

APLICAÇÃO DA OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS AO PROBLEMA DE ROTEAMENTO DE MÚLTIPLOS VEÍCULOS DE CAPACIDADE LIMITADA

HEITOR S. LOPES*, VILSON DALLE MOLLE†, CARLOS R. ERIG LIMA‡

*Programa de Pós-graduação em Engenharia e Informática – CPGEI
Universidade Tecnológica Federal do Paraná – UTFPR
Av. 7 de setembro, 3165 80230-901 Curitiba (PR)

Emails: hslopes@pesquisador.cnpq.br, vilson@md.utfpr.edu.br, erig@utfpr.edu.br

Abstract— The Capacitated Vehicle Routing Problem – CVRP is a problem present in many daily activities such as garbage collection, mail delivery, school bus transportation. CVRP includes not only the optimization of a path, but many of them simultaneously, since a fleet of evenly-capacitated vehicles have to deliver goods to geographically-distributed customers with variable demand, travelling the least distance as possible. This paper presents an approach for this problem using the Ant Colony Optimization algorithm. This heuristic method is inspired in the behavior of real ants when searching for food. We devised a two-level optimization scheme so as to accomplish both global and local search. We applied the system to several benchmark instances Results obtained so far are very promising for an heuristic method and encourages further research.

Keywords— Ant colony optimization, Capacitated vehicle routing problem, heuristics.

Resumo— O problema do roteamento de veículos com capacidade limitada é onipresente em inúmeras atividades diárias, tais como coleta de lixo, entrega de correspondências e transporte escolar. Este problema inclui simultaneamente a otimização de vários percursos e a distribuição dos mesmos a uma frota de veículos de igual capacidade, de tal maneira a minimizar os custos. Este trabalho apresenta uma solução para este problema utilizando a Otimização por Colônias de Formigas. Esta heurística é baseada no comportamento de formigas reais na busca por alimento. É proposto um esquema em dois níveis buscando tanto uma busca global quanto local. Os resultados da aplicação do método a 7 problemas são bastante promissores, sugerindo sua adequabilidade.

Palavras-chave— Otimização por colônias de formigas, Problema do roteamento de veículos com capacidade limitada, heurística.

1 Introdução

Nas áreas de computação e de engenharia há muitos problemas para os quais não há um algoritmo determinístico para a sua solução ou, quando há, é computacionalmente inviável. Assim, possibilidade de se aceitar uma solução aproximada, de acordo com um determinado critério de satisfabilidade, abre caminho para o desenvolvimento de novos paradigmas. A busca por métodos computacionais que sejam ao mesmo tempo simples (de fácil implementação), robustos (que funcionem satisfatoriamente nas mais variadas situações de um problema), flexíveis (que sirvam para uma vasta gama de problemas) e eficientes (que encontrem soluções de boa qualidade) tem ganhado espaço junto à comunidade de pesquisadores em ciência da computação, pesquisa operacional e engenharias. Assim, muitos métodos heurísticos de otimização têm ganhado atenção e, dentre eles, destacam-se os que são baseados em Computação Evolucionária e Inteligência de Enxames (*Swarm Intelligence*) (Bonabeau et al., 1999).

Há pouco mais de uma década um novo método heurístico de busca e otimização foi criado por Dorigo e seus colaboradores (Dorigo and Gambardella, 1997)(Dorigo and Stützle, 2004). Este método é baseado numa analogia com o processo pelo qual a maioria das formigas estabelecem rotas entre uma fonte de comida e a respectiva colônia. Este método, denominado *Ant Colony Optimiza-*

tion – ACO, pertence ao grupo de técnicas de Inteligência de Enxames e tem sido aplicado com sucesso na resolução de inúmeros problemas reais de grande complexidade, tais como: mineração de dados (Parpinelli et al., 2002), bioinformática (Perretto and Lopes, 2004), otimização combinatorial (Bu et al., 2004) e logística e comunicações (Dorigo and Stützle, 2004).

