



**Universidade de São Paulo
Instituto de Matemática e Estatística
Departamento de Ciências da Computação**

MAC-5758 Introdução ao Escalonamento e Aplicações

**Um Estudo Abrangente sobre Metaheurística,
incluindo um Histórico**

Aluna: Christiane Zim Zapelini

Professor: Dr. Alfredo Goldman vel Lejbman

Dezembro/2009



Sumário

1. Introdução	3
2. Visão geral	5
3. Propriedades desejáveis	7
4. Classificação das Metaheurísticas	10
4.1. Classificação de Mélian et al	10
4.2. Classificação de Sucupira	12
4.3. . Classificação de Laguna	14
5. Levantamento das Metaheurísticas	15
5.1. Classificação de Mélian et al	15
5.2. Classificação de Sucupira	16
5.3. Classificação de Laguna	18
6. Histórico	19
6.1. Referências Bibliográficas do Histórico	20
7. Conclusão	22
8. Bibliografia	23



1. Introdução

O objetivo dessa monografia é apresentar um estudo sobre metaheurísticas em um contexto mais amplo, não focando somente em métodos ou tipos específicos, mas sim em uma visão mais abrangente sobre o assunto, na **área de Ciência da Computação**. E como decorrência disso, mostrar um levantamento histórico, uma linha de tempo desde seu surgimento que data de 1951 com um artigo sobre métodos estocásticos de otimização de Robbins e Monro até os dias atuais.

Para entender onde o tema dessa monografia está inserido, é necessário algumas considerações iniciais. Em [1] define que **Otimização Combinatória** é um ramo da Ciência da Computação e da Matemática Aplicada que estuda problemas de otimização em conjuntos finitos.

Em um problema de otimização temos uma **função objetivo** e um conjunto de **restrições**, ambos relacionados às **variáveis de decisão**. Os valores possíveis às variáveis de decisão são delimitados pelas restrições impostas sobre essas variáveis, formando um conjunto discreto (finito ou não) de soluções factíveis a um problema. O problema pode ser de minimização ou de maximização da função objetivo. A resposta para o problema de otimização, ou seja, o *ótimo global*, será o menor (ou maior) valor possível para a função objetivo para o qual o valor atribuído às variáveis não viole nenhuma restrição. Em alguns casos, chegamos a valores cuja alteração discreta não conduz a resultados melhores, mas que não são também o *ótimo global* - a essas soluções chamamos de *ótimo local*.

Há muitas classificações possíveis para o problema de otimização, e algumas delas apresentarão métodos exatos e eficientes de resolução. Outras levarão à necessidade de métodos não-exatos (**heurísticas**), uma vez que sua formulação e/ou resolução exatas levariam a uma complexidade intratável.

Apenas a título de curiosidade, em Balieiro [2], cita que o termo heurístico tem origem grega, e cujo sentido é de: “encontrar, descobrir, inventar” expresso pelo verbo grego *euriskw*, com derivados como o adjetivo verbal *euJretov*, que se pode “encontrar ou inventar” e *euJretikov* ”inventivo, engenhoso”. Pelo dicionário português heurística significa “um conjunto de regras e métodos que conduzem à descoberta, à invenção e à resolução de problemas”, e vem da expressão grega *euJristikh; technv*, isto é, a arte relativa à descoberta, à invenção, sobre a qual encontra-se vestígios no tratado *O Método* de Arquimedes.

Segundo Sucupira [3], no contexto da Ciência da Computação, essa palavra não possui um único significado, mas quase sempre está relacionada à ausência de garantias no que diz respeito à utilização de recursos (em especial o tempo) ou à eficácia de um método algorítmico ou de certas características de um método, como procedimentos e decisões.

E por fim, o nome combina o prefixo grego "**meta**" ("mais além", aqui no sentido de "alto nível").

**IME-USP**

Baseando-se nessas definições, e ainda citando Sucupira [3], temos a definição de **metaheurística** como sendo um conjunto de conceitos que pode ser utilizado para definir métodos heurísticos aplicáveis a uma ampla gama de problemas diversos. Em outras palavras, uma metaheurística pode ser vista como uma estrutura algorítmica geral que pode ser empregada na resolução de diferentes problemas de otimização, com um número relativamente reduzido de modificações que a adaptem para o tratamento de cada problema específico.



IME-USP

2. Visão geral

Segundo Melián et al.[4], para resolver problemas de otimização utilizando heurísticas podemos usá-las de forma mais geral ou mais específica, dependendo do problema. Os métodos heurísticos específicos devem ser projetados para um problema específico, utilizando toda a informação disponível e a análise teórica do modelo. Procedimentos específicos bem projetados tendem a ter um rendimento significativamente mais alto dos procedimentos heurísticos gerais, que por outro lado, apresentam outros tipos de vantagens como simplicidade, adaptabilidade e robustez.

No entanto, as heurísticas gerais oriundas das metaheurísticas podem melhorar seu rendimento utilizando recursos computacionais e estratégias inteligentes. O conceito atual do que é uma metaheurística está baseado em diferentes interpretações do que é uma forma inteligente de resolver um problema.

As **metaheurísticas** são estratégias inteligentes para projetar ou melhorar procedimentos heurísticos mais gerais com alto rendimento.

O termo metaheurística apareceu pela primeira vez em um artigo sobre Busca Tabu de Fred Glover em 1986. A partir de então tem surgido múltiplas propostas para projetar bons procedimentos para resolver certos problemas que ampliam seu campo de aplicação, daí a denominação de **metaheurística**. A relevância das metaheurísticas seja em publicações de livros, artigos, ensaios, dissertações e teses nos últimos anos são variadas e vem aumentando, desde 1985.

