

UMA REVISÃO SOBRE MÉTODOS APROXIMATIVOS PARA PROBLEMAS DE OTIMIZAÇÃO COMPUTACIONAL¹

Péricles Pinheiro Feltrin²; Ana Paula Canal³

RESUMO

Este artigo tem por objetivo elucidar os principais métodos aproximativos de resolução de problemas de otimização computacional e sinalizar aqueles que possuem viabilidade ao desenvolvimento de algoritmo paralelo em aceleradores gráficos. Foi realizada uma revisão bibliográfica em artigos da área a fim de identificar a taxonomia dos métodos e, a partir disto, caracterizá-los. Identificou-se que a classe dos métodos aproximativos tem sido utilizada para paralelização em diferentes ambientes de execução paralela. Constatou-se que nem todos os métodos aproximativos tem versões implementadas para arquiteturas com aceleradores gráficos, mesmo tendo características adequada a estas, permitindo o desenvolvimento de pesquisas e elaboração de novas soluções.

Palavras-chave: Heurísticas; Metaheurísticas; Algoritmos; Otimização Combinatória.

1. INTRODUÇÃO

Problemas de Otimização Computacional ou Otimização Combinatória (OC) possuem um conjunto finito de possíveis soluções. Na solução de um problema deste tipo, busca-se otimizar a função objetivo, que deve obedecer o conjunto de restrições impostas pelo problema. Os algoritmos para a solução deste tipo de problema podem ser exatos ou heurísticos.

Um algoritmo exato encontra a solução ótima para o problema, ou seja, a solução exata que é a otimização da função objetivo. Um algoritmo heurístico ou aproximativo encontra uma solução viável ao problema que é uma solução próxima a solução ótima em tempo computacional aceitável (GOLDBARG e LUNA, 2005)

Este artigo tem por objetivo elucidar os principais métodos aproximativos de resolução de problemas de otimização computacional e sinalizar aqueles que possuem viabilidade ao desenvolvimento de algoritmo paralelo em aceleradores

¹ Trabalho de Pesquisa – UNIFRA.

² Acadêmico do Curso de Ciência da Computação – UNIFRA. feltrin10@gmail.com

³ Orientadora. Professora do Curso de Ciência da Computação – UNIFRA. apc@unifra.br

gráficos.

Este estudo trata dos métodos aproximativos, pois o trabalho de Carneiro et al (2014) faz um levantamento de métodos que vem sendo paralelizados com mais frequência em aceleradores gráficos. Da investigação feita por ele, 60% das soluções utilizam métodos aproximativos. Além disto, estes métodos são mais apropriados para aplicação em problemas reais que possuem instâncias grandes do problema. Se estes problemas fossem tratados com métodos exatos inviabilizaria a execução computacional, por não apresentarem tempo de resolução polinomial.

Neste texto, inicialmente está a definição geral de problemas de OC. Na seção 3, são apresentados os conceitos que definem os métodos exatos e aproximativos de resolução de problemas de OC. Na seção 4, a metodologia é descrita, posteriormente são discutidos os resultados do estudo e feitas as considerações finais do trabalho.

2. PROBLEMAS DE OTIMIZAÇÃO COMBINATÓRIA

A resolução de problemas de otimização consiste em procurar a melhor configuração do conjunto de variáveis que o define para alcançar um determinado objetivo. Blum e Roli (2013) definem um Problema de Otimização Combinatória (OC) ou Otimização Computacional $P = (S, f)$ como:

- um conjunto de variáveis $X = \{x_1, \dots, x_n\}$;
- o domínio das variáveis D_1, \dots, D_n ;
- as restrições entre variáveis;
- uma função objetivo f a ser minimizada (ou maximizada, dependendo do problema), onde $f : D_1 \times \dots \times D_n \rightarrow \mathbb{R}^+$

O conjunto de possíveis soluções ao problema é $S = \{s = \{(x_1, v_1), \dots, (x_n, v_n)\} \mid v_i \in D_i\}$ e s satisfaz todas as restrições do problema. S é também conhecido como espaço de busca ou espaço de solução.

Para resolver um problema de OC é preciso encontrar $s^* \in S$ com o mínimo (ou o máximo) valor da função objetivo. s^* é chamado solução ótima global de (S, f) .