Dentre os muitos problemas de logística, está o roteamento múltiplo de veículos, sendo um caso particular quando os veículos têm capacidade limitada (Toth and Vigo, 2001). Este problema está presente em muitas organizações, tais como transportadoras e distribuidores de produtos. Muitas tarefas essenciais do dia-a-dia podem ser modeladas como um problema desta natureza, por exemplo: coleta de lixo, transporte escolar e distribuição de correspondências. Este problema envolve não apenas a otimização de um percurso, mas de vários ao mesmo tempo. De maneira geral, a tarefa começa com um conjunto de pontos de parada que devem ser visitados. Baseando-se em algum método, tais pontos são divididos em subconjuntos, cada um correspondendo a um percurso. O conjunto de pontos de cada percurso deve ser ordenado de tal modo que a distancia percorrida seja a menor possível. O objetivo final é minimizar o somatório total das distâncias percorridas em todos os percursos.

2 A heurística ACO

O ACO é um método de busca distribuída e cooperativa inspirada no comportamento de formigas reais. A observação deste comportamento levou à criação de algoritmos para a otimização combinatorial. O algoritmo procura imitar a técnica empregada pelas formigas para estabelecer rapidamente o menor caminho entre uma fonte de alimento e a colônia (Bonabeau et al., 1999). As formigas iniciam a busca ao redor de sua colônia de forma aleatória. À medida que elas se deslocam, deixam no solo um rastro de um composto químico denominado feromônio. O feromônio é um meio de comunicação indireta entre as formigas da colônia mediado pelo ambiente. Este mecanismo é conhecido como estigmergia.

Ao encontrar alimento, uma formiga retorna à colônia marcando sua trilha com feromônio. Outras formigas, circulando aleatoriamente em busca de alimento detectam o rastro de feromônio e são atraídas para este caminho na medida da intensidade do feromônio depositado. Ao seguir um rastro de feromônio, as formigas depositam ainda mais feromônio na trilha que seguem. Quanto maior a quantidade de feromônio maior a probabilidade da trilha ser encontrada por outras formigas e maior a probabilidade destas virem a seguir a trilha. Assim, quanto mais formigas seguirem a trilha, mais consistente a trilha se torna. Entretanto verifica-se que existe uma pequena probabilidade de uma dada formiga não seguir a trilha em toda a sua extensão. Um pequeno desvio cria uma nova trilha, a qual pode ou não ser mais viável do que a anterior. Outras formigas podem detectar esta nova trilha e segui-la segregando seu feromônio (Beckers et al., 1992).

O feromônio segregado pelas formigas evapora continuamente. Desta forma, se a trilha deixa de ser usada por algum tempo ela tende a desaparecer, pois o nível de feromônio fica abaixo do limiar para influenciar outras formigas a seguirem a trilha. Este processo autocatalítico ajuda as formigas a estabelecerem o menor caminho, “esquecendo” trilhas menos atrativas (mais longas).

Este mecanismo comportamental pode ser utilizado para resolver problemas combinatoriais através de simulação, com formigas artificiais buscando no espaço possível de soluções. Uma memória adaptativa pode simular as trilhas de feromônio, bem como uma função de *fitness* pode medir a qualidade de uma solução como analogia à qualidade do alimento e/ou sua distância à colônia.

