Universidade de São Paulo

Instituto de Ciências Matemáticas e de Computação

Otimização por Enxame de Partículas (PSO) e Otimização por Colônias de Formigas (ASO) aplicadas ao Problema do Caixeiro Viajante (TSP)

Aluno: Fabricio Aparecido Breve

Prof.: Dr. André Ponce de Leon F. de Carvalho

São Carlos – São Paulo
Maio / 2007

Resumo

Este trabalho mostra a aplicação de dois algoritmos distintos: otimização por enxame de partículas (PSO) e Otimização por colônias de formigas (ASO) para resolver o tradicional problema do caixeiro viajante (TSP).

Introdução

Otimização por enxame de partículas (PSO) é uma forma de inteligência de enxames. Imagine um enxame de insetos, se um vê um caminho desejável para seguir (para obter comida, proteção, etc.) o restante dos elementos do enxame conseguirão segui-lo rapidamente, mesmo que estejam do outro lado do enxame. Por outro lado, para facilitar a exploração do espaço, tipicamente cada partícula deve ter um certo nível de aleatoriedade em seu movimento, para que o movimento do enxame tenha uma certa capacidade exploratória, ou seja, cada partícula deve ser influenciada pelo restante do enxame, mas também deve explorar independentemente até certa extensão.

As partículas modeladas em um espaço multidimensional tem posição e velocidade. Estas partículas estão voando no hiper-espaço e tem essencialmente duas capacidades: memorizar sua melhor posição e saber a melhor posição do enxame. Membros do enxame comunicam suas boas posições para os outros que ajustam suas própria posições e velocidades de acordo com essas boas posições.

Otimização por colônia de formigas (ACO), é uma técnica probabilística para resolver problemas computacionais, os quais podem ser reduzidos a encontrar bons caminhos em grafos. Esta técnica é inspirada no comportamento de formigas em procurar caminhos da colônia até fontes de alimento.

No mundo real, formigas inicialmente andam aleatoriamente em busca de comida, e quando encontram retornam as suas colônias deixando trilhas de feromônio. Se outras formigas encontrarem estes caminhos, elas tendem a não continuar viajando aleatoriamente, mas sim seguir a trilha, retornando e reforçando-a se eventualmente encontrar alimento.

Com o tempo, porém, as trilhas de feromônio tendem a evaporar, reduzindo sua força de atração. Quanto maior o tempo que uma formiga leva para viajar pelo caminho e voltar, mais feromônios irão evaporar. Assim, um caminho relativamente curto será atravessado mais rapidamente e, portanto sua densidade de feromônios será maior.

Portanto, quando uma formiga encontra um bom caminho (curto) da colônia até a fonte de alimento, outras formigas provavelmente seguirão aquele caminho, e através de um retorno positivo eventualmente todas as formigas seguirão tal caminho. A idéia do algoritmo de colônia é copiar esse comportamento com “formigas simuladas” andando pelos grafos que representam o problema a ser resolvido.

Na primeira etapa deste trabalho aplicamos Otimização por enxame de partículas ao tradicional problema do caixeiro viajante (TSP). Na segunda etapa fazemos o mesmo experimento, porém usando otimização por colônias de formigas. No final é aparesentada uma

comparação dos métodos utilizados neste trabalho, juntamente com o os Algoritmos Genéticos aplicados no trabalho anterior.

Especificação do Problema do Caixeiro Viajante (TSP)

Um caixeiro viajante deve visitar N cidades em sua área de vendas. Ele começa de uma base, visita cada cidade uma única vez e retorna à sua cidade no final. A cada viagem está associado um custo proporcional à distância percorrida. O caixeiro deve percorrer a rota mais curta.

Na tabela 1 temos uma lista de oito cidades e a respectiva distância entre elas. O objetivo é ordenar essas cidades (na ordem em que serão visitadas) de forma que a distância total percorrida seja a menor possível.

