

ESTUDO COMPARATIVO DAS METAHEURÍSTICAS: ALGORITMO GENÉTICO(AG), BUSCA TABU(TS), COLÔNIA DE FORMIGAS(ACS) E GRASP. Bruno Robles Benito, José Roberto Sanches Mantovani, Luis Gustavo Wesz da Silva.– Engenharia Elétrica – Departamento de Engenharia Elétrica – Faculdade de Engenharia de Ilha Solteira – Campus de Ilha Solteira.

O Objetivo deste trabalho é o estudo, análise e implementação das metaheurísticas: Algoritmo Genético, Busca Tabu, Colônia de Formigas e GRASP. Essas metaheurísticas foram implementadas para solução do Problema do Caixeiro viajante (PCV), um problema típico de análise combinatória, no qual um viajante parte de uma cidade inicial e tem que percorrer um determinado número de cidades e voltar para a cidade inicial, logicamente que o caminho a ser percorrido é o menor possível, o caixeiro não pode percorrer cidades já visitadas. Este problema é considerado importante pelo fato de se tornar muito complexo com o aumento de número de cidades, também por estar relacionado a modelos matemáticos de outros problemas físicos do mundo real, especificamente a problemas relacionados à expansão e confiabilidade de sistemas elétricos.

A palavra Metaheurística significa buscar uma solução ótima do problema com artifícios para não cair em ótimos locais, necessitando de alguns procedimentos como:

- Especificar uma forma de identificar um elemento no espaço de busca;
- Especificar a forma de encontrar a função objetivo do elemento candidato à solução;
- Especificar soluções vizinhas;
- Especificar se a transição é realizada de um ponto ou de vários pontos (candidatos à solução)
- Verificar se o processo se realiza através de soluções factíveis ou também podem ser consideradas soluções infactíveis;

Todos estes procedimentos são aplicados em uma metaheurística. Na tabela 1 apresenta-se uma análise comparativa entre os mecanismos básicos das metaheurísticas implementadas

Tabela 1. Comparação entre os mecanismos básicos do AG, TS, ACS e GRASP.

Mecanismos	AG	TS	ACS	GRASP
Codificação	De acordo com as Características físicas do Problema.	Igual do AG.	Igual do AG.	Igual do AG.
População Inicial	Gerada Aleatoriamente.	Uma configuração inicial é gerada aleatoriamente.	Gerada pela função probabilidade do problema.	Gerada pelos melhores custos incrementais da RCL.
Seleção das Configurações	Probabilística através da seleção por torneio.	Determinística através do processo de transição do TS e da lista tabu.	Seleção de um melhor candidato.	Igual do ACS.
Recombinação	Probabilística através da taxa de Recombinação e determinação aleatória do ponto de recombinação.	Determinística através da análise de vizinhança e busca do melhor vizinho com base na lista tabu.	Não apresenta o operador.	Igual do ACS.
Mutação	Probabilística com a diversificação da população através da alteração de bits.	Determinística usando o critério de aspiração.	Não apresenta o operador.	Igual do ACS.
Parâmetros de Controle e Critério de	-Número de Gerações -Tamanho da População -Taxas de racombinação	- Tamanho do tipo de vizinhança. -Tamanho da lista tabu.	-População de formigas. -Coeficiente de	-Coeficiente α . -Critério de Parada.

Parada	e mutação - Critério de parada variada.	-Critério de parada variado	evaporação do feromônio. -Parâmetros α e β -Critério de parada	
Infactibilidade Das Restrições	Tratadas através de funções penalidades	Igual do AG.	Igual do AG.	Igual do AG.

1. Algoritmo Genético (AG)

Desenvolvido por Holland na década de 70, esta metaheurística tem como característica o mecanismo de evolução genética dos indivíduos candidatos, esta evolução é determinada pela seleção de indivíduos melhores adaptados ao meio ambiente pelos processos de recombinação de elementos candidatos, ou seja, crossing-over que transmite para as gerações futuras qualidades dos indivíduos e o mecanismo de mutação. Juntos todos estes processos garantem a evolução dos indivíduos para uma solução do problema.

2. Busca Tabu (Tabu Search – TS)

TS foi um método desenvolvido usando os conceitos de inteligência artificial e vem sendo utilizado em muitos problemas de análises combinatórias, pois este é muito eficaz em problemas complexos.