3 O problema do caixeiro viajante e o roteamento de veículos

O problema do caixeiro viajante (*Travelling Salesman Problem* – TSP), pode ser enunciado como: Dado um conjunto de cidades, o viajante deverá

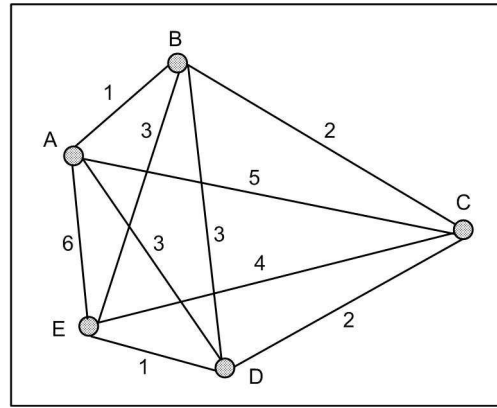


Figura 1: Uma instância exemplo do TSP simétrico com cinco cidades.

realizar um percurso por todas as cidades retornando ao ponto de partida. Cada cidade deve ser visitada uma única vez. A Figura 1 apresenta uma instância do TSP com cinco cidades ABCDE, representada pelo grafo $G(V, E)$ no qual cada vértice $v_i \in V'$ representa uma cidade e cada aresta $e_{ij} \in E$ representa um caminho entre um par de cidades (Toth and Vigo, 2001).

Encontrar a solução para o TSP consiste em encontrar o menor caminho passando por todas as cidades uma única vez e retornando ao local de partida. Um exemplo de percurso seria *ABCDEA* com custo total de 12 unidades, sendo que o custo é a soma dos pesos dos arcos na seqüência. Outro percurso poderia ser *ABCEDA* com custo 11.

O problema do roteamento de veículos é um problema central em logística de distribuição. Em particular, o problema de roteamento de veículos capacitado – CVRP (*Capacited Vehicle Routing Problem*) pode ser definido como: a partir de um depósito central (em alguns casos pode ser mais de um), um determinado número de clientes devem ser atendidos em suas demandas por um mesmo produto, através de uma frota de veículos cuja capacidade é conhecida. Deve-se encontrar o conjunto de percursos de custo mínimo e que atenda à demanda de todos os clientes (Toth and Vigo, 2001). Como restrições básicas, tem-se:

- cada cliente é atendido uma única vez por um único veículo;
- a demanda total coberta por um veículo não pode exceder a sua capacidade;
- cada uma dos percursos inicia-se e termina no depósito;
- a distância total percorrida em um percurso não pode exceder a autonomia do veículo.

A Figura 2 representa um exemplo de instância do problema com um depósito e dez consumidores. Neste exemplo, para suprir a demanda, são

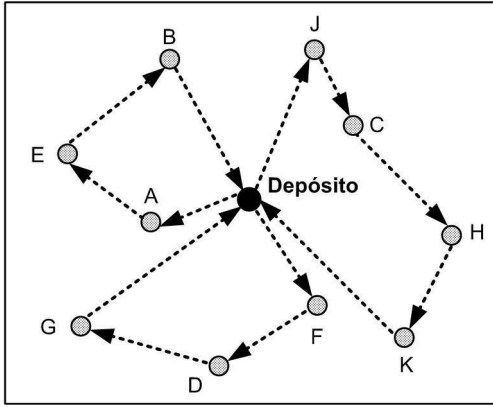


Figura 2: Uma instância exemplo de CVRP com um depósito e dez consumidores.

necessárias três percursos (linhas tracejadas) que partem do depósito e retornam a ele fazendo o menor percurso possível entre as cidades visitadas.

O CVRP é um problema *NP*-difícil (Lenstra and Kan, 1981), pois contém o problema do caixeiro viajante como subproblema. Obviamente, o CVRP é muito mais difícil de ser resolvido do que o TSP, pois é um problema em dois níveis: primeiro, há que se descobrir quais destinos serão agregados em percursos (sem o conhecimento prévio do número de percursos); segundo, para cada percurso, deve-se encontrar o menor percurso possível. De maneira genérica, o primeiro nível se trata de um problema de empacotamento (*bin packing*), enquanto que o segundo, é um problema de TSP.

