



METAHEURÍSTICA PARALELA HÍBRIDA CPU/GPU DISTRIBUÍDA

**Projeto de Pesquisa enviado ao Programa de Pós-Graduação em Ciência da
Computação
Universidade Federal de Ouro Preto**

Proponentes:

André Luís Barroso Almeida - andlui.eng@gmail.com
Candidato

Orientador:

Marco Antonio Moreira de Carvalho – mamc@ufop.edu.br
Universidade Federal de Ouro Preto – Departamento de Computação

24 de janeiro de 2021

Sumário

1	Introdução	2
1.1	Motivação	3
2	Objetivos geral e específicos	3
2.1	Objetivos Específicos	3
3	Metodologia proposta	4
4	Etapas de execução do projeto com respectivo cronograma de atividades	5
5	Produtos esperados	6
6	Colaborações e infraestrutura	7
6.1	Colaborações ou parcerias já estabelecidas para a execução do projeto	7
6.2	Disponibilidade infraestrutura	7

1 Introdução

A busca por soluções ótimas está se tornando uma necessidade na sociedade contemporânea que suporta cada vez menos desperdícios. A otimização consiste em obter a melhor solução dentro das diversas soluções disponíveis (Hussain et al., 2019), e permeia diferentes áreas: o projeto de novas peças, de novos processos industriais, o planejamento de negócios e de investimentos e em diversas áreas da engenharia e da ciência.

Na prática, nem todos os problemas podem ser resolvidos de forma ótima em um tempo razoável. Existe uma classe de problemas de otimização conhecidos como *NP-difícil* que devido à sua complexidade, não é possível resolvê-los de forma satisfatória em tempo razoável, à medida que suas instâncias aumentam de tamanho e complexidade. Como consequência, neste tipo de cenário, surgiu como alternativa as metaheurísticas, que apesar de não garantirem que a solução ótima será encontrada, são capazes de encontrar uma solução boa em tempo razoável (Pedemonte, 2017).

Metaheurísticas são ferramentas essenciais para resolver problemas *NP-difíceis* (Garey e Johnson, 1979) e vêm demonstrando resultados interessantes nos últimos anos (Essaid et al., 2019). Em geral, são algoritmos estocásticos com alto nível de abstração (Pedemonte, 2017). Por definição, as metaheurísticas orquestram uma interação entre a intensificação e a diversificação das soluções, buscando a melhor interação entre as duas. Na intensificação, a solução encontrada é explorada, já na diversificação novas áreas no espaço de busca são exploradas (Glover e Kochenberger, 2006).

Um grande número de novas metaheurísticas foram propostas com o crescente interesse na área, porém somente algumas se consolidaram na prática (Pedemonte, 2017). Algumas das metaheurísticas mais utilizadas são *Iterated Local Search* (Glover e Kochenberger, 2006; Hoos e Stützle, 2004), *Simulated Annealing* (Kirkpatrick et al., 1983), *Tabu Search* (Glover e Kochenberger, 2006), *Variable Neighborhood Search* (Mladenović e Hansen, 1997) e *Greedy Randomized Adaptive Search Procedure* (Feo e Resende, 1995).

No projeto de uma metaheurística busca-se maior qualidade na solução em menor tempo de execução para diversos problemas e instâncias, o que não é nada trivial, pois a qualidade da solução depende das características do problema e da instância (Crainic, 2016). Com o aumento na demanda por metaheurísticas que possuam a capacidade de obter soluções ótimas ou cada vez mais próximas da ótima em menor tempo e para instâncias e problemas mais complexos, o uso do paralelismo se tornou um caminho natural, não só para reduzir o tempo de execução mas, para melhorar a qualidade da solução (Vidal et al., 2017).

A grande maioria das atuais metaheurísticas não são escaláveis e sua performance tende a crescer à medida que a complexidade dos problemas e instâncias aumentam (Essaid et al., 2019). Neste cenário, as pesquisas na área de metaheurísticas paralelas estão evoluindo para suprir essa demanda. Na última década, o avanço em novos hardwares paralelos tanto para execução tanto para comunicação impulsionaram a pesquisa na área (Alba et al., 2013).