Sucupira [3] salienta que muitas metaheurísticas diferem significativamente entre si, três características freqüentemente requerem atenção especial: as soluções inviáveis, a função objetivo e o critério de parada. Cada uma dessas questões pode ser tratada conforme descrito a seguir:

- O critério para lidar com as soluções inviáveis dificilmente é estipulado pelas metaheurísticas. Portanto, é tarefa do programador solucionar essa questão. A prática mais comum é a *relaxação* do problema, de maneira a tornar viáveis todas as soluções e fazer com que a função objetivo penalize intensamente as soluções outrora inviáveis. Porém, em diversas situações, pode ser interessante adotar alguma outra estratégia, como, por exemplo, impedir a manipulação de soluções inviáveis.
- A função objetivo é um problema relevante apenas nos casos em que avaliar uma solução é uma tarefa computacionalmente custosa. Nessas situações, proveniente da modelagem do problema normalmente se utiliza uma aproximação simplificada da função objetivo ou se estabelece uma estratégia para avaliar cada nova solução a partir, se for o caso, de uma ou mais soluções que tenham dado origem a ela.
- O critério de parada – estratégia que define o momento em que o algoritmo está concluído – pode ser definido com base em diversos fatores, especialmente nos casos – que constituem a grande maioria – em que a metaheurística não o define. Entre esses

**IME-USP**

fatores, estão o tempo consumido, o número de iterações realizadas, o número de iterações subseqüentes que não geraram uma solução superior à melhor solução já produzida, etc.

Podemos ainda definir algumas características gerais das metaheurísticas como [5]:

- ✓ São estratégias que guiam o processo de busca.
- ✓ O objetivo é explorar eficientemente o espaço de busca a fim de achar soluções ótimas ou quase ótimas.
- ✓ São algoritmos variando de um simples procedimento de busca local a complexos processos de aprendizado.
- ✓ São algoritmos aproximados e usualmente não determinísticos.
- ✓ Podem incorporar mecanismos para evitar ficar presos em áreas confinadas do espaço de busca.
- ✓ Não são específicas para um determinado problema.
- ✓ Podem usar um conhecimento específico do problema na forma de heurísticas que são controladas por uma estratégia de nível superior.



IME-USP

3. Propriedades desejáveis

Mélian et al [4] discorre sobre algumas propriedades desejáveis para as metaheurísticas. Propriedades desejáveis são aquelas que favorecem o interesse prático e teórico das metaheurísticas. Indicam as direções para onde os esforços devem ser seguidos para contribuir no desenvolvimento e construção de uma metaheurística. A seguir algumas das propriedades:

Simples: a metaheurística deve ser baseada em um princípio simples e claro, fácil de compreender.

Precisa: os passos e fases da metaheurística devem ser formulados em termos concretos.

Coerente: os elementos de uma metaheurística devem deduzir naturalmente seus princípios.

Efetiva: os algoritmos derivados de uma metaheurística devem proporcionar soluções de alta qualidade, ótimas ou muito próximas do ideal.

Eficaz: a probabilidade de alcançar soluções ótimas em casos realistas com a metaheurística deve ser alta.

Eficiente: deve realizar um bom aproveitamento dos recursos computacionais, como tempo de execução e uso de memória.

Geral: deve ser utilizada com bom rendimento em uma ampla variedade de problemas.

Adaptável: deve ser capaz de adaptar-se a diferentes contextos de aplicações ou modificações importantes do modelo.

Robusta: seu comportamento deve ser pouco sensível a pequenas variações do modelo no contexto da aplicação.

Interativa: deve permitir que o usuário possa aplicar seus conhecimentos para melhorar o rendimento do procedimento.

Múltipla: deve fornecer diferentes soluções alternativas de alta qualidade entre as que o usuário possa escolher.

Autônoma: deve permitir um funcionamento autônomo, livre de parâmetros ou que se possa estabelecê-los automaticamente.

Várias dessas propriedades estão inter-relacionadas e apontam para a mesma direção, como a simplicidade, a precisão e a coerência. A simplicidade de uma metaheurística facilita a sua utilização e contribui para dar-lhe ampla aplicabilidade.

**IME-USP**

A precisão na descrição dos elementos que compõem uma metaheurística é crucial para concretizar um procedimento de alta qualidade e fácil de implementar. Os passos dos procedimentos básicos dos algoritmos devem traduzir coerentemente os princípios que os inspiram. Deve-se fugir de sentenças sem sentido ou vagas. Frequentemente apresentam-se como extensões de uma metaheurística a incorporação de ferramentas ou recursos computacionais padrão, ou de outras metaheurísticas quando na verdade devem ser caracterizadas como hibridizações de si mesmas.

A evolução do rendimento de uma metaheurística deve atender tanto a eficiência quanto à efetividade e eficácia dos procedimentos heurísticos obtidos. Para validar a efetividade e eficácia de uma metaheurística, ela deve resolver com êxito problemas de um conjunto de casos reais. Caso esse conjunto de casos reais não existir, devem-se construir processos de simulação que se aproximem de tais casos. A eficiência do método se constata experimentalmente pelo emprego de um tempo computacional moderado (ou ao menos, razoável) para alcançar com êxito os problemas considerados.

O tamanho dos problemas considerados em aplicações práticas dos métodos de otimização se limitam pelas ferramentas disponíveis para resolvê-los mais do que pela necessidade dos potenciais usuários. Quando as metaheurísticas são aplicadas a instâncias realmente grandes, seus pontos fortes e fracos aparecem mais claramente. As metaheurísticas podem melhorar seu rendimento estendendo-se em várias direções e possivelmente, hibridizações. Os procedimentos heurísticos resultantes ficam complicados se utilizam muitos parâmetros. Isso pode melhorar sua eficiência, mas mascara as razões de seu êxito. Às vezes, a alta especialização de uma metaheurística leva a um ajuste fino de parâmetros sobre algum conjunto específico de problemas.