CVRP pode ser definido como se segue. Dado um grafo $G(V, E)$, sendo V o conjunto de vértices $V = \{v_0, v_1, \dots, v_n\}$ e E o conjunto de arestas $E = \{e_{ij}\}$ onde $e_{ij} = \text{aresta}(v_i, v_j) \forall v_i, v_j, i \neq j$. Considerando-se a localização do depósito em v_0 , tem-se $V' = V \setminus \{v_0\}$ como o conjunto de destinos. A cada vértice $v_i \in V'$ está associado um valor $d \in \mathbb{R}$ que representa a demanda do destino. O conjunto E é o conjunto de caminhos possíveis. Logo, a cada arco e_{ij} de E está associado um custo c_{ij} correspondendo à distância entre os vértices v_i e v_j . Para um dado veículo X_i com capacidade de carga C^x , o vetor R^{X_i} contém o conjunto de segmentos referentes ao percurso deste veículo, ou seja, o conjunto de destinos atendidos pelo veículo X_i . A solução do problema é o conjunto de vetores R^{X_i} cuja soma do custo de cada caminho seja mínimo.

4 Metodologia

O algoritmo proposto divide o CVRP em dois níveis. No primeiro, as formigas são postas a marcar o conjunto de percursos necessários à tarefa. Durante este processo as formigas artificiais imitam a vida real e determinam uma solução aproximada. Cada formiga é levada a desempenhar o trabalho

da frota como um todo, estabelecendo trilhas para todos os veículos. No segundo nível, cada percurso de um veículo é submetido a um novo conjunto de formigas, executando um algoritmo ACO/TSP com 20 ciclos, na expectativa de conseguir um melhoramento local da solução. O processo se repete como um todo até que o critério de parada seja atingido, sendo este um número pré-estabelecido de ciclos. Esta etapa se baseia no algoritmo ACO original proposto por Dorigo e Gambardella para o TSP (Dorigo and Gambardella, 1997). Ao final do último ciclo, a melhor solução encontrada passa novamente pelo algoritmo ACO/TSP, desta vez com 100 iterações, para tentar melhorar ainda mais a melhor solução obtida.

Conceitualmente, esta metodologia é consistente com a formalização do CVRP pois, como o segundo nível reduz-se a um problema de TSP para cada percurso separadamente, procura-se, em primeiro lugar, uma otimização global (distribuição dos destinos em percursos, satisfazendo as restrições) e, em seguida, uma otimização local (minimizando cada percurso previamente definido). Esta abordagem metodológica é a principal diferença deste trabalho com outros semelhantes, tais como (Gambardella et al., 1999) e (Bullnheimer et al., 1997).

4.1 Marcação da trilha e construção de percursos

Cada trilha é marcada através da segregação de feromônio ao término de cada iteração. A trilha de feromônio (τ) é depositada sobre os percursos de cada veículo na proporção inversa do custo da solução (S_i), conforme a equação 1:

$$\tau^{S_i} = \frac{r}{c^{S_i}} \quad (1)$$

onde r é a quantidade unitária de feromônio para a marcação de trilhas e c^{S_i} é o custo da solução S_i , dado pelo somatório das distâncias Euclidianas entre os pontos definidos pelo percurso, iniciando e terminando no depósito.

A cada percurso, a segregação ocorre segundo o produto da proporção do feromônio por custo do percurso, pela proporção da demanda atendida por este curso, conforme a equação 2:

$$\tau^{P_i} = \frac{\tau^{S_i} \cdot d}{c^{P_i}} \quad (2)$$

onde τ^{P_i} é a quantidade de feromônio depositada ao longo do percurso P_i , $i = 1..k$; c^{P_i} é o custo do percurso P_i e d é a demanda atendida (proporcional à demanda total). Assim, para cada par qualquer de vértices (m, n) pertencente ao percurso, haverá uma quantidade de feromônio $\tau_{m,n}$ como contribuição de todas as formigas da colônia.