Uma das grandes evoluções em computação paralela, que utilizava somente as *Central Processing Units* (CPUs), foi o surgimento das *Graphics Processing Units* (GPUs), que surgiram como uma plataforma massivamente paralela, obtendo bons resultados em aplicações científicas (Owens et al., 2007). Algumas metaheurísticas foram propostas usando GPUs, porém a grande maioria são bio inspiradas. dentre alguns trabalhos podemos citar algoritmo genético (Pedemonte, 2017), *particle swarm optimization* (Rabinovich et al., 2012) e outros (Langdon, 2013; Maitre et al., 2012).

Dentre as diversas formas de metaheurísticas paralelas, nenhuma utiliza todo o poder computacional de um *cluster* composto por CPUs e GPUs de forma simultânea. A maioria dos trabalhos utiliza somente a CPU ou somente a GPU e quando utiliza ambas, somente um *core* da CPU é utilizado para orquestrar a GPU deixando os outros *cores* em espera (Pedemonte, 2017). O desperdício de uma quantidade considerável de processamento chamou a atenção de alguns pesquisadores (Cardel-

lini et al., 2013; Agulleiro et al., 2012; Rabinovich et al., 2012) que buscaram aproveitar todo o poder computacional disponível. Porém, há diversos desafios no desenvolvimento de uma metaheurística paralela híbrida CPU/GPU. Além de se encontrar poucos trabalhos sobre metaheurísticas paralelas híbridas CPU/GPU não há relatos na literatura de uma metaheurística paralela capaz de utilizar todo o recurso computacional das CPUs/GPUs de forma distribuída.

1.1 Motivação

Resolver problemas de otimização vem se tornando cada vez mais fundamental na sociedade, porém devido a complexidade e o tamanho das instâncias não é possível resolvê-los de forma satisfatória. Neste cenário se faz necessário a utilização de metaheurísticas, que exploram o espaço de busca procurando uma solução boa em um tempo razoável, sem a necessidade de avaliar todas as soluções.

Os problemas *NP-difíceis* possuem geralmente um espaço de busca exponencial ou fatorial o que limita a avaliação de todas as soluções, fazendo-se necessário a utilização de metaheurísticas. Elas navegam por algumas regiões do espaço em busca da melhor solução, porém a quantidade de regiões avaliadas é limitada pelo tempo de execução. Desta forma, existe uma possibilidade de se melhorar a qualidade da solução utilizando paralelismo, aumentando drasticamente a quantidade de regiões e soluções avaliadas em um tempo razoável.

2 Objetivos geral e específicos

Propor e implementar uma arquitetura/framework CPU/GPU para execução de métodos de propósito geral para solucionar problemas de otimização. Devido ao uso do paralelismo será possível explorar novas áreas do espaço de busca, assim aumentando a probabilidade de se encontrar soluções melhores em tempo razoável.

2.1 Objetivos Específicos

1. Elaborar uma revisão de literatura relacionada à métodos heurísticos e metaheurísticos paralelos e identificar suas lacunas;
2. Elaborar métodos heurísticos e metaheurísticos paralelos consistentes e robustos que possam ser utilizadas em contextos reais que permitam a obtenção rápida de soluções próximas da solução ótima;
3. Pesquisar técnicas para refinamento de soluções obtidas por heurísticas e metaheurísticas utilizando diferentes técnicas;
4. Buscar a aplicação prática dos métodos desenvolvidos em contextos reais, a fim de que também seja constituído um avanço para as indústrias;
5. Publicar os resultados deste trabalho de pesquisa em periódicos e eventos nacionais e internacionais.

3 Metodologia proposta

Esse trabalho se desdobrará em três partes: i) uma revisão de literatura relacionada as metaheurísticas e heurísticas paralelas; ii) o desenvolvimento de soluções metaheurísticas paralelas e, iii) uma pesquisa experimental que irá avaliar os algoritmos propostos.