A aplicabilidade de uma metaheurística deve estar sustentada na generalidade, mas também em sua adaptabilidade e robustez. A robustez deve ser testada experimentalmente analisando o desempenho em relação às flutuações das características dos problemas. A robustez é refletida no número de parâmetros nas diversas aplicações, que deve permanecer baixo. A generalidade se reflete na diversidade dos campos de aplicações para os quais se utiliza com êxito, e a adaptabilidade permite que as conclusões obtidas ao defrontar com um tipo de problema particular possam ser aproveitadas em outros contextos.

Uma interface amigável deve ser perseguida para favorecer a utilidade de uma metaheurística. A interatividade dos sistemas baseados em metaheurística favorece a colaboração com outros campos que proporcionam conhecimentos específicos de problemas para melhorar o rendimento da metaheurística. A possibilidade de oferecer diversas soluções de alta qualidade, mas que sejam diferentes, entre as que podemos escolher, contribui para a disseminação de seu uso. A relativa autonomia de implementações permite ganhar a confiança dos usuários pouco experientes em otimização ou em outros campos de aplicação.

Uma característica que contribui para divulgar uma metaheurística é a novidade a que ela está associada, enquanto a originalidade dos princípios que a inspiraram e os campos de repercussão social que se aplicam. Este aspecto é revelado, por exemplo, na inspiração em fenômenos naturais de algoritmos genéticos, na aplicação de uma prova

**IME-USP**

matemática de uma metaheurística de entorno variável, ou na aplicação de técnicas de engenharia genética. No entanto, em ambientes científicos, tecnológicos, de engenharia ou de negócios, o aspecto mais relevante é o sucesso associado à eficiência e eficácia de cada algoritmo metaheurístico na resolução de problemas de grande tamanho, ou decorrentes de aplicações reais.



4. Classificação das Metaheurísticas

Quando o assunto refere-se à classificação das metaheurísticas não existe, ainda, um consenso entre os autores para uma classificação padrão. Na bibliografia existem algumas classificações que levam em consideração ora a função, ora o método ou outra característica da metaheurística. Assim sendo, serão mencionados nessa monografia três classificações distintas com o objetivo de ilustrar o assunto, sem contudo, concluí-lo.

4.1. Classificação de Mélian et al

Para Mélian et al [4], metaheurística é uma estratégia para projetar procedimentos heurísticos, e portanto, os tipos de metaheurística se estabelecem em função do tipo de procedimento ou método a que se referem. E esses tipos fundamentais dividem-se em: metaheurística de métodos de relaxação, metaheurística para processos construtivos, metaheurística de busca por entornos, metaheurística para métodos evolutivos e metaheurística híbrida.

Assim, uma definição de cada tipo de metaheurística se faz necessário:

a) Metaheurísticas de Métodos de Relaxação [6]

Em problemas cujo cálculo da função objetivo consome grande tempo de execução, torna-se importante uma modificação da função objetivo, de forma a diminuir o tempo consumido por este cálculo.

Um conjunto de restrições rígidas tende a limitar o espaço de busca do algoritmo. Desta forma, tornando as restrições mais fracas ou, em alguns casos, excluindo essas restrições é possível encontrar soluções ótimas viáveis em regiões do espaço de busca consideradas inalcançáveis pelas restrições originais.

Desta forma podemos descrever uma metaheurística de relaxação como sendo um método que, através de modificações na modelagem do problema original, geralmente alterando a função objetivo e o conjunto de restrições, cria um problema relaxado cuja solução é encontrada de forma eficaz. A solução do problema relaxado é usada como guia pelo algoritmo na busca da solução do problema original, que geralmente consiste na adequação da solução do problema relaxado as restrições do problema original.

b) Metaheurísticas para Processos Construtivos [6]

Metaheurísticas Construtivas consistem em estabelecer uma estratégia para a construção de soluções viáveis de forma gradativa. Assim, partindo de uma solução inicial vazia, a solução parcial deve ser incrementada gradativamente, de forma a não tornar-se inviável, até que o critério de parada seja atingido.

A estratégia gulosa é a mais clássica dessas estratégias, consistindo em, a cada interação, enquanto o critério de parada não for alcançado, adicionar à solução parcial



IME-USP

um novo elemento que propicie, de forma imediata, o melhor resultado possível, sem tornar a solução inviável.

c) Metaheurísticas de Busca por Entornos [6]

Uma solução parcial ou completa pode ser modificada de forma a gerar outra solução, sendo ambas viáveis ou não, por um método que no contexto das metaheurísticas de busca por entornos é denominado, operador. O conjunto de soluções que podem ser geradas, a partir, da aplicação de um operador a uma solução S é denominado vizinhança. Desta forma as metaheurísticas de busca por entornos podem ser definidas como algoritmos que percorrem espaços de buscas, formados por soluções, sendo que a cada interação é considerado a vizinhança obtida na interação anterior. Na Figura 1 pode-se observar a estrutura genérica os algoritmos de busca por entornos.

1. $S_0 = \text{GeraSoluçãoInicial}()$.
2. Seja S_{atual} a solução que apresenta o melhor valor para a função objetivo encontrada pelo algoritmo e S_n a solução corrente.
3. $S_{\text{atual}} = S_0$.
4. $S_n = S_0$.
5. Enquanto não for satisfeito o critério de parada faça:
 - 5.1. Aplique o(s) operador(es) à solução S_n , gerando a solução S_{n+1}
 - 5.2. Submeta S_{n+1} a função de avaliação.
 - 5.3. Se S_{n+1} apresenta melhor valor que S_{atual} .
 - 5.3.1. $S_{\text{atual}} = S_{n+1}$
 - 5.4. $S_n = S_{n+1}$

Figura 1 - Estrutura Genérica das Metaheurísticas de Busca por Entornos

Estes algoritmos necessitam de uma solução prévia, comumente chamada de solução inicial, que pode ser gerada utilizando-se diversas estratégias diferentes, dentre elas: aleatoriedade, que usa uma distribuição aleatória dos elementos do problema para gerar uma solução; trivialidade, que gera uma solução básica viável do problema; através de métodos construtivos, heurísticas que geram de forma incremental a solução.