No primeiro ciclo do ACO, quando não há nenhuma trilha de feromônio ainda, as formigas escolhem o próximo passo (consumidor) aleatoriamente. Nos ciclos seguintes, uma vez que uma

formiga esteja num dado vértice do grafo, ela escolhe o próximo levando em conta tanto a sua experiência acumulada (dada pela quantidade de feromônio depositada nas arestas do grafo – τ) e um termo heurístico – η), conforme referenciado anteriormente. Quando uma formiga k estiver no vértice i , a probabilidade que o vértice j seja escolhido pela formiga e incluído no percurso parcial atual é dado pela equação 3 (Dorigo and Stützle, 2004):

$$p_{ij}^k = \frac{(\tau_{ij})^\alpha \cdot (\eta_{ij})^\beta}{\sum_{i,j \in T} \{(\tau_{ij})^\alpha \cdot (\eta_{ij})^\beta\}} \quad (3)$$

onde α e β são parâmetros controlam a intensidade da busca global e local, respectivamente, através da ponderação da quantidade de feromônio já existente na trilha e da distância percorrida. T é o conjunto de vértices ainda não visitados.

É importante notar que somente vértices não-visitados são considerados por uma formiga a cada ciclo evitando, assim, a construção de percursos inválidos. De acordo com a heurística ACO, evaporação ocorre em função do tempo, o qual é discretizado segundo a velocidade da formiga. Assim, uma unidade de tempo corresponde a um passo da formiga. As formigas se movimentam em conjunto e, após cada iteração, ocorre a atualização do conjunto de percursos e seus respectivos valores de feromônio.

4.2 Implementação

O sistema foi desenvolvido em Java para ter portabilidade para várias plataformas diferentes. O sistema aceita como entrada arquivos texto, no formato usual dos arquivos de *benchmark*. Estes arquivos contêm informações sobre a instância a ser executada (p.ex. dimensão, número máximo de veículos, capacidade dos veículos, etc), uma seção com as coordenadas Euclidianas dos clientes, uma seção com as demandas individuais de cada cliente e uma seção com a localização do depósito (a presente versão trabalha somente com instâncias com um depósito). A interface com o usuário mostra, de maneira dinâmica, o mapa dos clientes e depósito, bem como as rotas formadas pelas formigas, conforme mostrado na figura 3. Desta maneira, o usuário pode acompanhar o andamento do processo, bem como intervir no mesmo mudando os parâmetros de controle durante a rodada. Estão acessíveis ao usuário os seguintes parâmetros de controle do ACO: velocidade de evaporação do feromônio, número máximo de unidades de feromônio depositadas numa trilha, porcentagem de feromônio residual (valor mínimo para que uma trilha seja considerada), α e β , número de formigas, e a escala do desenho. O usuário também visualiza o nome do arquivo de teste em execução e o custo total da melhor solução obtida até o momento.

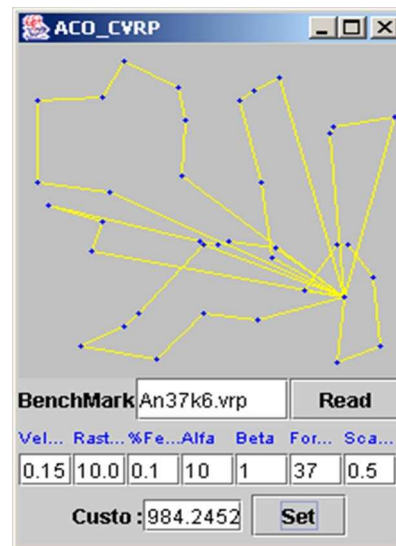


Figura 3: Interface com o usuário do sistema ACO-CVRP.

5 Experimentos, Resultados e Discussão

Para a avaliação do sistema proposto, foram utilizadas instâncias de CVRP obtidas na Internet¹ que são freqüentemente utilizadas como *benchmark* para teste de algoritmos. A tabela 1 mostra as instâncias utilizadas, sendo que o número após “n” representa a dimensão (isto é, o número total de clientes a serem atendidos) e o número após “k” significa o número máximo de veículos para o atendimento dos clientes. As letras iniciais das instâncias significam os autores que primeiramente as introduziram, respectivamente “A”, “E” e “F”, para (Augerat et al., 1998), (Christofides and Eilon, 1969) e (Fisher, 1994).