Na primeira parte, será realizada uma revisão integrativa de literatura por meio de levantamento bibliográfico de artigos nacionais e internacionais publicados nos últimos quinze anos contendo os temas: metaheurísticas e heurísticas paralelas. Inicialmente, serão definidos o tema, os objetivos do estudo, as palavras-chaves e seleção da questão norteadora da pesquisa. Em seguida, será realizado um levantamento bibliográfico nas bases de dados *Web of Science* e *Google Scholar*. A busca terá como critérios de inclusão artigos originais e de revisão publicados entre os anos de 2005 a 2020. Posteriormente, será realizada a análise seletiva dos artigos levantados, seguindo critérios de inclusão e exclusão pré-definidos. Os artigos selecionados serão analisados mediante a leitura dos textos na íntegra, buscando-se delimitar as categorias de análise, de modo a responder aos objetivos da pesquisa. Dessa forma, será feita a interpretação dos dados, destacando os trabalhos que trouxeram maior contribuição para responder à questão norteadora da pesquisa. E, por fim, será realizada a discussão das informações e evidências encontradas.

Na segunda parte, será realizado o desenvolvimento de metaheurísticas paralelas e a definição dos problemas que serão utilizados nos experimentos das soluções propostas. Os problemas selecionados deverão pertencer a classe *NP-difícil* e possuir instâncias e resultados já validados pela comunidade científica. As soluções deverão utilizar do paralelismo das CPUs, GPUs, híbrido CPU/GPU e da arquitetura distribuída em um *cluster* de CPUs/GPUs.

Inicialmente, será definida a arquitetura da metaheurística paralela que será utilizada na CPU, idealizada mediante a análise de arquiteturas promissoras que ainda não foram consideradas na literatura. Essa arquitetura deverá explorar regiões do espaço de busca utilizando dos benefícios da arquitetura *multicore* das CPUs como demonstra a Figura 1 (a). Em seguida, após a definição da arquitetura da metaheurística paralela, uma solução será desenvolvida utilizando uma plataforma massivamente paralela (GPU) possibilitando a exploração de diversas regiões do espaço de busca simultaneamente (Figura 1 (b)), essa solução deverá utilizar a CPU somente para orquestrar as tarefas da GPU. Ademais, uma metaheurística paralela híbrida CPU/GPU será desenvolvida. Esta deverá utilizar todo o poder computacional disponível tanto da CPU tando da GPU. Por fim, além de utilizar uma metaheurística paralela híbrida CPU/GPU, será intensificada a busca pela solução utilizando um *cluster*, o que possibilitará escalar a busca (Figura 2).

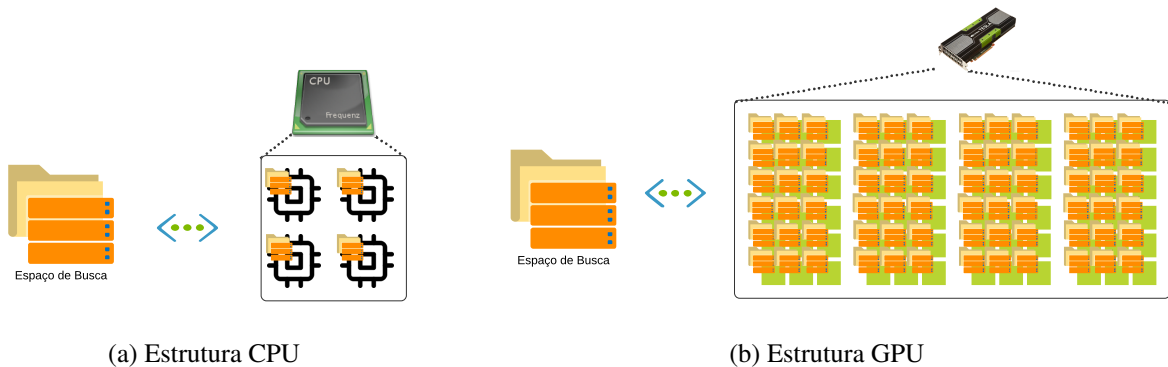


Figura 1: Estruturas CPU e GPU.

Todas as soluções serão desenvolvidas utilizando a linguagem C++ e a plataforma CUDA (Com-

pute Unified Device Architecture) que possibilita o uso das GPUs Nvidia para computação de propósitos gerais (Owens et al., 2007; Kirk e Wen-Mei, 2016). Elas passarão por testes preliminares para validação e por aprimoramentos para melhorias finas das soluções.