Alguns exemplos de problemas de OC são: Problema do Caixeiro Viajante, Problema de Roteamento de Veículos, Problema da Mochila, Problema de

Escalonamento, Problema de Particionamento de Circuitos e Problema do Clique Máximo (GOLDBARG e LUNA, 2005).

3. MÉTODOS PARA RESOLUÇÃO DE PROBLEMAS DE OTIMIZAÇÃO

A solução de Problemas de OC tem aplicação, como por exemplo na diminuição dos custos de fabricação de um produto, no estabelecimento da melhor rota a ser seguida para alcançar um destino ou entregar mercadorias, a otimização do desenho para corte de placas de metal para confecção de produtos, o uso de matéria prima, a elaboração de quadros de horários, entre outros. Devido a sua relevância prática, muitos algoritmos para resolver estes problemas foram desenvolvidos, constituindo os métodos de resolução (GOLDBARG e LUNA, 2005).

Basicamente, duas classes de algoritmos são identificadas na literatura: os métodos exatos e os métodos aproximativos. Os algoritmos exatos são os que garantem encontrar uma solução ótima para uma instância válida do problema em determinado espaço de tempo (CARNEIRO, 2014). Problemas que possuem algoritmos exatos e tempo de resolução polinomial compõem a classe de problemas denominada classe P (*Polinomial Time*). Problemas onde não se conhecem algoritmos que encontram solução em tempo polinomial pertencem à classe NP (*Non Polinomial Time*) e são considerados de difícil solução. Os problemas NP-Difícil caracterizam-se por ter redução polinomial a partir de todo problema pertencente à classe NP. A classe NP-completo são problemas em NP, mais difíceis, sendo definidos como a interseção das classes NP-Difícil e NP (TOSCANI e VELOSO, 2001). Assim, no pior caso de resolução do problema, os algoritmos exatos podem alcançar tempo exponencial de computação, o que não é aceitável para propósitos práticos (BLUM e ROLI, 2003).

Os algoritmos aproximativos abrem mão da solução exata ótima por um conjunto de soluções viáveis que resolve o problema em tempo computacional aceitável (polinomial), obedecendo as restrições do problema (BLUM e ROLI, 2003). Os métodos aproximativos também são chamados de heurísticas. Entre os algoritmos que as heurísticas podem utilizar-se estão os algoritmos construtivos geram uma solução aleatória adicionado a uma solução inicial vazia e os algoritmos de busca que partem de uma solução inicial e iterativamente substituem esta

solução por uma melhor aproximando a vizinhança desta solução no espaço de busca.

4. METODOLOGIA

Este trabalho constitui-se em uma revisão teórica sobre métodos aproximativos para resolução de problemas de Otimização Combinatória. Para tanto, foi realizada pesquisa bibliográfica e análise sobre literatura disponível, a fim de ser feito um recorte conforme a questão elucidada. Esta é uma pesquisa exploratória, para identificar e descrever alguns métodos na resolução de problemas de Otimização Combinatória, promovendo a identificação do estado da arte desta questão.

“A pesquisa bibliográfica remete às contribuições de diferentes autores sobre um assunto” (GONSALVES, 2007, p.38). Para tanto, esta pesquisa baseou-se no artigo de Blum e Roli (2003) “*Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison*”, no artigo de Boussaïd, et al (2013) “*A survey on optimization metaheuristics*” e no artigo de Carneiro, et al. (2014) “Um Levantamento na Literatura sobre a Resolução de Problemas de Otimização Combinatória através do uso de Aceleradores Gráficos”.

5. RESULTADOS

Na classe de métodos aproximativos, estão as metaheurísticas. Blum e Roli (2003) definem as metaheurísticas como algoritmos aproximativos que combinam métodos heurísticos básicos em um *framework* de mais alto nível na exploração de um espaço de busca.