Para cada uma das instâncias foram realizados vários experimentos, de tal maneira a experimentar diferentes configurações dos parâmetros de controle. Para os resultados reportados aqui, os seguintes parâmetros foram idênticos para todas as instâncias: número de ciclos até a parada: 500; $\alpha=10$; $\beta=1$; e unidades de feromônio depositadas: 10. Para estas mesmas instâncias, outros parâmetros foram experimentados nos seguintes intervalos: velocidade de evaporação de feromônio: 0,15 a 2,0; número de formigas: 40-250. Os parâmetros α e β foram ajustados empiricamente de tal maneira a privilegiar a busca global em relação à busca local, numa relação de 10:1. Quando o algoritmo era executado com valores baixos de α (em relação a β) foi observado que a convergência ocorria muito rapidamente para soluções de baixa qualidade. Isto acontecia mesmo compensando-se com uma diminuição na quantidade unitária de feromônio depositado ou com um aumento da taxa de evaporação de feromônio. Isto se deve à meto-

¹<http://elib.zib.de/pub/Packages/mp-testdata/tsp/tsplib/vrp> e <http://neo.lcc.uma.es/radi-aeb/WebVRP/data/instances/Augerat>

dologia com que foi implementado o sistema, com otimização em dois níveis.

Na tabela 1 são mostrados os custos das melhores soluções conhecidas e os respectivos custos encontrados com o ACO proposto. Nesta tabela, a coluna “Diferença” significa o valor relativo (em porcentagem) entre a melhor solução encontrada e o ótimo conhecido.

Tabela 1: Instâncias de teste utilizadas juntamente com o ótimo conhecido e os resultados obtidos com o ACO proposto.

| Instância | Ótimo | Melhor | Diferença (%) |
|-----------|-------|---------|---------------|
| A-n32k5 | 784 | 798,29 | 1,82 |
| A-n37k6 | 949 | 984,24 | 3,71 |
| A-n53k7 | 1010 | 1043,13 | 3,28 |
| A-n60k9 | 1408 | 1420,43 | 0,88 |
| A-n80k10 | 1764 | 1886,13 | 6,92 |
| E-n101k8 | 825 | 899,13 | 8,98 |
| F-n135k7 | 1165 | 1234,42 | 5,96 |

As melhores soluções obtidas para duas das instâncias estudadas são mostradas nas figuras 4 e 5 a seguir. As soluções para as demais instâncias, bem como a sua comparação visual com a solução ótima não são mostrados aqui pela limitação de espaço.

A figura 6 mostra a curva de evolução do algoritmo ACO para a instância A-n60k9. Como pode ser observado, a convergência é bastante significativa no início do processo e posteriormente apresenta uma comportamento assintótico, sugerindo a estagnação da evolução. A razão disto é que no início do processo, é grande a probabilidade de uma formiga encontrar um percurso de menor custo do que a melhor solução corrente, posto que há muitas possibilidades de caminhos alternativos a serem explorados. Desta forma, mesmo com comportamento estocástico, o algoritmo apresenta alta convergência. À medida que o algoritmo evolui e a solução corrente vai sendo melhorada em direção do ótimo global, o número de modificações benéficas possíveis a cada passo se torna paulatinamente menor e, conseqüentemente, a probabilidade de uma formiga escolher aleatoriamente um novo percurso que implique numa melhoria da solução torna-se cada vez menor. A aplicação de TSP nos percursos individuais de cada veículo promoveu uma pequena, porém importante, extensão no processo de convergência. Isto sugere que algoritmo possa escapar, eventualmente, de máximos locais. Durante a evolução, não é possível distinguir claramente o efeito deste procedimento. Porém, pode ser evidenciado no final da evolução. No gráfico da figura 6 observa-se uma pequena diminuição no custo da solução exatamente no final da evolução, sugerindo, então, que a busca local pode ser efetiva para uma melhoria adicional da solução obtida até então.