Tabela 1. Distância entre cidades

	A	B	C	D	E	F	G	H
A	0	42	61	30	17	82	31	11
B	42	0	14	87	28	70	19	33
C	61	14	0	20	81	21	8	29
D	30	87	20	0	34	33	91	10
E	17	28	81	34	0	41	34	82
F	82	70	21	33	41	0	19	32
G	31	19	8	91	34	19	0	59
H	11	33	29	10	82	32	59	0

Otimização por Enxame de Partículas (PSO)

Ao codificarmos uma string de cidades (ou caminhos), considerando n como o número de cidades teríamos n^n soluções possíveis, mas apenas $n!$ soluções válidas, pois o problema determina que cada cidade só poderá ser visitada uma única vez e que todas as cidades devem ser visitadas. Esta limitação já foi observada no trabalho anterior sobre algoritmos genéticos. Esse problema pode ser resolvido através de modificações nas funções de movimentação no hiper-espaço, porém tal alternativa descaracterizaria o algoritmo PSO, cuja principal identidade está nesse algoritmo de movimentação. Dessa forma optamos por apenas penalizar soluções inválidas com um valor de aptidão bastante alto. Assim precisamos contar com uma população relativamente maior e deixar o algoritmo rodar por um tempo maior para tentar contornar essas limitações.

Para realização do experimento foi determinada aleatoriamente uma população de 50 indivíduos, cujos vetores correspondem a uma lista de cidades na ordem em que serão visitadas.

Após 6000 iterações o menor caminho encontrado tem um custo de 140, o mesmo obtido com o algoritmo genético no trabalho anterior. A Figura 1 mostra a variação da maior aptidão obtida no enxame a cada iteração. Podemos observar que no início o valor é bastante alto, devido à penalidade sofrida por todas as partículas, pois inicialmente a posição delas não

correspondia a nenhum caminho válido no problema do TSP. Após algumas iterações caminhos válidos passaram a ser obtidos. Uma melhoria gradativa do melhor caminho é observada a partir daí. Após 1800 iterações o algoritmo parece estabilizar em um valor, mas mesmo com quase 5000 iterações ainda podemos observar mais uma ligeira melhoria.