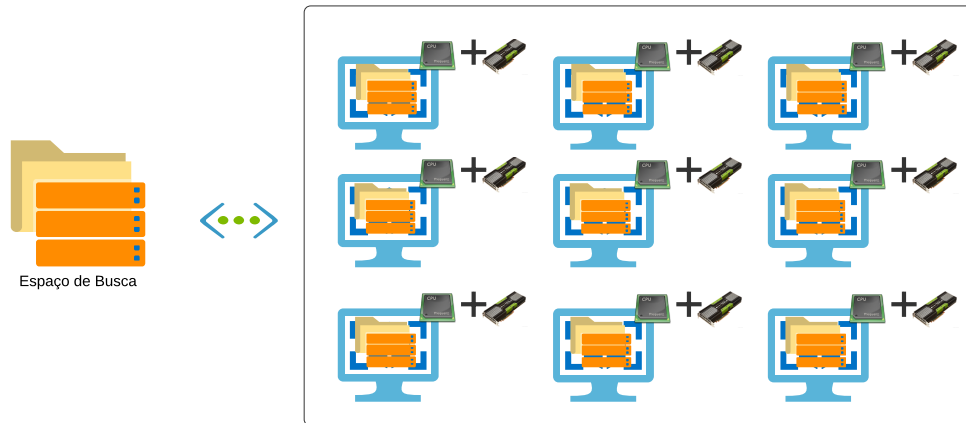


Figura 2: *Cluster* híbrido CPU/GPU

Na terceira parte, uma pesquisa experimental será realizada para avaliar a qualidade da solução e o tempo de execução das metaheurísticas paralelas propostos em diferentes problemas de otimização e em diferentes conjuntos de instâncias. Os resultados encontrados serão comparados com o estado da arte.

4 Etapas de execução do projeto com respectivo cronograma de atividades

As atividades que serão desenvolvidas nesse projeto começarão por uma revisão de literatura que irá desencadear o desenvolvimento da arquitetura/framework da solução paralela das metaheurísticas selecionadas culminando na publicação dos resultados em periódicos e eventos. As atividades específicas são listadas a seguir.

1. Fundamentação Teórica

1.1. Estudo e classificação das principais metaheurísticas e heurísticas paralelas.

2. Especificação da Arquitetura

2.1. Avaliação de técnicas e estruturas de metaheurísticas paralelas promissoras que não foram ainda consideradas na literatura.

3. Definição dos Problemas

3.1. Definição clara e inambígua dos problemas que serão abordados nos experimento computacionais.

4. Implementações Iniciais

- 4.1. Projeto e análise experimental de algoritmos construtivos;
- 4.2. Projeto e análise experimental de algoritmos de busca local;
- 4.3. Implementação das técnicas heurísticas e metaheurísticas consideradas mais promissoras.
5. Validação e Testes
 - 5.1. Verificação da viabilidade das soluções geradas pelos métodos implementados e testes preliminares considerando conjuntos de instâncias da literatura.
6. Aprimoramento das Implementações
 - 6.1. Reavaliação das estruturas e modelagem utilizadas, propostas de alterações de modo a melhorar o desempenho dos métodos propostos;
 - 6.2. Projeto, análise e implementação de técnicas para melhoria fina de soluções (*polishing*).
7. Realização de Experimentos Computacionais Extensivos
 - 7.1. Avaliação estatística das implementações por meio da realização de *benchmarking* utilizando bases de dados disponíveis na literatura.
8. Realização de Estudos Comparativos
 - 8.1. Comparação do desempenho dos métodos propostos e os métodos que compõem o estado da arte em relação aos problemas tratados;
 - 8.2. Análise qualitativa a respeito da complexidade computacional e da eficiência de acordo com os valores obtidos para solução média, tempo de execução médio, convergência e robustez, entre outros critérios.
9. Elaboração de Publicações
 - 9.1. Escrita e submissão de artigos científicos.
10. Participação em eventos científicos
 - 10.1. Participação em simpósios e congressos nacionais, para manutenção de contato com a comunidade acadêmica e o estado da arte relacionado aos temas abordados, além apresentação do trabalho de pesquisa em seus diversos estágios, quando adequado.

O gráfico de *Gantt* da Figura 3 apresenta o cronograma de execução do projeto de pesquisa, indicando cada uma de suas etapas. Por se tratar de um projeto com execução de quatro anos, o período de tempo é discretizado em trimestres, indicando a duração aproximada para determinada atividade.