Faz-se necessário a definição da quantidade e características dos operadores a serem aplicados para uma correta implementação das metaheurísticas de busca por entornos. Esta decisão exige experiência do programador a fim de evitar a limitação do espaço de busca e a desaceleração da evolução do algoritmo e suas soluções, causados, respectivamente, pelo reduzido número de operadores selecionados para aplicação no algoritmo e excessivo número de operadores aplicados a cada interação.

d) Metaheurísticas para Métodos Evolutivos [6]

No contexto das metaheurísticas evolutivas cada solução, viável ou não, é chamadas de indivíduo, e um conjunto de soluções denomina-se população. A cada interação, comumente denominada de geração, o algoritmo manipula a população escolhendo



IME-USP

indivíduos que reproduziram, gerando indivíduos novos que constituirão de forma parcial ou total a população da próxima geração.

A seleção dos indivíduos que irão reproduzir é um passo importante para a evolução do algoritmo, pois se os indivíduos mais aptos sempre forem selecionados, a diversidade da população será comprometida, assim como, algumas características favoráveis presentes nos indivíduos menos aptos poderão ser perdidas. Desta forma, as chances de um indivíduo ser selecionado para reproduzir dependerá da função de aptidão, ou *fitness*, que em geral consiste na função objetivo.

Outro importante passo para a evolução compreende a escolha e aplicação dos operadores, responsáveis por realizar a reprodução dos indivíduos selecionados, promovendo a combinação das características de dois ou mais indivíduos, e ou, simplesmente a alteração de alguma característica de indivíduos isoladamente, de forma a gerar novos indivíduos.

Segundo a teoria evolutiva Darwiniana, os organismos que melhor se adaptam às condições ambientais têm maiores chances de reproduzir, gerando descendentes férteis, de forma a perpetuar as suas características genéticas. Assim, as metaheurísticas evolutivas podem ser definidas como algoritmos baseados na teoria evolutiva Darwiniana, que buscam, a cada geração, promover a evolução da população através da reprodução de seus indivíduos, a fim de encontrar a solução ótima para o problema, como pode ser visualizado na Figura 2.

1. $S_0 = \text{GeraPopulaçãoInicial}()$.
2. Seja S_n a população atual, $S_n = S_0$.
3. Enquanto não for satisfeito o critério de parada faça:
 - 3.1. Utilizando um método de seleção, selecione indivíduos da população S_n que serão submetidos aos operadores.
 - 3.2. Aplicando os operadores aos indivíduos selecionados, gere novos indivíduos.
 - 3.3. Seguindo os critérios do algoritmo, utilize os novos indivíduos e a população S_n , para gerar a população S_{n+1} .

Figura 2 - Estrutura genérica das Metaheurísticas Evolutivas

e) Metaheurísticas Híbridas

Algumas metaheurísticas apresentam uma abordagem completamente diferente, não pertencendo a nenhuma das classes vistas anteriormente, ou utilizam mais de uma das abordagens descritas acima, sendo assim denominadas metaheurísticas híbridas.

4.2. Classificação de Sucupira

Sucupira [3] define que muitos aspectos podem ser considerados ao realizarmos classificações dos métodos metaheurísticos. Porém, observando as características mais



IME-USP

comumente analisadas, é possível notar a existência de duas grandes classes, que enquadram quase todas as metaheurísticas. Esse fato, aliado à inexistência de uma categorização padronizada, nos permite dividir esses métodos em **metaheurísticas de busca por entornos** e **metaheurísticas populacionais**.

a) Metaheurísticas de Busca por Entornos

Um *operador*, no contexto dos métodos de busca por entornos, pode ser definido como qualquer procedimento que modifica uma solução parcial ou completa – viável ou não – produzindo outra solução. O termo *vizinhança*, quando relacionado a uma solução S , refere-se ao conjunto das soluções que podem ser geradas através da aplicação de algum operador a S . Chamamos metaheurística de busca por entornos a qualquer método que percorra espaços de busca – compostos por soluções – levando em conta fundamentalmente, em cada passo, a vizinhança da solução obtida na iteração anterior.

Naturalmente, esses algoritmos requerem uma solução prévia. Essas *soluções iniciais* podem ser produzidas de diversas maneiras: aleatoriamente (por exemplo: uma partição aleatória dos elementos, em um problema de particionamento), trivialmente (como uma solução em que todos os veículos permanecem em suas posições iniciais, em um problema de roteamento), através de métodos construtivos (heurísticas que constroem uma solução de maneira incremental, a partir da seleção metódica de cada componente) ou utilizando quaisquer outras estratégias.

Para a definição e a utilização da *estrutura de entornos* (quantidade de operadores e características destes), questão raramente abordada pela própria metaheurística, há duas situações recorrentes a serem evitadas: a aplicação e avaliação de um número demasiadamente elevado de operadores em cada iteração, desacelerando a evolução das soluções; a restrição a um pequeno grupo definitivo de operadores aplicáveis, limitando excessivamente o espaço de busca. Com isso, a implementação adequada da estrutura de entornos freqüentemente é consequência da experiência do programador e de avaliações experimentais, podendo a carência em um desses fatores ser compensada pela alta disponibilidade do outro.

Em sua concepção primária, as buscas por entornos são *monótonas*, ou seja, nenhuma de suas iterações tem como resultado a atualização da solução corrente de maneira que a nova solução não tenha custo estritamente menor que o da anterior.

Porém, muitas vezes a estrutura de entornos não permite a análise completa da vizinhança em cada iteração. Nesses casos, ainda é aconselhável selecionar, em cada passo, uma “boa” solução da vizinhança corrente.

b) Metaheurísticas Populacionais

As metaheurísticas populacionais lidam com um conjunto de soluções, muitas vezes chamado de *população*, que evolui, através da interação entre seus elementos e de processos individuais, procurando enfatizar as características desejáveis das soluções em toda a população, num processo que objetiva aumentar a qualidade média sem comprometer a diversidade do conjunto.