Nos últimos anos, as metaheurísticas tem sido soluções de grande interesse, devido as suas características (BOUSSAÏD et al, 2013). Algumas propriedades que caracterizam as meta-heurísticas são (BLUM e ROLI, 2003):

- a) meta-heurísticas são estratégias que guiam o processo de busca;
- b) o objetivo é explorar eficientemente o espaço de busca a fim de encontrar ou aproximar-se da solução ótima;
- c) incluem desde Busca Local até algoritmos complexos de aprendizagem;

- d) são algoritmos aproximativos e comumente não determinísticos;
- e) não são um problema específico;
- f) podem fazer uso de conhecimento específico do domínio na forma de heurísticas que são controladas por estratégias de mais alto nível.

As descrições dos métodos apresentadas aqui são de Blum e Roli (2003) e Boussaïd et al (2013). Identificou-se que as metaheurísticas podem ser classificadas como métodos baseados em trajetória e métodos baseados em população.

O processo de busca nos métodos de trajetória é caracterizado pela trajetória no espaço de busca, ou seja, começam a busca sobre uma solução inicial e movem-se adiante a partir desta solução, descrevendo uma trajetória no espaço de busca. Os métodos baseados em população, em cada iteração possuem uma população de soluções para explorar o espaço de busca.

Como métodos de trajetória, tem-se Busca Local, *Simulated Annealing* (SA), Busca Tabu, GRASP – *The Greedy Randomized Adaptive Search Procedure*, VNS – *Variable Neighborhood Search*, GLS – *Guide Local Search*, ILS – *Iterated Local Search*.

Métodos baseados em população como os chamados métodos de Computação Evolucionária (EC) são os inspirados por capacidades da natureza e os métodos da classe de Inteligência de Multidão (*Swarm Intelligence* – SI) são baseados no comportamento de multidões.

A Figura 1 traz a classificação das meta-heurísticas conforme Blum e Roli (2003).

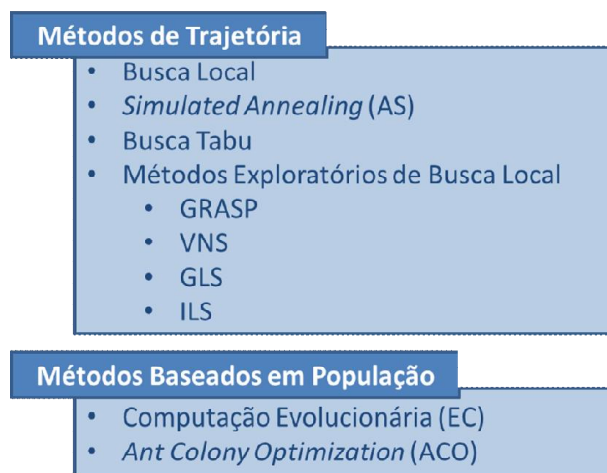


Figura 1: Classificação das Meta-heurísticas conforme Blum e Roli (2003)

A classificação das meta-heurísticas apresentada por Boussaïd et al (2013) também apresenta dois grandes grupos classificando-as: métodos baseados em trajetória e métodos baseados em população, como pode ser observado na Figura 2. Ainda, no segundo grupo traz outros métodos aproximativos.

