

Um algoritmo genético com infecção viral para o problema do caixeiro viajante

Allison da Costa Batista Guedes, Jéssica Neiva de Figueiredo Leite e Dario José Aloise

Departamento de Informática de Matemática Aplicada, Programa de Educação Tutorial – PET Ciência da Computação

Resumo

O Problema do Caixeiro Viajante (PCV) é um problema clássico da área de otimização combinatória, NP-árduo, cujo objetivo é encontrar o ciclo hamiltoniano de menor custo em um grafo ponderado. Devido à sua complexidade computacional, o PCV tem sido amplamente abordado no desenvolvimento de algoritmos aproximativos e metaheurísticas. No presente trabalho, é apresentado um Algoritmo Genético (AG) com Infecção Viral para esse problema, no qual, se mostrou com melhor performance e eficiência do que o AG padrão. Para se chegar a essa conclusão foram feitos testes e análises dos resultados com instâncias disponibilizadas na TSPLIB.

Palavras-chave: Problema do Caixeiro Viajante, Algoritmo Genético, Infecção Viral.

Abstract

The Traveling Salesman Problem (TSP) is a classic problem in the combinatorial optimization area, NP-hard, whose objective is to find the minimal cost hamiltonian cycle on a weighted graph. Because of its computational complexity, the TSP has been widely employed on the development of approximative algorithms and metaheuristics. In the current paper, it is presented a Genetic Algorithm (GA) with Viral Infection for this problem, which has achieved a better performance and efficiency than the standard GA. This conclusion came out from analysis of the results of tests made with instances from the TSPLIB.

Keywords: Traveling Salesman Problem, Genetic Algorithm, Viral Infection.

Introdução

O Problema do Caixeiro Viajante (PCV) é um dos mais conhecidos na área de otimização combinatória (HOFFMAN e WOLFE, 1985); (MELAMED *et al.*, 1990). Pela sua importância prática e dificuldade de solução, ele tem atraído a atenção de muitos pesquisadores das áreas da matemática, da engenharia e da ciência da computação (PAPADIMITRIOU e STEIGLITZ, 1982); (LAPORTE *et al.*, 1996); (GUTIN e PUNNEN, 2002). Esse problema pode ser estabelecido da seguinte forma: dado um conjunto de cidades e o custo da viagem entre cada par delas, procurar encontrar a maneira mais barata de visitar todas as cidades e retornar ao seu ponto de origem, visitando cada cidade exatamente uma vez. De uma maneira formal temos: dado um grafo $G(V, E)$, com $|V| \geq 3$ e custos c_{ij} , $(i, j) \in E$, associados às arestas, o Problema do Caixeiro Viajante (PCV) consiste em encontrar o ciclo hamiltoniano de custo mínimo; que no caso de um grafo completo com n vértices (cidades) significa buscar a melhor rota dentre $(n - 1)!/2$ possibilidades.

Embora possa ser descrito de uma maneira simples, o PCV pertence à classe dos problemas NP-difíceis (GAREY e JOHNSON, 1979), ou seja, não pode ser resolvido por um algoritmo polinomial, a menos que a classe $P = NP$, tornando-o intratável para obtenção da solução por métodos exatos para problemas de grande porte. Por exemplo: na manufatura de uma placa de circuito impresso, é importante determinar a melhor ordem em que um laser perfurará milhares dos furos (CAMPOLLO e MACULAN, 1994). Da mesma forma, esse procedimento poderia ser utilizado em aplicações de soldas/rebites na fabricação de veículos.

Com isso, abordagens aproximativas (heurísticas/metaheurísticas) e exatas têm sido buscadas para a resolução de instâncias de grande porte desse problema. Como referências para avanços na abordagem exata, destacamos, dentre vários, os trabalhos de: (DANTZIG *et al.*, 1954), (GRÖTSCHEL, 1980); (CROWDER e PADBERG, 1980); (PADBERG e RINALDI 1987, 1991); (GRÖTSCHEL e HOLLAND, 1991); (APPLEGATE *et al.*, 1995). Na abordagem aproximativa, destacamos para as heurísticas clássicas os trabalhos de: (LIN e KERNIGHAN, 1973); (CHRISTOFIDES, 1976); (ROSENKRANTZ *et al.*, 1977) e (GOLDEN *et al.*, 1980). E na vertente das metaheurísticas, as referências: (REEVES, C. R., 1993); (JOHNSON, D.S. e MCGEOCH, 1997); (GLOVER, 1999); (HANSEN e RIBEIRO, 2001); (GLOVER e KOCHENBERGER, 2003) e (IBARAKI, T. *et al.*, 2005).