IME-USP

4.3. Classificação de Laguna

Em seu artigo “*Global Optimization and Meta-heuristics*”, Laguna [7] cita que a evolução das metaheurísticas durante os últimos dez anos foi expressiva. Metaheurísticas em sua forma moderna são baseadas em uma variedade de interpretações do que constitui uma busca “inteligente”. Essas interpretações lidam com escolhas de projetos que podem ser utilizados como mecanismos para sua classificação.

Entretanto, uma rigorosa classificação de diferentes metaheurísticas é uma questão difícil e arriscada porque os principais defensores de métodos alternativos, muitas vezes diferem entre si sobre a natureza essencial dos métodos que eles defendem. Isto pode ser ilustrado considerando-se a classificação das metaheurísticas em termos das suas características com relação a três escolhas básicas:

- 1) Uso de memória adaptativa;
- 2) Tipo de exploração de vizinhança utilizada, e
- 3) Número de soluções correntes utilizadas na próxima iteração.

Essas opções podem ser incorporadas em um esquema de classificação da forma $x/y/z$, onde as escolhas para x são A (se a metaheurística emprega memória adaptativa) e M (se o método é "sem memória"). As escolhas para y são N (para um método que emprega alguma pesquisa sistemática de vizinhança ou para selecionar o próximo movimento ou para melhorar uma determinada solução) e S (para os métodos baseando-se numa amostragem aleatória). Finalmente, z pode ser 1 (se o método move de uma solução atual para a abordagem seguinte, após cada iteração) ou P (para uma base populacional, com uma população de tamanho P). Este simples esquema tridimensional nos dá uma base preliminar de classificação, revela que a forma de colocar etiquetas em várias metaheurísticas está longe de ser uniforme.



5. Levantamento das Metaheurísticas

De acordo com os três tipos de classificação descritos na seção anterior, será mostrado um levantamento de algumas metaheurísticas, a seguir.

5.1. Classificação de Mélian et al

- a) Metaheurística de métodos de relaxação: são procedimentos de resolução de problemas que utilizam flexibilizações do modelo original (ou seja, modelo com modificações que tornam o problema mais fácil de resolver), cuja solução fornece a solução para o problema original.

Exemplo: [8]

- ✓ **Relaxação Lagrangeana**: remove algumas restrições de um problema de programação linear, atribui um peso (multiplicador de Lagrange) a cada uma delas e altera a função objetivo para penalizar as soluções que seriam inviáveis no problema original.

- b) Metaheurística para processos construtivos: baseia-se em procedimentos que tratam da obtenção de uma solução a partir da análise e seleção paulatina dos componentes que a formam.

Exemplos: [8]

- ✓ **GRASP (Greedy Randomized Adaptive Search Procedure)**: cada iteração é composta por uma fase construtiva e uma fase de busca por entornos. Em cada passo da fase construtiva:
 - Selecionam-se os componentes que causam melhor efeito se adicionados à solução atual.
 - Acrescenta-se um desses elementos (selecionado aleatoriamente) à solução.

- c) Metaheurística de busca por entornos: chamamos dessa forma para qualquer método que percorra espaços de busca – compostos por soluções – levando em conta fundamentalmente, em cada passo, a vizinhança da solução obtida na iteração anterior.

Exemplos: [8]

- ✓ **GLS (Guided Local Search)**: busca monotônica. Porém, altera a função objetivo ao encontrar um ótimo local.
- ✓ **Simulated Annealing**: não monotônica. Probabilidade de “movimentos ruins” decresce exponencialmente com o aumento na diferença de custo.
- ✓ **Busca Tabu**: não monotônica. Classifica como *tabu* os componentes de soluções adicionados ou removidos recentemente.
- ✓ **Busca Reativa**: Busca Tabu com detecção de ciclos.

- d) Metaheurística para métodos evolutivos: enfocam os métodos baseados em conjuntos de soluções que evoluem sobre o espaço de soluções.

Exemplos: [8]

- ✓ **Algoritmos genéticos**.



IME-USP

- ✓ **Algoritmos meméticos:** algoritmos genéticos que realizam otimização local.
- ✓ **Estimação de distribuição:** verificam a distribuição dos componentes nas melhores soluções e criam a geração seguinte aleatoriamente, segundo essa distribuição.
- ✓ **Busca dispersa e *Path relinking*** : criam caminhos entre soluções e geram a população seguinte a partir das soluções que aparecem nesses caminhos.

e) Metaheurística híbrida: são metaheurísticas intermediárias em relação aos quatro tipos anteriores.

Exemplos:[8]

- ✓ **Meta-heurísticas de decomposição:** indicam como decompor o problema em instâncias e utilizar as soluções resultantes na construção da solução para o problema original.
- ✓ ***Ant colony optimization*** : formigas artificiais percorrem um grafo em que cada aresta representa um componente de solução.
- ✓ **Otimização extrema:** em cada iteração, remove o pior componente.
- ✓ ***Particle swarm optimization***.
- ✓ ***Iterated local search*** : tenta realizar uma espécie de busca por entornos nos ótimos locais.

5.2. Classificação de Sucupira

a) Metaheurísticas de Busca por Entornos:

Exemplos:

- ✓ ***Hill-Climbing*** cujo critério de parada consiste em finalizar o algoritmo quando se encontra uma solução S_f cuja vizinhança não oferece possibilidades de aumento imediato de qualidade. Nessa situação, dizemos que foi encontrado um *ótimo local*: a solução S_f . Chamamos de *busca local* qualquer procedimento que tenha como objetivo encontrar ótimos locais.
- ✓ **Busca por Entornos Variáveis (*Variable Neighbourhood Search*)** emprega um conjunto de estruturas de entornos para realizar alterações sistemáticas na definição de vizinhança. Mesmo em sua forma mais simples, essa metaheurística possui diferentes versões.
 - ✓ A ***Busca Gulosa por Entornos Variáveis (Variable Neighbourhood Descent)***, por exemplo, emprega uma diferente estrutura de entornos em cada iteração, selecionando sempre a melhor solução da vizinhança corrente. O método retorna à primeira estrutura sempre que uma iteração produz diminuição de custo e avança para a próxima estrutura sempre que isso não ocorre – caso não haja mais estruturas, considera-se a execução concluída.
 - ✓ A ***BEV Reduzida (Reduced VNS)*** difere da anterior por selecionar uma solução aleatória – e não a melhor – da vizinhança em cada iteração, privilegiando a eficiência. Nesse caso, o processo como um todo pode ser repetido diversas vezes, de acordo com alguma condição de parada.