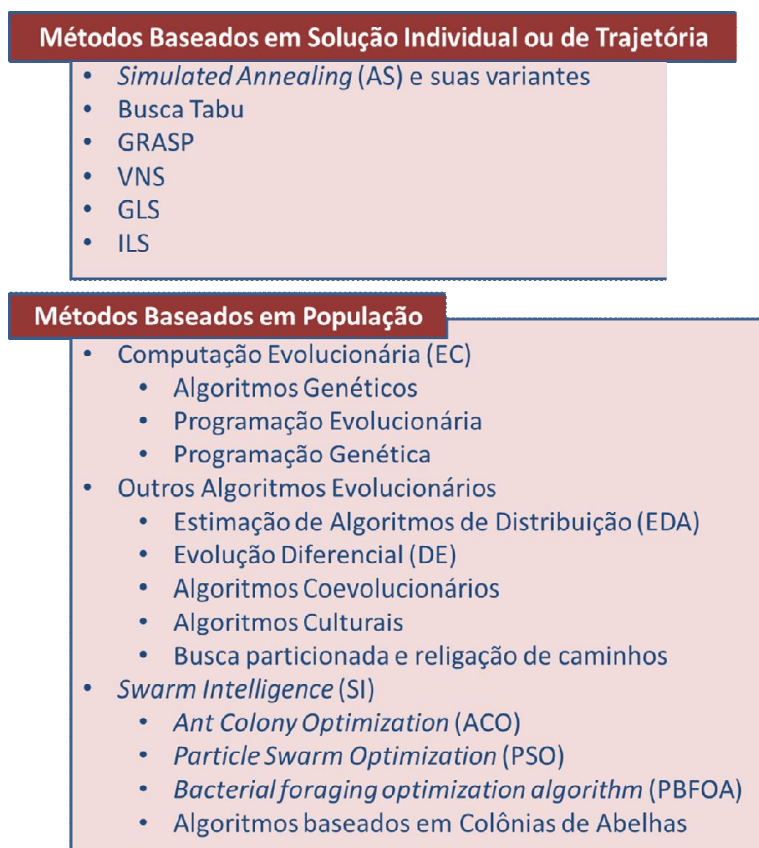


Figura 2: Classificação das Meta-heurísticas conforme Boussaïd et al (2013)

No artigo de Carneiro et al (2014), como resultado de sua pesquisa sobre problemas de otimização combinatória paralelizados em GPUs – *Graphics Processing Unit*, foram encontrados ao todo 50 trabalhos, sendo que 30 destes usaram métodos aproximativos. Destes, 28 trabalhos utilizaram meta-heurísticas e 2 deles usaram heurísticas específicas para os problemas que se propuseram atacar (CARNEIRO et al, 2014).

Carneiro et al (2014) comenta: “Um fato que nos chamou a atenção foi a ausência de trabalhos que utilizem a GRASP, uma metaheurística *multistart*, amplamente utilizada e há muito paralelizada com sucesso em diferentes arquiteturas.” Neste contexto, são caracterizadas a Busca Local e a meta-heurística GRASP.

5.1 Busca Local

No método de Busca Local, em cada iteração ou movimento, é proposta uma nova solução e somente substitui a solução atual se for encontrada uma solução melhor na vizinhança. O movimento procura na vizinhança uma solução que seja melhor que a solução atual. A busca termina assim que for alcançado um mínimo local, ou seja, nenhuma melhoria no processo de busca é possível. A Busca Local pode atingir um mínimo local. Este é um aspecto negativo do método, pois se a solução inicial produzir um mínimo local, este pode estar muito longe do mínimo global do espaço de busca. A Figura 3 ilustra o algoritmo de Busca Local.

```

s ← geraçãoSoluçãoInicial()
repeat
    s ← movimento(V(S))
until nenhuma melhoria ser possível
  
```

Figura 3: Algoritmo básico de Busca Local.

Fonte: BLUM, C.; ROLI, A. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. **ACM Computing Surveys**, vol. 35, no. 3, p. 268-308, September 2003. p.273

Inicialmente, é gerada a solução inicial e em cada iteração do laço de repetição, é feito um movimento e procurado na vizinhança uma solução. As iterações param se nenhuma melhoria na solução é possível.

O algoritmo de Busca Local tem seu desempenho intimamente associado à definição da solução inicial e sua vizinhança. Geralmente, não produz soluções de boa qualidade. Muitas técnicas tem sido usadas para escapar do mínimo local. Mesmo assim, é um algoritmo importante pois é usado por outras metaheurísticas, incluindo GRASP.

5.2 GRASP – *The Greedy Randomized Adaptive Search Procedure*

GRASP - *The Greedy Randomized Adaptive Search Procedure* combina heurísticas construtivas e Busca Local. É uma meta-heurística proposta por Feo e Resende. O algoritmo possui dois passos: construção e busca local. A etapa de construção desenvolve uma solução possível usando uma heurística gulosa de

forma aleatória. No segundo passo, esta solução é usada como solução inicial para uma Busca Local (BOUSSAÏD et al, 2013). Este processo é repetido por um dado número de iterações e a solução encontrada é então retornada. A Figura 4 ilustra este processo.