O atual trabalho apresenta uma metaheurística denominada Algoritmo Genético, usando Infecção Viral, para obtenção de uma solução aproximada de boa qualidade e em tempo polinomial para o Problema do Caixeiro Viajante. A pesquisa tem por objetivo fazer uma comparação entre o método convencional e a variante proposta.

Algoritmo Genético

Os Algoritmos Genéticos foram introduzidos pelo matemático John H. Holland (1975), para a resolução de problemas de diversas áreas, dentre as quais, se inserem a Biologia, as Engenharias, a Matemática aplicada e a Ciência da Computação (HOLLAND, 1975).

O princípio de funcionamento desse algoritmo consiste em explorar o espaço de busca através de uma população de soluções viáveis (cromossomos), que evolui a cada geração por meio de operadores genéticos de *crossover* e de mutação, em analogia ao processo da evolução natural.

Os cromossomos podem ser representados de inúmeras formas. No caso do problema em questão, os cromossomos são representados por permutações do conjunto de cidades. Por exemplo: em um conjunto de cidades representadas pelos números de 1 a 8, a permutação {3, 8, 7, 2, 1, 5, 4, 6} representaria uma das soluções viáveis possíveis para o problema.

O AG padrão pode ser descrito, de uma maneira geral, pelo seguinte pseudo-código:

1. [Inicialização] Gerar uma população inicial de n cromossomos, aleatoriamente, e determinar a fitness de cada cromossomo;
2. [Geração da Nova população] Criar uma nova população através da aplicação das seguintes etapas:
 - a) [Seleção] Selecionar dois cromossomos-pais da população atual de acordo com sua fitness;
 - b) [Crossover] Fazer o cruzamento dos pais para formar novos indivíduos (filhos);
 - c) [Mutaç o] Aplicar muta o nos novos indiv duos;
3. [Avaliar nova popula o] Calcular a fitness de cada cromossomo da popula o recém-gerada;
4. [Teste de parada] Se condi o de parada satisfeita: finalizar retornando a melhor solu o encontrada. Caso contr rio, voltar ao passo 2.

As opera es de *crossover* e de muta o podem ocorrer, ou n o, dentro de uma probabilidade fixa (a probabilidade de muta o   individual para cada elemento do cromossomo). Para o algoritmo em quest o, esses par metros foram deixados fixos em 80% e 1%, respectivamente, como   encontrado na literatura sobre algoritmos gen ticos. Os processos de *crossover* e de muta o podem ocorrer de diversas formas, dependendo do tipo de representa o (dos cromossomos) utilizada. Neste estudo, foram utilizados os operadores de *crossover* de ordem OX1 e de *muta o* de troca EM (LARRAÑAGA *et al*, 1999). Para a sele o dos pais, foi utilizado um processo de sele o por *rank*, no qual, a probabilidade de um indiv duo ser escolhido depende da sua posi o na popula o, depois dessa ser ordenada de

acordo com a *fitness* dos cromossomos (os de maior *fitness* têm maiores chance de serem escolhidos).

Algoritmo Genético com Infecção Viral

O Algoritmo Genético com Infecção Viral (AGIV) é um variante do AG (inspirado na teoria da evolução das espécies de Darwin) que utiliza o operador de infecção no lugar do operador de mutação. Esse método consiste em manter uma população extra, denominada População de Vírus, que contém trechos de soluções viáveis do problema (KANO, H. *et al*, 1996); (KANO, H. e NAKAMURA, T., 2000); (KANO, H. e NAKAMURA, N., 2000). Nesse caso, cada vírus representa uma parte da rota a ser determinada. Por exemplo: seja uma rota representada pela sequência {8, 1, 3, 5, 2, 6, 4, 7, 8}; um possível vírus poderia ser o trecho {3, 5, 2}.