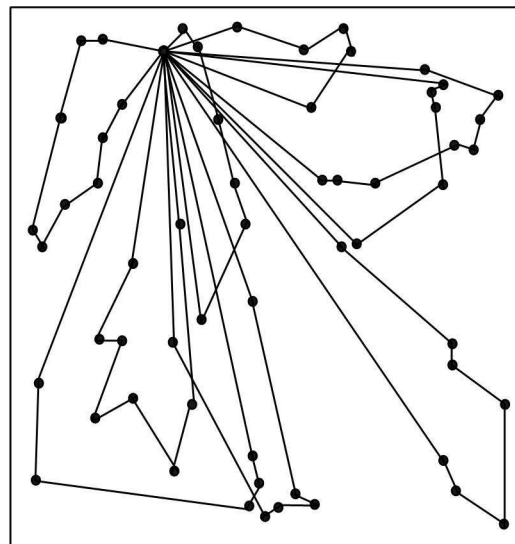


Figura 4: Melhor solução obtida para a instância A-n60k9.

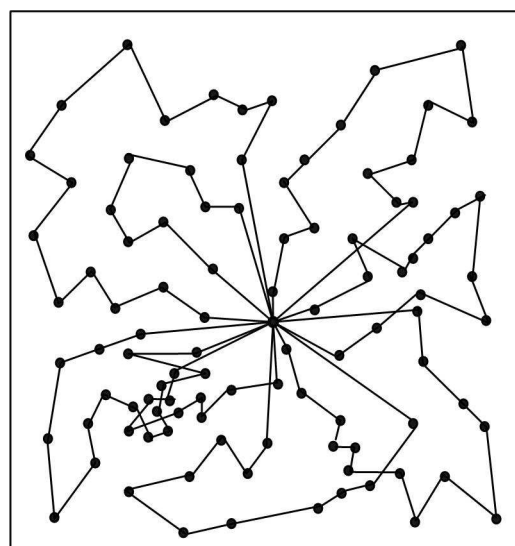


Figura 5: Melhor solução obtida para a instância E-n101k8.

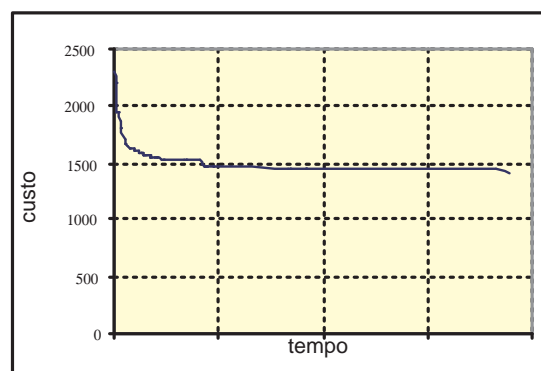


Figura 6: Curva de evolução do ACO para a instância A-n60k9 do CVRP. Neste gráfico, o eixo x refere-se às iterações do algoritmo.

6 Conclusões e Trabalhos futuros

Este trabalho apresentou os resultados da aplicação da heurística ACOa um problema relevante de logística: o problema de roteamento múltiplo de veículos com limitação de capacidade. A abordagem proposta divide o problema em dois níveis de otimização. Primeiramente as formigas buscam descobrir percursos dividindo os clientes em grupos e, em seguida, aplica-se uma otimização local, na forma de um TSP, para otimizar cada percurso individualmente.