Esta metaheurística usa dispositivos como memória adaptiva (adaptive memory) possibilitando que o algoritmo não “volte” em configurações já visitadas, daí o nome Busca Tabu; e a exploração sensível (sensitive exploration) que implica que uma escolha ruim pelo TS é melhor que uma boa escolha aleatória.

O TS pode ser resumido como um algoritmo que através de uma solução inicial, são analisadas as suas vizinhanças, e se encontrada um elemento com qualidade superior, este passa a ser a nova solução e a anterior armazenada numa lista (Lista Tabu), para que esta não seja a solução novamente visitada evitando “buracos” no algoritmo. Este processo é realizado até que não seja encontrada nenhuma solução melhor para o problema.

3. Sistema Colônia de Formigas (Ant Colony System – ACS)

O algoritmo ACS foi inspirado no comportamento apresentado pelas formigas reais. Esses insetos apresentam um comportamento coletivo, isto é, visam o bem da colônia e não de cada indivíduo. O que chama a atenção dos cientistas é que as formigas sendo cegas apresentam rotas mínimas entre suas fontes de alimento e o formigueiro. Elas também apresentam a habilidade para se adaptarem a mudanças no ambiente como, por exemplo, encontrar um novo e menor caminho devido a um obstáculo novo. Foi descoberto que o meio usado entre as formigas, para troca de informações relacionadas com o caminho e usado para decidir qual caminho seguir, consiste no cheiro do feromônio. Uma formiga, ao passar por uma rota, deixa uma quantidade de feromônio, marcando o caminho para outras formigas. A probabilidade de uma formiga seguir uma trilha com grande quantidade de feromônio aumenta de acordo com a quantidade dessa substância, assim, quanto maior o número de formigas que passam por certo caminho e o tempo necessário para evaporação do feromônio, maior a probabilidade desse caminho ser percorrido por estes insetos. Desta forma caracteriza-se um comportamento coletivo desses pequenos insetos.

3. GRASP

GRASP (Greedy Randomized Adaptive Search Procedures) é uma metaheurística que consiste em duas fases: fase de construção e a fase de busca local. Na fase de construção é produzida uma solução factível, cujo seus vizinhos são analisados até um mínimo local ser atingido na fase de busca. Em cada iteração da fase de construção um caminho candidato é selecionado para constituir a solução parcial sem

destruir a factibilidade desta. A seleção do próximo candidato para a solução parcial é feita de maneira a escolher os que possuem menor custo incremental da solução. A escolha de cada elemento determina uma Lista de Candidatos Restritos (RCL), essa lista é constituída pelos melhores caminhos candidatos que apresentam menores custos incrementais para a solução, daí o caráter guloso (GREEDY) da metaheurística. O elemento a ser selecionado na RCL é escolhido aleatoriamente (RANDOMIZED) para a solução parcial, que é a característica probabilística da metaheurística GRASP, uma vez escolhido o elemento da lista, esta é atualizada e reavaliados os custos incrementais de cada elemento, que é o aspecto adaptivo (ADAPTIVE) do GRASP.

Depois de construída a solução total, esta não sendo necessariamente a melhor solução do problema, seus 'vizinhos' são analisados a fim de encontrar um ótimo local. O algoritmo é finalizado quando mais nenhum 'vizinho' apresentar melhor função objetivo.

Todos os algoritmos analisados até agora foram programados em linguagem Fortran, uma linguagem de alto nível para engenheiros. Através das metaheurísticas: Algoritmo Genético, TS, ACS e GRASP pode-se analisar o desempenho das mesmas na solução do problema do caixeiro viajante.

O problema do Caixeiro Viajante utilizados nos testes é composto de 100 cidades, e as coordenadas das cidades são as seguintes:

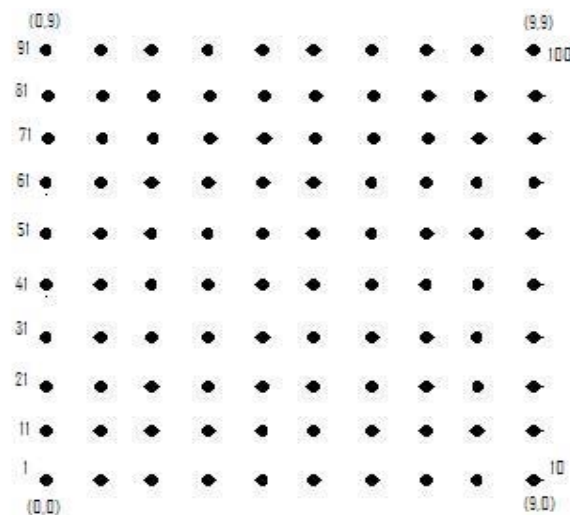


Fig.1. Coordenadas do PCV para 100 cidades.