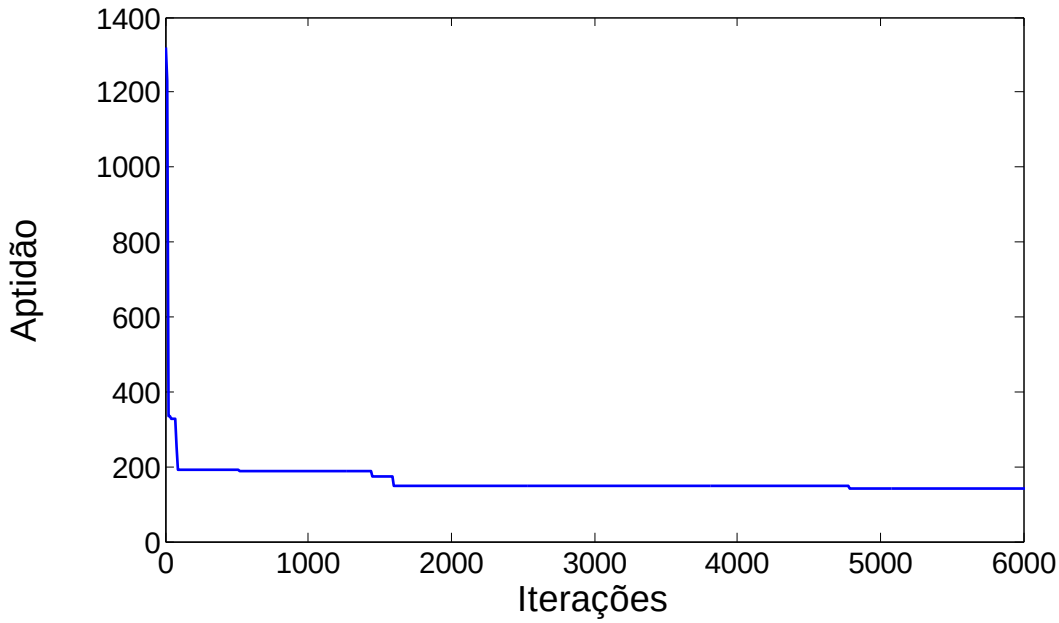

Figura 1 – Variação da maior aptidão entre os elementos da população em cada iteração para o problema do caixeiro viajante utilizando otimização por enxames de partículas (PSO)

Foram feitos alguns experimentos com um número maior de cidades, mas mesmo com um grande número de iterações raramente um caminho válido era encontrado, mesmo utilizando populações maiores e um número bastante grande de iterações. Dessa forma o algoritmo PSO original fica limitado a um número pequeno de cidades, e isto se deve ao fato de a proporção de caminhos inválidos no total de caminhos crescer exponencialmente à medida que se aumenta o número de cidades. Melhores resultados poderiam ser obtidos mesclando outras técnicas, porém descaracterizando o PSO original.

Otimização por Colônia de Formigas (ACO)

Para o experimento com Colônia de Formigas foi escolhida uma população de 8 elementos (número igual a dimensionalidade do problema). A Figura 2 mostra os resultados obtidos. É importante notar que já na segunda iteração o menor caminho já é obtido, este é o mesmo custo do caminho obtido por GA e PSO, porém o ACO produziu o resultado em um tempo muito menor.

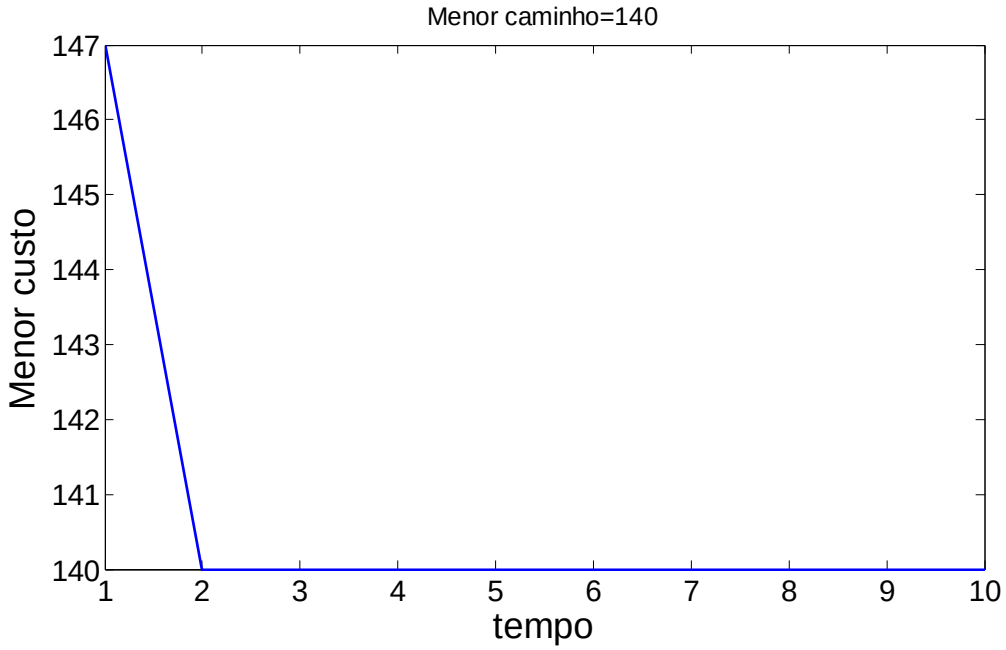

Figura 2 – Variação da maior aptidão dos elementos da população em cada iteração para o problema do caixeiro viajante com 8 cidades utilizando Otimização por colônia de formigas (ACO)

