x$  e reordene  $RefSet$ ;  
7            $NewSolutions \leftarrow TRUE$ ;  
         fim-se  
8       Remova  $s$  de  $NewSubsets$ ;  
       fim-enquanto;  
   fim-enquanto;  
fim ScatterSearch;
```

Figura 2: Pseudocódigo de um SS básico [3].

- 1 Introdução
 - Contextualização
- 2 O Algoritmo
 - Projeto Básico
 - Fluxograma
 - Pseudocódigo
- 3 Problema do Caixeiro-Viajante
- 4 SS para o PCV
 - Método de Geração Diversificado e Método de Refinamento
 - Método de Atualização do Conjunto de Referência
 - Método de Geração de Subconjuntos e Método de Combinação de Soluções
 - Parâmetros

Problema do Caixeiro-Viajante

- O problema do caixeiro-viajante (PCV) é um problema de otimização combinatória.
- É um problema difícil de resolver, pertencente à classe NP-completo.
- É comum utilizá-lo para testar implementações de heurísticas.

Problema do Caixeiro-Viajante

- Sendo $G = (V, A)$ um grafo com n vértices V e arestas A com um certo peso, é necessário encontrar um circuito de menor distância possível que passe por cada vértice uma única vez e retorne ao vértice inicial.
- A Figura 3 mostra uma instância do PCV com 49 cidades [4].

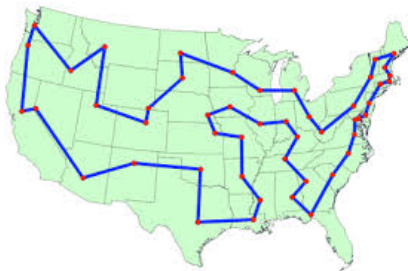


Figura 3: PCV com 49 cidades.

- 1 Introdução
 - Contextualização
- 2 O Algoritmo
 - Projeto Básico
 - Fluxograma
 - Pseudocódigo
- 3 Problema do Caixeiro-Viajante
- 4 SS para o PCV
 - Método de Geração Diversificado e Método de Refinamento
 - Método de Atualização do Conjunto de Referência
 - Método de Geração de Subconjuntos e Método de Combinação de Soluções
 - Parâmetros

- Para utilizar o SS na resolução do PCV, devem ser definidas algumas funções para se adequarem ao problema:
 - Função objetivo: soma das distâncias;
 - Representação: permutação das cidades;
 - Método de geração diversificado;
 - Método de refinamento;
 - Método de atualização do conjunto de referência;
 - Método de geração de subconjuntos;
 - Método de combinação de soluções.

Método de Geração Diversificado

- O método de geração diversificado deve garantir tanto boas soluções quanto soluções diversificadas pelo espaço de busca.
- Nesta etapa, podem ser utilizados diversos métodos que funcionam bem com o PCV, incluindo outras metaheurísticas.
- Neste tutorial, é sugerida a utilização do método GRASP (*Greedy Randomized Adaptive Search Procedure*).

Método de Geração Diversificado

- O GRASP é uma metaheurística construtiva que constrói a solução a partir de um método guloso aleatório, que funciona com base num parâmetro fornecido pelo usuário.
- Assim, é possível utilizá-lo, começando em cidades aleatórias, para construir um conjunto de diferentes soluções.
- O GRASP conta ainda com uma etapa de melhoramento local, que coincide com o *Scatter Search*.

- O método de refinamento utilizado pelo GRASP e recomendado neste tutorial é a busca local.
- A busca local também pode ser implementada com diferentes funções de vizinhança, mas é recomendado o uso do 2-opt.
- Note que outros métodos de busca mais refinados, como uma busca tabu, também poderiam ser utilizados e testados para a comparação de desempenho.

Método de Atualização do Conjunto de Referência

- A partir do método de geração diversificado, será gerado um conjunto de tamanho P , a partir do qual será construído o conjunto de referência.
- Na etapa de construção, devem ser considerados tanto a qualidade das soluções quanto o fator diversidade, para evitar uma rápida convergência para um ótimo local.
- Uma forma de fazer isso é dividir o conjunto de referência, de tamanho b , em duas partes de tamanhos b_1 e b_2 , tal que $b = b_1 + b_2$.

Método de Atualização do Conjunto de Referência

- A construção da parte do conjunto de tamanho b_1 é simples. Basta ordenar os P elementos gerados e selecionar os b_1 melhores, retirando-os de P . Esta parte do conjunto garante a qualidade das soluções.
- Para construir o restante do conjunto, que garante a diversidade, os elementos restantes em P devem ser comparados com os elementos escolhidos para b .
- Esta comparação é feita com base numa métrica definida, que para o PCV pode ser a distância de Hamming ou a diferença nas suas distâncias.
- Assim, a comparação de todos os elementos restantes em P é feita com base na sua distância mínima a algum membro já escolhido, e seleciona-se o elemento que tem a maior distância mínima.

Método de Geração de Subconjuntos

- O tamanho dos subconjuntos pode ser definido com base no método de combinação escolhido. Neste caso, será utilizado tamanho dois.
- Assim, para gerar os subconjuntos de tamanho dois, podem ser feitas todas as combinações de pares para cada item no conjunto de referência.

Método de Combinação de Soluções

- Dado os subconjuntos, deve ser aplicado o método de combinação de soluções nos mesmos.
- Esta fase é equivalente à etapa de cruzamento num algoritmo genético. Assim, pode ser utilizado o *ox-crossover* para gerar novas soluções.
- Com as soluções geradas, o método de melhoramento é novamente aplicado e o conjunto de referência é atualizado.

- Os parâmetros a serem ajustados são:
 - P : número de soluções a serem geradas no método de geração diversificada;
 - b : tamanho do conjunto de referência, onde $b = b_1 + b_2$;
 - b_1 : número de soluções boas escolhidas para compor b ;
 - α : taxa utilizada para o método de construção gulosa aleatória.

Referências Bibliográficas

- [1] RAFAEL MARTÍ, MANUEL LAGUNA e FRED GLOVER. “Principles of Scatter Search”. Em: *European Journal of Operational Research* 169 (2006), pp. 359–372.
- [2] El-Ghazali Talbi. *METAHEURISTICS: FROM DESIGN TO IMPLEMENTATION*. 1st. Hoboken, New Jersey: John Wiley & Sons, Inc., 2009.
- [3] Marccone Jamilson Freitas Souza e Puca Huachi Vaz Penna. *Busca Dispersa (Scatter Search)*.
<http://www.decom.ufop.br/prof/marccone/Disciplinas/InteligenciaComputacional/BuscaDispersa.pptx>. 2021.
- [4] Usman Lateef, Rufai Idowu, Olayinka Olusanya e Adedeji Oluwaseun. “Solving Travelling Salesman Problem Using an Improved Ant Colony Optimization Algorithm”. Em: - 6 (jun. de 2021), pp. 158–170.