O sistema foi aplicado a sete instâncias de CVRP, com dimensão variando de 32 a 135 clientes (n) e frota de 5 a 10 veículos (k). Os resultados obtidos foram bastante promissores para um método heurístico, considerando também que não foi feita nenhuma tentativa para a otimização dos parâmetros de controle do algoritmo. Foi observado que os parâmetros de controle têm uma influência bastante significativa no resultado final, podendo fazer o algoritmo estagnar rapidamente ou então não convergir para uma solução satisfatória.

Num primeiro momento, não foi possível determinar se há alguma relação específica entre a complexidade da instância (dimensão de n e de k) e a dificuldade do algoritmo para prover soluções satisfatórias. O que é evidente, e esperado, é que a qualidade da solução tende a diminuir à medida que ambos n e k aumentam.

De maneira geral, os resultados sugerem a viabilidade da pesquisa. Dada à sensibilidade do algoritmo aos parâmetros de controle, poderá ser necessário um teste exaustivo para encontrar um conjunto de parâmetros otimizados para vasta gama de instâncias. Num aperfeiçoamento futuro pretende-se modificar o algoritmo para que cada formiga simule um único veículo. Assim, um conjunto de k formigas representaria uma frota de k veículos. Para esta abordagem as formigas compartilhariam uma referência ao conjunto de cidades ainda não visitadas. Observou-se que os percursos gerados nas instâncias estudadas são pequenos e, eventualmente poderia ser viável uma busca exaustiva para a otimização de tais percursos, mesmo às custas de um custo computacional elevado. Se, eventualmente, o percurso a ser otimizado for muito grande, algum outro procedimento de busca local, do tipo *branch-and-bound* poderia ser igualmente útil.

Agradecimento

O autor H.S. Lopes agradece o CNPq pelo apoio através da bolsa PQ, processo 305720-2004-0.

Referências

Augerat, P., Belenguer, J., Benavent, E. and et al. (1998). Separating capacity constraints in the

cvrp using tabu search, *European Journal of Operations Research* **106**(2-3): 546–557.

Beckers, R., Deneubourg, J. and Goss, S. (1992). Trails and u-turns in the selection of the shortest path by the ant *Lasius niger*, *Journal of Theoretical Biology* **159**: 397–415.

Bonabeau, E., Dorigo, M. and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, Oxford, UK.

Bu, T., Yu, S. and Guan, H. (2004). Binary-coding-based ant colony optimization and its convergence, *Journal of Computer Science and Technology* **19**(4): 472–478.

Bullnheimer, B., Hartl, R. and Strauss, C. (1997). Applying the ant system to the vehicle routing problem, *Proc. 2nd International Conference on Metaheuristics*.

Christofides, N. and Eilon, S. (1969). An algorithm for the vehicle dispatching problem, *Operations Research Quarterly* **20**: 309–318.

Dorigo, M. and Gambardella, L. (1997). Ant colonies for the travelling salesman problem, *Biosystems* **43**: 73–81.

Dorigo, M. and Stützle, T. (2004). *Ant Colony Optimization*, The MIT Press, Cambridge, USA.

Fisher, M. (1994). Optimal solution of vehicle routing problems using minimum k-trees, *Operations Research* **42**: 626–642.

Gambardella, L., Taillard, E. and Agazzi, G. (1999). *New Ideas in Optimization*, McGraw-Hill, New York, chapter ACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows, pp. 63–76.

Lenstra, J. and Kan, A. R. (1981). Complexity of vehicle routing and scheduling problems, *Networks* **11**: 221–227.

Parpinelli, R., Lopes, H. and Freitas, A. (2002). Data mining with an ant colony algorithm, *IEEE Transactions on Evolutionary Computation* **6**(4): 321–332.

Perretto, M. and Lopes, H. (2004). Reconstruction of phylogenetic trees using the ant colony optimization paradigm, *Proc. of Third Brazilian Workshop on Bioinformatics*, pp. 49–56.

Toth, P. and Vigo, D. (2001). *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications, SIAM Press, Philadelphia, USA.