Para explorar melhor o potencial do ACO foi realizado mais um teste, dessa vez com 70 cidades. O número de elementos da população foi novamente o mesmo número de dimensões (70). A Figura 3 mostra os resultados obtidos. Nela podemos observar que o melhor caminho dentre os 70 elementos em cada iteração varia bastante, porém logo nas primeiras iterações o algoritmo já encontra um caminho com custo bastante razoável. O menor caminho do experimento como um todo é obtido na iteração de número 104, e apesar de o algoritmo continuar rodando até 1000 iterações nenhum outro caminho melhor foi encontrado. Na Figura 4 observamos as 70 cidades representadas por pontos verdes e o melhor caminho encontrado pelo algoritmo ACO traçado em azul. Analisando a imagem é possível perceber que o caminho encontrado não é ótimo, porém é um caminho bastante razoável visualmente, principalmente se considerarmos que existem $1{,}20^{100}$ caminhos válidos possíveis e que esta solução foi encontrada em muito menos tempo do que levaria uma busca por força-bruta.

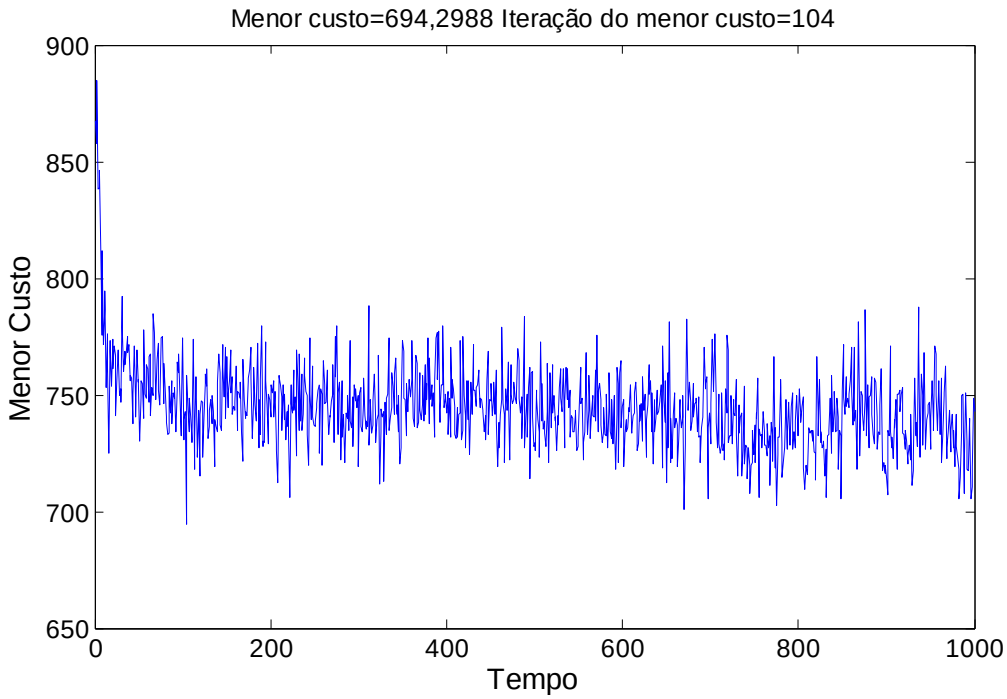

Figura 3 – Variação da maior aptidão dos elementos da população em cada iteração para o problema do caixeiro viajante com 70 cidades utilizando Otimização por colônia de formigas (ACO)