5 Produtos esperados

A seguir são apresentados os produtos esperados em decorrência da execução deste projeto, indicando o tipo de produto, quantidade, e especificação.

Produto: *SOFTWARES*

Quantidade: 3

Especificação: metaheurísticas paralelas desenvolvidas.

Produto: SUBMISSÃO DE TRABALHOS COMPLETOS A CONGRESSOS CIENTÍFICOS

Quantidade: 4

Especificação: Submissão de trabalhos completos submetidos a congressos científicos nacionais.

Produto: PARTICIPAÇÃO EM CONGRESSOS CIENTÍFICOS

Quantidade: 4

Especificação: Participação em congressos científicos nacionais com possibilidade de apresentação oral de trabalhos completos.

Produto: SUBMISSÃO DE ARTIGOS COMPLETOS EM PERIÓDICOS ESPECIALIZADOS

Quantidade: 4

Especificação: Submissão de artigos completos a periódicos internacionais de índice Qualis restrito.

6 Colaborações e infraestrutura

6.1 Colaborações ou parcerias já estabelecidas para a execução do projeto

Não há colaborações ou parcerias estabelecidas previamente para aplicação neste projeto.

6.2 Disponibilidade infraestrutura

Atualmente já existem no Departamento de Ciência da Computação da Universidade Federal de Ouro Preto as instalações e os *softwares* e *hardwares* necessários para a realização das atividades iniciais da pesquisa previstas neste projeto. Os recursos adicionais necessários para o desenvolvimento deste projeto será via computação em nuvem e custeados pelo proponente.

