

Lista 01 - Metaheurística

Carlos Miguel

1. (a) No campo da Otimização sempre queremos encontrar qual a melhor solução para o nosso problema. "Melhor" pode ter diversos sentidos e isso vai depender do problema, por exemplo desde minimizar custo ou maximizar envio de materiais e na literatura existem vários métodos para encontrar a melhor solução. Contudo com o crescimento do problema isso pode gerar problemas tanto na questão de tempo computacional como na garantia de existir uma melhor solução. Nesse sentido uma *heurística* é um procedimento na qual não estamos mais atrás da melhor solução, mas sim de alguma solução que seja viável em problemas onde achar a melhor solução é custoso.
- (b) Historicamente as heurísticas foram sendo desenvolvidas para tipos específicos de problemas, contudo surgiu uma necessidade e uma convergência entre a Computação Evolutiva e a Pesquisa Operacional para criar heurísticas aplicadas a vários problemas, essas são chamadas de *meta-heurísticas*.
- (c) Seja a função $f(x)$ definida sobre um domínio D , um *ótimo global* é o ponto x_0 ou na qual $f(x) \leq f(x_0)$ se x_0 for máximo global e $f(x) \geq f(x_0)$ se for mínimo global. Dada a mesma função um *ótimo local* é o ponto x_1 na qual existe um valor $\epsilon > 0$ tal que $f(x \pm \epsilon) \leq f(x_1)$ ou $f(x \pm \epsilon) \geq f(x_1)$.
- (d) Em metaheurísticas, o termo limite inferior (lower bound) refere-se ao valor mínimo possível de uma solução para um determinado problema de otimização, enquanto o limite superior (upper bound) é o valor máximo possível. Esses limites são importantes para avaliar o desempenho das metaheurísticas, pois permitem avaliar se uma solução encontrada é factível e, se sim, se é próxima o suficiente do valor ótimo.
- (e) Um algoritmo construtivo é um tipo de algoritmo que constrói uma solução passo a passo, adicionando componentes ou elementos um de cada vez, até que a solução final seja alcançada.
- (f) Quando definimos mínimos locais, precisamos de uma vizinhança ϵ do valor x_1 , ou seja um conjunto de valores que distanciam até ϵ do valor x_1 .
- (g) Busca local é uma técnica de otimização que envolve a busca por soluções ótimas em um espaço de busca local, ou seja, uma região re-

restrita do espaço de soluções viáveis. Ao contrário de outras técnicas de otimização, como a busca global, que exploram o espaço de soluções inteiro em busca da melhor solução, a busca local começa com uma solução inicial e faz pequenas alterações incrementais nessa solução, avaliando se essas alterações melhoram ou não a qualidade da solução.

2. Em questão de Metaheurísticas podemos fazer três coisas em relação as restrições: 1. Penalização de restrições: Nessa abordagem, as restrições são incorporadas diretamente à função objetivo, através da adição de termos de penalização quando uma restrição é violada; 2. Geração de soluções viáveis: Nessa abordagem, a busca é restrita a soluções viáveis, ou seja, que atendem a todas as restrições do problema; 3. Remoção de restrições.
3. (a) Algoritmos Genéticos: É uma técnica de otimização baseada na evolução biológica, que utiliza conceitos como seleção natural, mutação e recombinação para gerar uma população de soluções candidatas que são avaliadas por uma função de aptidão. A partir disso, as soluções são selecionadas e reproduzidas para gerar novas soluções. Esse processo é repetido até que uma solução satisfatória seja encontrada.
- (b) Simulated Annealing: É um algoritmo de otimização que utiliza um processo estocástico para buscar uma solução ótima. O método é baseado no conceito físico de tratamento térmico de metais. O algoritmo começa com uma solução inicial e a partir dela são geradas soluções vizinhas. A cada iteração, a solução atual é substituída pela solução vizinha com probabilidade determinada por uma função de temperatura. Conforme o algoritmo avança, a temperatura é reduzida, diminuindo a probabilidade de se aceitar soluções piores.
- (c) k-opt: É um algoritmo que faz parte da família de algoritmos de busca local. Ele consiste em aplicar diversas operações de remoção e inserção de arestas em um caminho ou ciclo Hamiltoniano para tentar melhorar a solução. O k-opt se refere ao número de arestas que são removidas e substituídas durante o processo. Quanto maior o valor de k, mais intensa é a busca, mas também mais computacionalmente custosa.
- (d) Busca Tabu: É uma técnica de busca local que utiliza uma lista tabu para evitar que o algoritmo explore soluções já visitadas. Essa lista é atualizada a cada iteração do algoritmo e armazena informações sobre as soluções visitadas recentemente. A ideia é que o algoritmo não visite essas soluções novamente, a menos que não haja outras opções disponíveis. Além disso, o algoritmo pode utilizar estratégias de diversificação para tentar escapar de mínimos locais.
- (e) GRASP: É um algoritmo de busca construtiva que consiste em construir soluções candidatas de forma aleatória e refiná-las através de uma busca local. A ideia é que o algoritmo construa soluções iniciais que sejam viáveis, mas não necessariamente ótimas. A partir daí, é

realizada uma busca local para melhorar a solução. Esse processo é repetido várias vezes para tentar encontrar a melhor solução possível.