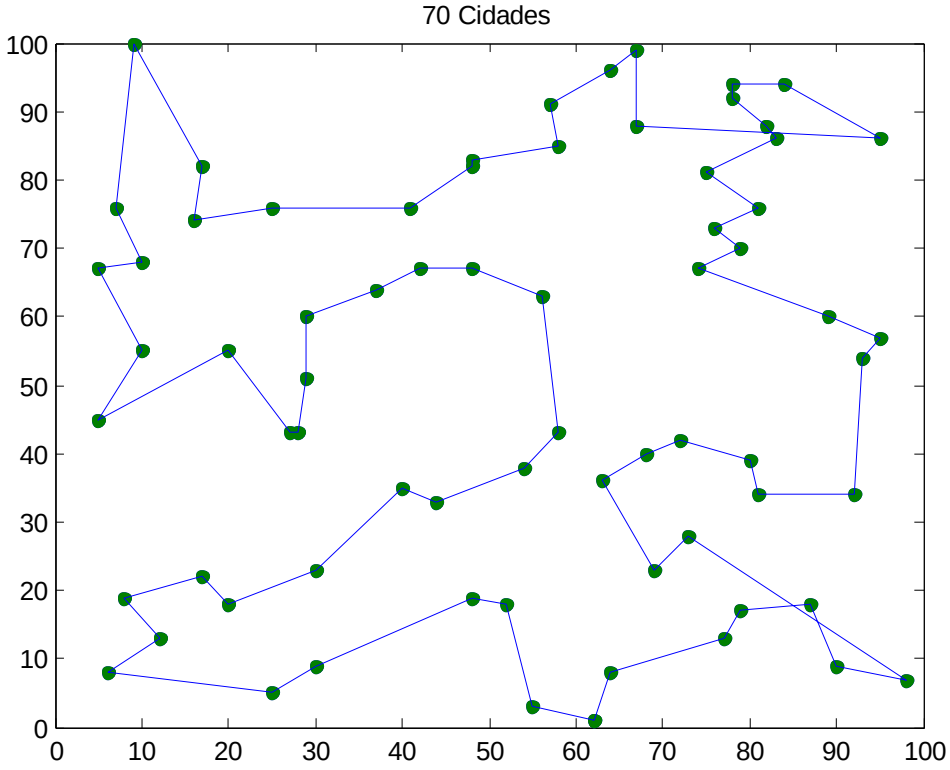

Figura 4 – Problema TSP com 70 cidades e o melhor caminho obtido por Otimização por Colônia de Formigas (ACO)

Conclusão

Dentre os três algoritmos estudados: Algoritmos Genéticos (GA), Enxame de Partículas (PSO) e Colônia de Formigas (ACO), sem dúvida o mais indicado para resolver o problema do Caixeiro Viajante (TSP) é o algoritmo ACO.

O algoritmo ACO é a escolha natural para o problema TSP, pois sua essência consiste em encontrar bons caminhos em grafos. Os algoritmos GA e PSO se baseiam em encontrar soluções navegando em espaços de atributos de alta-dimensionalidade, no caso do TSP as soluções válidas nesse espaço é um número muito reduzido comparado as soluções possíveis de serem obtidas com os algoritmos, assim em sua forma tradicional os algoritmos GA e PSO acabam fazendo algo mais próximo de uma busca aleatória, o que obviamente não é o forte desses algoritmos.

No algoritmo de GA ainda foi possível introduzir mudanças nos algoritmos de geração, crossover e mutação para que o mesmo se tornasse eficiente no problema TSP. Apesar das modificações a essência do algoritmo, que consiste em gerar novos elementos com herança genética de seus pais, é mantida. Porém no caso do PSO, modificações no sistema de vôo teriam de ser introduzidas e isto certamente deixaria o algoritmo muito mais parecido com o que foi feito no GA, porém longe de sua concepção original, e assim tais modificações foram evitadas.

Os experimentos práticos comprovam essas conclusões teóricas, pois notamos que o algoritmo ACO rapidamente encontra o melhor caminho, superando por uma larga margem os algoritmos PSO e GA originais. Mesmo o algoritmo GA modificado, apesar de ser bastante eficiente ainda é ligeiramente inferior ao ACO.