Este PCV foi simulado com todas metaheurísticas implementadas e o melhor tour é o que apresenta 100 u.m. Depois de obtidos todos os resultados se pôde comparar a eficiência das metaheurísticas implementadas na solução do PCV.

4. Análise Comparativa – AG, TS, ACS e GRASP

Na tabela 2 apresenta-se resumidamente a análise comparativa do desempenho entre todas as metaheurísticas implementadas.

Tabela 2. Comparação do Desempenho das Metaheurísticas Implementadas

Algoritmo	Tempo (CPU)	Função Objetivo	Nº Iterações
AG	5.70 s	100 u.m	200
TS	4.09 s	100 u.m	600
ACS	3.22 s	100 u.m	200
GRASP	1.48 s	100 u.m	10

Observa-se que todas as metaheurísticas analisadas e implementadas convergiram para a solução ótima do problema. O AG apresentou um tempo computacional maior, o que significa um grande esforço

computacional, TS e ACS apresentaram tempos computacionais abaixo do AG, o que mostrou mais eficiência para este PCV. O GRASP se mostrou eficaz e poderoso para este problema, pois apresentou um tempo computacional e número de iterações menor do que os outros algoritmos em questão.

Neste trabalho foram estudadas e implementadas as metaheurísticas do Algoritmo Genético, Busca Tabu, Sistema Colônia de Formigas e GRASP. Estas metaheurísticas foram aplicadas em problemas de análise combinatória de grande importância, que é o caixeiro viajante que é um problema básico, simples, mas que ao aumentar o número de cidades torna-se um problema combinatorial de grandes dimensões.

Todas as metaheurísticas pesquisadas até o momento convergiram para a solução ótima do problema PCV para 100 cidades. O AG se apresentou satisfatório, pois de acordo com seus princípios, foi possível observar a melhoria dos indivíduos da população, convergindo para a melhor solução do problema. O TS convergiu com mais iterações, um esforço computacional maior, mas vale a pena lembrar que o TS programado, não apresentou dispositivos avançados, como técnicas de escolha ou redução de vizinhança, podendo se tornar mais eficaz se implantado esses dispositivos avançados.

ACS se mostrou rápido e eficaz, sendo necessário o uso de alguns artifícios, como a escolha da segunda maior probabilidade da cidade a ser visitada, para uma convergência mais rápida das formigas e evitar ótimos locais, consequentemente, a busca da solução ótima do problema.

A metaheurística GRASP, como citada anteriormente, apresenta poucos parâmetros a ser calibrado no algoritmo, consequentemente, é um algoritmo simples, mas as suas características de construção de soluções permitem que estas sejam de qualidade logo nas primeiras iterações, possibilitando, assim, uma convergência mais rápida para a solução ótima do PCV até a instância considerada para o problema.

- [1] – Dorigo, M., Stützle, T. *Ant Colony Optimization*. Cambridge, Massachusetts, 2004.
- [2] – Dorigo, M., Maniezzo, V., Coloni. *The Ant System: Optimization by a colony of cooperating agents*. Brussels, Belgica, 1996.
- [3] – <http://iridia.ulb.ac.be/dorigo/dorigo>
- [4] – Dorigo, M., Di Caro, G. *Ant Algorithms for Discrete Optimization*. Brussels, Bélgica, 1999.
- [5] Romero, R. e Mantovani, J.R.S. *Introdução às Metaheurísticas*, Anais do 3º Congresso Temático de Dinâmica e Controle da SBMAC. Ilha Solteira, SP, 2004.
- [6] - http://xpusp.sourceforge.net/ga_tutorial.html
- [7] - <http://www.densis.fee.unicamp.br/~franca/EA043/Transpa-Cap-4a.pdf>
- [8] – Glover, F, e Kochenberger, G. *A Handbook of metaheuristic*, Kluwer Academic Publishes, 2003.

Bolsa: CNPq