IME-USP

- ✓ A *BEV Básica (Basic VNS)* e a *BEV Geral (General VNS)* combinam características das duas variantes anteriores.
- ✓ **Busca Local Guiada** (*Guided Local Search*) é um método de alta qualidade cuja principal estratégia é penalizar, através de alterações na função objetivo, os elementos (de soluções) que aparecem com frequência em ótimos locais.
- ✓ **Recozimento Simulado** (*Simulated Annealing*) é um método clássico, amplamente estudado e utilizado, capaz de obter resultados de grande qualidade quando executado por um período suficientemente longo.
- ✓ **Busca Tabu** é uma meta-heurística de busca por entornos muito popular, cuja principal característica é a capacidade de exploração do histórico do processo de busca, organizado em estruturas que compõem o que se chama de *memória adaptativa*.
- ✓ **Busca Reativa** é uma extensão da busca tabu, especialmente no que diz respeito à diversificação.
- ✓ **GRASP** (*Greedy Randomized Adaptive Search Procedures*) é um método formado pela reiteração de um processo de duas fases: construção gradual inteligente e busca por entornos operando sobre soluções completas.
- ✓ **Busca Local Iterada** (*Iterated Local Search*) opera uma heurística de busca por entornos H , definida no momento da implementação.

b) Metaheurísticas Populacionais:

Alguns exemplos de metaheurísticas evolutivas são:

- ✓ **Algoritmos Genéticos,**
- ✓ **Algoritmos Meméticos,**
- ✓ **Algoritmos de Estimação de Distribuição,**
- ✓ **Busca Dispersa,**
- ✓ **Reconexão de Caminhos,**
- ✓ **Otimização com Formigas Artificiais** e o,
- ✓ Método *Particle Swarm Optimization*.

Os **algoritmos genéticos** são os métodos mais populares dentre as metaheurísticas evolutivas e já foram aplicados a um grande número de problemas de otimização, nos mais diversos contextos. Esses algoritmos se inspiram nos processos de seleção natural que ocorrem em populações de seres vivos, característica que trouxe termos como “geração”, “gene” e “aptidão” para o contexto dos métodos heurísticos.

Os **algoritmos meméticos** são métodos heurísticos que combinam características de metaheurísticas evolutivas com processos de busca por entornos. Embora esses métodos se assemelhem fortemente aos algoritmos genéticos, suas particularidades permitem classificá-los como constituintes de uma metaheurística distinta.



IME-USP

5.3. Classificação de Laguna

Conforme a definição mostrada na seção 4.3, um exemplo de classificação usando os critérios definidos por Manuel Laguna, podemos citar os exemplos mostrados na tabela abaixo:

Metaheurística	Classificação 1	Classificação 2
Algoritmos genéticos	M/S/P	M/N/P
Busca dispersa	M/N/P	A/N/P
Recozimento simulado	M/S/1	M/N/1
Busca tabu	A/N/1	A/N/P

Tabela 1. Classificação das Metaheurísticas.

Duas classificações são mostradas para cada procedimento. A primeira classificação mais se aproxima da concepção “popular” e a segunda é defendida por um significativo grupo de pesquisadores. As diferenças entre estas classificações ocorrem por motivos diferentes, dependendo do método. Algumas diferenças estão presentes desde o momento em que os métodos foram propostos, enquanto outros representam mudanças recentes que foram introduzidas por um subgrupo de defensores ardorosos. Por exemplo, a forma original do recozimento simulado chegou a ser modificada por um grupo que considera elementos mais fortes de pesquisa de vizinhança que devem ser incorporados. Uma mudança semelhante aconteceu com algoritmos genéticos em meados dos anos 1980. Ainda assim, deve salientar-se que nem todos os defensores do recozimento simulado e algoritmos genéticos vêem essas mudanças como apropriadas.

Por outro lado, entre os exemplos em diferentes classificações estavam presentes desde o início, os documentos originais da busca tabu incluíam elementos de base populacional, na forma de estratégias para a exploração de conjuntos de soluções salvos durante a pesquisa. No entanto, uma parte significativa da literatura não usava população, usavam características de busca tabu até recentemente. Da mesma forma, a busca de dispersão foi acompanhada de elementos de memória adaptativa, como resultado de ser associada às idéias de busca tabu anteriormente.

Os poucos defensores do recozimento simulado e algoritmos genéticos recentemente tem ido mais longe a modificar as concepções originais do que o indicado na Tabela 1, e propõe a inclusão de elementos de memória adaptativa tal como consagrado na busca tabu. Essas propostas são freqüentemente descritas pelos seus autores como métodos híbridos, devido ao seu casamento de aspectos de diferentes *frameworks*.



IME-USP

6. Histórico

Um estudo abrangente sobre metaheurísticas não seria completo sem mostrar seu histórico desde o surgimento até artigo e/ou pesquisa mais recentes. Após várias pesquisas na Internet em busca de um histórico, foi encontrado material na Wikipedia [9], em inglês, muito interessante que será mostrado a seguir.

Os itens ao final de cada ano em sobrescrito são as referências do histórico e encontra-se em detalhes ao final desse capítulo em 6.1.