- (f) Particle Swarm Optimization - PSO: É um algoritmo de otimização baseado em inteligência de enxame, inspirado no comportamento dos pássaros e insetos que se movem em grupo. O algoritmo utiliza uma população de partículas que representam soluções candidatas. Cada partícula é atualizada de acordo com sua posição atual e a melhor posição encontrada pela população. O objetivo é que as partículas convergam para a melhor solução possível.
- (g) Colônia de formigas: É uma técnica heurística que se inspira no comportamento das formigas para encontrar soluções ótimas em problemas de otimização. As formigas se movem aleatoriamente pelo ambiente e depositam feromônios para marcar caminhos. Esses feromônios atraem outras formigas para seguir o mesmo caminho e, assim, reforçam a trilha. O algoritmo de colônia de formigas utiliza essa ideia para construir soluções, atribuindo feromônios às soluções e aprimorando a busca por meio da seleção e atualização de soluções.
- (g) Scatter search: É uma técnica heurística que busca soluções de alta qualidade, explorando e combinando soluções parciais. O algoritmo é composto por duas fases principais: diversificação e intensificação. Na fase de diversificação, um conjunto de soluções é gerado de forma aleatória. Na fase de intensificação, as soluções são combinadas para gerar novas soluções de melhor qualidade. Essas fases são executadas alternadamente até que uma solução satisfatória seja encontrada.
- (h) Path relinking: É uma técnica heurística que busca soluções de alta qualidade, combinando duas soluções diferentes para gerar uma nova solução intermediária. O algoritmo começa com duas soluções distintas e constrói um caminho entre elas. Em seguida, busca-se uma nova solução intermediária ao longo desse caminho, combinando características das duas soluções originais. Esse processo é repetido até que uma solução satisfatória seja encontrada.
- (i) Evolução diferencial: É uma técnica heurística baseada em população que busca soluções de alta qualidade, explorando e combinando soluções parciais. O algoritmo utiliza a ideia de mutação e recombinação genética para gerar novas soluções. As soluções são avaliadas por meio de uma função de aptidão e as melhores soluções são mantidas na população. A cada iteração, novas soluções são geradas a partir das melhores soluções da população atual.
- (j) Variable neighborhood search - VNS: É uma técnica heurística que busca soluções de alta qualidade, explorando diferentes vizinhanças em torno de uma solução. O algoritmo utiliza a ideia de perturbação, onde uma solução é perturbada aleatoriamente para gerar uma nova solução candidata. Em seguida, busca-se uma solução melhor na vizinhança dessa nova solução, utilizando uma busca local. O processo é

repetido para diferentes vizinhanças, até que uma solução satisfatória seja encontrada.

- (k) Relaxação lagrangeana: É uma técnica heurística que busca soluções para problemas de otimização com restrições. O algoritmo consiste em relaxar as restrições do problema e adicionar termos de penalidade à função objetivo. Em seguida, utiliza-se a técnica de otimização sem restrições para encontrar uma solução para o problema relaxado. A solução encontrada é avaliada em relação às restrições originais e ajustada por meio de uma técnica de atualização de multiplicadores de Lagrange.
 - (l) Relax-and-Fix: o método Relax-and-Fix é uma técnica de decomposição que lida com problemas de otimização combinatória com restrições. Ele é usado para resolver problemas onde algumas variáveis têm que ser fixas, ou seja, assumir um valor específico e não podem ser alteradas. O Relax-and-Fix consiste em relaxar a formulação do problema original de forma que as variáveis que precisam ser fixas sejam tratadas como constantes e, em seguida, resolver a formulação relaxada usando um algoritmo de otimização. Depois de obter uma solução ótima para a formulação relaxada, as variáveis fixas são substituídas pelos valores correspondentes, e o processo é repetido até que uma solução viável para o problema original seja encontrada.
 - (m) Fix-and-Optimize: o método Fix-and-Optimize é uma técnica de decomposição usada para resolver problemas de otimização combinatória com restrições. Ele é usado quando o problema original é muito grande e complexo, tornando impossível resolvê-lo em um tempo razoável. O método Fix-and-Optimize divide o problema em subproblemas menores e mais fáceis de resolver. Em seguida, um algoritmo de otimização é usado para resolver cada subproblema, enquanto as restrições do problema original são mantidas fixas. Depois de obter uma solução ótima para cada subproblema, essas soluções são combinadas para formar uma solução viável para o problema original. O processo é repetido até que uma solução ótima seja encontrada. O método Fix-and-Optimize é uma abordagem útil para resolver problemas de otimização combinatória grandes e complexos que não podem ser resolvidos diretamente.
4. No que tange o assunto de grafos e redes existem vários trabalhos sobre. O recente trabalho de [1] ele aplica meta-heurísticas na detecção de comunidades. No caso de [3] utiliza-se Algoritmos Genéticos para identificar quais os melhores sítios a serem removidos de uma infecção dado os seus auto-valores da matriz de Adjacência. Por fim [2] usa Local Search para colocar novos sítios em uma rede minimizando a betweenness centrality.

References

- [1] Bara'a A. Attea et al. "A review of heuristics and metaheuristics for community detection in complex networks: Current usage, emerging development and future directions". In: *Swarm and Evolutionary Computation* 63 (June 2021), p. 100885. DOI: 10.1016/j.swevo.2021.100885. URL: <https://doi.org/10.1016/j.swevo.2021.100885>.
- [2] Manuel Lozano and Humberto M. Trujillo. "Optimizing node infiltrations in complex networks by a local search based heuristic". In: *Computers & Operations Research* 111 (Nov. 2019), pp. 197–213. DOI: 10.1016/j.cor.2019.06.011. URL: <https://doi.org/10.1016/j.cor.2019.06.011>.
- [3] Asep Maulana, Marios Kefalas, and Michael T. M. Emmerich. "Immunization of networks using genetic algorithms and multiobjective metaheuristics". In: *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, Nov. 2017. DOI: 10.1109/ssci.2017.8285368. URL: <https://doi.org/10.1109/ssci.2017.8285368>.