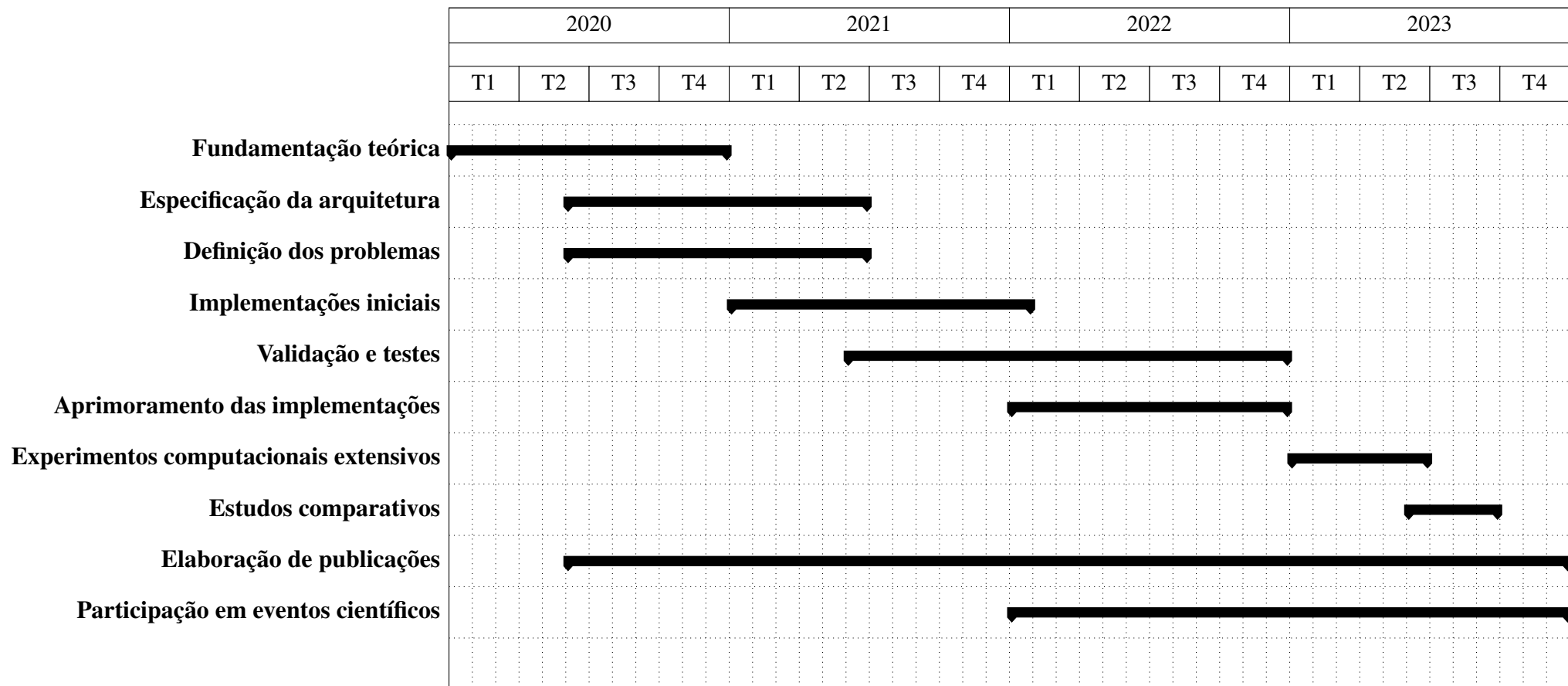


Figura 3: Cronograma de execução das atividades.

Referências

- J. I. Agulleiro, F. Vazquez, E. M. Garzon, e J. J. Fernandez. Hybrid computing: Cpu+ gpu co-processing and its application to tomographic reconstruction. *Ultramicroscopy*, 115:109–114, 2012.
- E. Alba, G. Luque, e S. Nesmachnow. Parallel metaheuristics: recent advances and new trends. *International Transactions in Operational Research*, 20(1):1–48, 2013.
- V. Cardellini, A. Fanfarillo, e S. Filippone. Heterogeneous sparse matrix computations on hybrid gpu/cpu platforms. In *PARCO*, pages 203–212, 2013.
- T. G. Crainic. *Parallel Meta-heuristic Search*. 01 2016. doi: 10.1007/978-3-319-07153-4_40-1.
- M. Essaid, L. Idoumghar, J. Lepagnot, e M. Bréviliers. Gpu parallelization strategies for metaheuristics: a survey. *International Journal of Parallel, Emergent and Distributed Systems*, 34(5):497–522, 2019.
- T. A. Feo e M. G. Resende. Greedy randomized adaptive search procedures. *Journal of global optimization*, 6(2):109–133, 1995.
- M. R. Garey e D. S. Johnson. A guide to the theory of np-completeness. *Computers and intractability*, pages 641–650, 1979.
- F. W. Glover e G. A. Kochenberger. *Handbook of metaheuristics*, volume 57. Springer Science & Business Media, 2006.
- H. H. Hoos e T. Stützle. *Stochastic local search: Foundations and applications*. Elsevier, 2004.
- K. Hussain, M. N. M. Salleh, S. Cheng, e Y. Shi. Metaheuristic research: a comprehensive survey. *Artificial Intelligence Review*, 52(4):2191–2233, 2019.
- D. B. Kirk e W. H. Wen-Mei. *Programming massively parallel processors: a hands-on approach*. Morgan kaufmann, 2016.
- S. Kirkpatrick, C. D. Gelatt, e M. P. Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.
- W. B. Langdon. Large-scale bioinformatics data mining with parallel genetic programming on graphics processing units. In *Massively Parallel Evolutionary Computation on GPGPUs*, pages 311–347. Springer, 2013.
- O. Maitre, F. Krüger, S. Querry, N. Lachiche, e P. Collet. Easga: specification and execution of evolutionary algorithms on gpgpu. *Soft Computing*, 16(2):261–279, 2012.
- N. Mladenović e P. Hansen. Variable neighborhood search. *Computers & operations research*, 24(11):1097–1100, 1997.
- J. D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krüger, A. E. Lefohn, e T. J. Purcell. A survey of general-purpose computation on graphics hardware. In *Computer graphics forum*, volume 26, pages 80–113. Wiley Online Library, 2007.
- M. Pedemonte. Systolic genetic search, a parallel metaheuristic for gpus. 2017.

- M. Rabinovich, P. Kainga, D. Johnson, B. Shafer, J. J. Lee, e R. Eberhart. Particle swarm optimization on a gpu. In *2012 IEEE International Conference on Electro/Information Technology*, pages 1–6. IEEE, 2012.
- P. Vidal, E. Alba, e F. Luna. Solving optimization problems using a hybrid systolic search on gpu plus cpu. *Soft Computing*, 21(12):3227–3245, 2017.