- ✓ 1952: Primeiros trabalhos com **métodos estocásticos de otimização**.^[1]
- ✓ 1954: Barricelli realiza as primeiras simulações do processo de evolução e as utiliza em problemas de otimização geral.^[2]
- ✓ 1965: Rechenberg cria o primeiro algoritmo usando estratégias de evolução.^[3]
- ✓ 1966: Fogel, Owens e Walsh propõe a **programação evolucionária**.^[4]
- ✓ 1970: Hastings concebe o algoritmo *Metropolis-Hastings*, que pode provar qualquer função densidade de probabilidade.^[5]
- ✓ 1975: John Holland propõe o primeiro **algoritmo genético**.^[6]
- ✓ 1980: Smith descreve a programação genética^[7]
- ✓ 1983: Baseado no trabalho de Hastings, Kirkpatrick, Gelatt e Vecchi concebem o **recozimento simulado**.^[8]
- ✓ 1985: Independentemente, Černý propõe o mesmo algoritmo.^[9]
- ✓ 1986: Primeira menção do termo **metaherurística** por Fred Glover, durante a concepção do Busca Tabu.^[10] :
 - *La recherche avec tabou peut être vue comme une "méta-heuristique", superposée à une autre heuristique.*
 - ("A busca com tabu pode ser vista como uma **meta-heurística**, sobrepondo outra heurística.")
- ✓ 1986: Farmer, Packard e Perelson trabalham com sistemas de imunidade artificiais.^[11]
- ✓ 1988: A primeira conferência sobre algoritmos genéticos é organizada pela Universidade de Illinois em Urbana-Champaign.
- ✓ 1988: Trabalhos com comportamento coletivo de alguns tipos de formigas em aplicações de Inteligência Artificial.^[12]
- ✓ 1988: Koza registra sua primeira patente de programação genética.^[13]
- ✓ 1989: Goldberg publica um dos mais conhecidos livros sobre algoritmos genéticos.^[14]
- ✓ 1989: Evolver, o primeiro software de otimização usando algoritmo genético, é liberado pela companhia Axcelis.
- ✓ 1989: O termo "**algoritmo memético**" é usado primeiramente por Moscato^[15]
- ✓ 1991: Os algoritmos de colônia de formigas são propostos por Marco Dorigo, em sua tese de doutorado.^[16]
- ✓ 1993: O jornal *Evolutionary Computation* inicia sua publicação no MIT.
- ✓ 1995: Feo e Resende propõe o **procedimento de busca adaptativa aleatória gulosa - GRASP** (*Greedy Randomized Adaptive Search Procedure*).^[17]
- ✓ 1995: Kennedy e Eberhart concebem o **Particle Swarm Optimization**^{[18][19]}
- ✓ 1996: Mühlenbein e Paaß trabalham com algoritmos de distribuição estimados.^[20]
- ✓ 1997: Storn e Price propõe um algoritmo de evolução diferencial.^[21]



IME-USP

- ✓ 1997: Rubinstein trabalha com método de entropia cruzada.^[22]
- ✓ 1999: Boettcher e Percus propõe uma otimização extrema.^[23]
- ✓ 2000: Primeiro algoritmo genético interativo.^[24]
- ✓ 2001: Geem, Kim e Longanathan propõe a **Busca Harmonia**.^[25]
- ✓ 2004: Nakrani e Tovey descrevem o **algoritmo da colméia** (*honey bee algorithm*).^[26]
- ✓ 2005: Krishnanand e Ghose propõe o *Glowworm Swarm Optimization* (GSO)^[27]^[28] para ótima captação simultânea de múltiplas funções multimodais.
- ✓ 2005: Karaboga propõe o **Algoritmo Colônia de Abelhas Artificiais** (*Artificial Bee Colony Algorithm*).^[29]
- ✓ 2005: Pham D.T et al. propõem uma novela algorítmica denominada de algoritmo das abelhas.^[30] ^[31]
- ✓ 2008: Yang descreve o **algoritmo vaga-lume** (*firefly algorithm*).^[32]

6.1. Referências Bibliográficas do Histórico

1. Robbins, H. and Monro, S., *A Stochastic Approximation Method*, Annals of Mathematical Statistics, vol. 22, pp. 400-407, 1951
2. Barricelli, Nils Aall, *Esempi numerici di processi di evoluzione*, Methodos, pp. 45-68, 1954
3. Rechenberg, I., *Cybernetic Solution Path of an Experimental Problem*, Royal Aircraft Establishment Library Translation, 1965
4. Fogel, L., Owens, A.J., Walsh, M.J., *Artificial Intelligence through Simulated Evolution*, Wiley, 1966
5. W.K. Hastings. *Monte Carlo Sampling Methods Using Markov Chains and Their Applications*, Biometrika, volume 57, issue 1, pages 97-109, 1970
6. Holland, John H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975
7. Smith, S.F., *A Learning System Based on Genetic Adaptive Algorithms*, PhD dissertation (University of Pittsburgh), 1980
8. S. Kirkpatrick, C. D. Gelatt et M. P. Vecchi, *Optimization by Simulated Annealing*, Science, volume 220, issue 4598, pages 671-680, 1983
9. V. Černý *A thermodynamical approach to the travelling salesman problem : an efficient simulation algorithm* Journal of Optimization Theory and Applications, volume 45, pages 41-51, 1985
10. Fred Glover, *Future Paths for Integer Programming and Links to Artificial Intelligence*, Comput. & Ops. Res. Vol. 13, No.5, pp. 533-549, 1986
11. J.D. Farmer, N. Packard and A. Perelson, *The immune system, adaptation and machine learning*, Physica D, vol. 22, pp. 187--204, 1986
12. F. Moyson, B. Manderick, *The collective behaviour of Ants : an Example of Self-Organization in Massive Parallelism*, Actes de AAI Spring Symposium on Parallel Models of Intelligence, Stanford, California, 1988
13. Koza, John R. *Non-Linear Genetic Algorithms for Solving Problems*. United States Patent 4,935,877. Filed May 20, 1988. Issued June 19, 1990
14. Goldberg, David E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Kluwer Academic Publishers, Boston, MA., 1989
15. P. Moscato, *On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts : Towards Memetic Algorithms*, Caltech Concurrent Computation Program, C3P Report 826, 1989.