Na construção da solução viável com a heurística gulosa, a solução candidata é construída iterativamente, ou seja, a cada iteração, um elemento é incorporado na solução parcial, até a solução completa ser construída. Isto significa, que para um dado problema, a solução é um conjunto de elementos. A cada iteração da heurística, a lista de elementos candidatos é formada por todos os elementos que podem ser incluídos na solução parcial. A lista é ordenada conforme a função gulosa que mede a vantagem em se selecionar o elemento. Assim, o elemento que será adicionado à solução parcial é aleatoriamente escolhido entre os melhores candidatos da lista. A lista dos melhores candidatos é chamada de Lista Restrita de Candidatos (RCL). Esta lista pode ser limitada pelo número de elementos ou por algum critério de qualidade. Para a primeira etapa da GRASP, pode ser usada, além da busca local, uma meta-heurística como Busca Tabu, *Simulated Annealing*, entre outras (BOUSSAÏD et al, 2013).

```
repeat
    construir solução viável com heurística gulosa
    aplicar Busca Local na solução construída
until critério parada satisfeito
retorna Melhor Solução Encontrada
```

Figura 4: Algoritmo básico de Busca Local.

Fonte: BOUSSAÏD, I. et al. A survey on optimization metaheuristics. **Information Sciences**, vol 237, p. 82-117, 2013. p.87.

A GRASP é considerada uma meta-heurística *multistart* e que consegue arranjar boas soluções aos problemas de Otimização Combinatória (BLUM e ROLI, 2003). Bons resultados podem ser alcançados com a GRASP e com pequenas variações nos parâmetros do método como o tamanho da lista ou a qualidade da solução, diferentes resultados são alcançados e ajustes nos critérios de qualidade podem levar a melhores soluções.

6. CONSIDERAÇÕES FINAIS

Os métodos aproximativos são adequados à resolução de problemas de Otimização Combinatória, pois estes quando aplicados a problemas práticos apresentam instâncias grandes, tornando sua resolução por métodos exatos, inviável em tempo polinomial de processamento. Pelo fato da existência de grande quantidade de dados e demanda por processamento computacional, muitos métodos aproximativos tem sido paralelizados em diferentes arquiteturas paralelas.

A fim de fazer a paralelização de um método aproximativo, este estudo foi realizado e a partir dele pode-se concluir que não há uma conceituação universalmente aceita pelos autores de Otimização Combinatória quanto a definição de heurísticas e meta-heurísticas. Há uma grande variedade de métodos aproximativos. Novos métodos tem sido criados pela combinação de métodos tradicionalmente usados. Muitos autores comentam que com meta-heurísticas, o próprio pesquisador pode adaptar o método de acordo com as características do problema que está resolvendo, produzindo sua própria meta-heurística.

Quanto à paralelização, métodos do grupo de algoritmos evolucionários (Algoritmos Genéticos e variações), ACO- Ant Colony Optimization e métodos do grupo de algoritmos de trajetória como Busca Tabu, VNS – *Variable Neighborhood Search*, GLS – *Guide Local Search* e ILS – *Iterated Local Search*, já possuem versões paralelas para aceleradores gráficos. Para a GRASP não foram encontradas versões paralelizadas em aceleradores gráficos. Como trabalhos futuros, pretende-se desenvolver o algoritmo paralelo do GRASP para um problema de Otimização Computacional.

REFERÊNCIAS

BLUM, C.; ROLI, A. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. **ACM Computing Surveys**, vol. 35, no. 3, p. 268-308, September 2003.

BOUSSAÏD, I. et al. A survey on optimization metaheuristics. **Information**

Sciences, vol 237, p. 82-117, 2013.

CARNEIRO, T. et al. Um Levantamento na Literatura sobre a Resolução de Problemas de Otimização Combinatória através do uso de Aceleradores Gráficos. In: ***Proceedings of the XXXV Iberian Latin-American Congress on Computational Methods in Engineering – CILAMCE***, Fortaleza, CE, Brazil, November, 2014. 23-26.

GOLDBARG, M. C. e LUNA, H. P. L. **Otimização Combinatória e Programação Linear**: modelos e algoritmos. Rio de Janeiro: Elsevier, 2005. 2ª Ed.

GONSALVES, E. P. **Iniciação à Pesquisa Científica**. Campinas: Editora Alínea, 2007. 4ª Ed.

TOSCANI, L. V.; VELOSO, P. A. S. **Complexidade de algoritmos**. Porto Alegre: Bookman, 2009.