IME-USP

16. M. Dorigo, *Optimization, Learning and Natural Algorithms*, Ph.D. Thesis, Politecnico di Milano, Italy, 1992.
17. Feo, T., Resende, M., *Greedy randomized adaptive search procedure*, Journal of Global Optimization, tome 42, page 32--37, 1992
18. Eberhart, R. C. et Kennedy, J., *A new optimizer using particle swarm theory*, Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan. pp. 39-43, 1995
19. Kennedy, J. et Eberhart, R. C., *Particle swarm optimization*, Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ. pp. 1942-1948, 1995
20. Müllhenbein, H., Paaß, G., *From recombination of genes to the estimation of distribution I. Binary parameters*, Lectures Notes in Computer Science 1411: Parallel Problem Solving from Nature, tome PPSN IV, pages 178--187, 1996
21. Rainer Storn, Kenneth Price, *Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces*, Journal of Global Optimization, volume 11, issue 4, pages 341-359, 1997
22. Rubinstein, R.Y., *Optimization of Computer simulation Models with Rare Events*, European Journal of Operations Research, 99, 89-112, 1997
23. Stefan Boettcher, Allon G. Percus, "Extremal Optimization : Methods derived from Co-Evolution", Proceedings of the Genetic and Evolutionary Computation Conference (1999)
24. Takagi, H., *Active user intervention in an EC Search*, Proceedings of the JCIS 2000
25. Geem Z. W., Kim J. H., and Loganathan G. V., A new heuristic optimization algorithm: harmony search, Simulation, vol. 76, 60 (2001)
26. Nakrani S. and Tovey S., On honey bees and dynamic server allocation in Internet hosting centers, Adaptive Behaviour, vol. 12, 223 (2004)
27. K.N. Krishnanand and D. Ghose., Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions, *Swarm Intelligence*, vol. 3(2), 87 - 124, (2009)
28. K.N. Krishnanand and D. Ghose., "Detection of multiple source locations using a glowworm metaphor with applications to collective robotics", *Proceedings of the IEEE Swarm Intelligence Symposium*, 84-91, 2005
29. D. Karaboga, AN IDEA BASED ON HONEY BEE SWARM FOR NUMERICAL OPTIMIZATION, TECHNICAL REPORT-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
30. D. T. Pham et al., "The bees algorithm—a novel tool for complex optimisation problems," in Intelligent production machines and systems: 2nd I* PROMS Virtual Conference, 3-14 July 2006, 2006, 454.
31. Pham DT, Ghanbarzadeh A, Koc E, Otri S, Rahim S and Zaidi M. The Bees Algorithm. Technical Note, Manufacturing Engineering Centre, Cardiff University, UK, 2005
32. Yang X. S., Firefly algorithm (chapter 8) in: Nature-inspired Metaheuristic Algorithms, Luniver Press, (2008)



IME-USP

7. Conclusão

Podemos dizer que **metaheurísticas** são métodos ou ferramentas algorítmicas de alto nível para explorar espaços de busca, aplicadas a diferentes problemas de otimização na área de Ciência da Computação com modificações relativamente pequenas, para torná-la adaptável a um problema específico.

Existem vários autores que definem e defendem classificações ou divisões, ou mesmo tipos de metaheurísticas, nesse trabalho foram citados três deles, contudo o objetivo foi de expor algumas dessas idéias, não fechar o assunto. Bem como os exemplos que cada um dos três autores citou para cada tipo mencionado.

E por fim, o histórico que se iniciou em 1951 com um artigo de Robbins e Monro sobre estudo de métodos estocásticos em otimização combinatória e termina com o algoritmo vaga-lume em 2008 de Yang.

Concluindo a monografia, gostaria de deixar meu parecer sobre a mesma. Foi instigante fazer esse estudo, que devido ao tempo não foi muito aprofundado, mas acredito ter atingido seu objetivo que foi ter um conhecimento mais amplo sobre metaheurística.



IME-USP

8. Bibliografia

- [1] Wikipedia:
http://pt.wikipedia.org/wiki/Otimiza%C3%A7%C3%A3o_combinat%C3%B3ria,
acessado em 14/12/2009.
- [2] BALIEIRO, Inocêncio Fernandes, Tese de Doutorado: **Arquimedes, Pappus, Descartes e Polya - Quatro Episódios na História da Heurística**. Rio Claro: IGCE . Cp. de Rio Claro-UNESP, 2004.
- [3] SUCUPIRA, Igor Ribeiro, Dissertação de Mestrado: **Um Estudo Empírico de Hiper-Heurísticas**. IME/USP, 2007.
- [4] B. MELIÁN, J. PÉREZ, e J. VEGA. **Metaheuristics: A Global View**. *Revista Iberoamericana de Inteligencia Artificial (Asociación Española de Inteligencia Artificial)*, 2(19), 2003.
- [5] BECCENERI, José Carlos. **Meta-heurísticas bio-inspiradas aplicadas a problemas de otimização**, apresentação em PowerPoint®, LAC/CNPq/INPE, acessado em 14/12/2009.
- [6] NETO, Dimitrius Fraga, Monografia de Conclusão de Curso: **Algoritmos de Estimação de Distribuição Aplicados ao Problema do Roteamento de Veículos**. Salvador: UFBA, 2008.
- [7] LAGUNA, Manuel. **Global Optimization and Meta-heuristics**, College of Business, University of Colorado at Boulder, USA.
- [8] SUCUPIRA, Igor Ribeiro, **Apresentação: Métodos Heurísticos Genéricos: Meta-heurísticas e Hiper-heurísticas**. IME/USP, 2007.
- [9] Wikipedia: <http://en.wikipedia.org/wiki/Metaheuristic>, acessado em 31/10/2009.