O AGIV idealizado é descrito pelo seguinte pseudo-código:

1. [Inicialização] Gerar uma população inicial de n cromossomos, aleatoriamente, e determinar a *fitness* de cada cromossomo. Gerar uma população inicial de vírus, aleatoriamente;
2. [Infecção] Aplicar o operador de infecção nos melhores indivíduos da população.
3. [Geração da Nova população] Criar uma nova população através da aplicação das seguintes etapas:
 - a) [Seleção] Selecionar dois cromossomos-pais da população atual de acordo com sua *fitness*;
 - b) [Crossover] Fazer o cruzamento dos pais para formar novos indivíduos (filhos).
4. [Avaliar nova população] Calcular a *fitness* de cada cromossomo da população recém gerada;
5. [Teste de parada] Se condição de parada satisfeita: finalizar retornando a melhor solução encontrada. Caso contrário, voltar ao passo 2.

Infecção Viral

No procedimento de infecção, um vírus é selecionado pelo mesmo processo de *rank* que selecionou os pais. Porém, como o vírus não possui valor de *fitness* (afinal não é uma solução completa), é atribuído a ele um valor o qual chamamos de Infectibilidade. Esse é um indicador de quão bem o vírus tem atuado. Depois de selecionarmos o vírus que efetuará a infecção (no cromossomo dado), realizamos o processo de transcrição, que consiste em o vírus modificar o cromossomo infectado de forma que esse contenha um trecho idêntico ao representado pelo agente infectante e fazer os ajustes necessários para a manutenção da viabilidade do indivíduo receptor da carga viral.

Quando um vírus é criado (ou modificado, como veremos a seguir), sua infectibilidade é configurada em um valor inicial fixo. Quando a infecção ocasiona uma melhora da *fitness* do cromossomo, o valor da infectibilidade do vírus é aumentado em 1 (até um certo limite, fixado antes da execução do algoritmo). Caso a infecção faça com que a *fitness* do cromossomo piore, decrementamos a infectibilidade do vírus (até chegar a 0). Caso o vírus chegue a uma infectibilidade igual a 0, ele descarta uma de suas partes e copia uma parte do cromossomo para si (a esse processo, demos o nome de transdução). Esse procedimento caracteriza a analogia de infecção viral nos seres vivos, onde o vírus se modifica para melhor se adaptar ao ambiente.

Esse processo se repete a cada interação e, à medida que a população de indivíduos completos melhora, a população de vírus tende a ficar com sub-rotas melhores, ou seja, trechos de soluções que são mais desejáveis em uma rota completa.

Experimentos computacionais

Para fazer uma comparação entre os métodos, vários testes foram realizados. Esses testes consistiram de executar cada um dos algoritmos (conforme explicado adiante) e verificar a qualidade média das soluções encontradas por cada um deles.

Os experimentos computacionais foram realizados da seguinte maneira: foram escolhidas 10 instâncias da TSPLIB, que é uma biblioteca de exemplos específicos para o Problema do Caixeiro Viajante (Traveling Salesman Problem) com seus valores exatos e disponibilizada via Internet no site: <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95>.

Para cada uma dessas instâncias, foram realizados 30 testes com o tempo limite de 10 minutos, num computador Pentium IV 1.4 GHz, usando cada um dos algoritmos heurísticos (AG e AGIV). Portanto, foram realizados $30 \times 10 \times 2 = 600$ testes.

No decorrer dos testes, para cada uma das instâncias, foram sendo verificadas as performances dos algoritmos AG e AGIV de 1 em 1 minuto até atingir o limite de 10 minutos. Isso foi feito para cada um dos 30 testes relativos a cada instância.

Resultados

Na figura 1, está representada a performance temporal da fitness, para a instância rbg403 (assimétrica), indicando melhor performance e eficiência do AGIV.

A tabela 1 mostra o ganho obtido pelo Algoritmo Genético com Infecção Viral (AGIV), o que respalda a pesquisa realizada.

Tabela 1. Resultados comparativos da performance dos algoritmos, em 10 instâncias da TSPLIB, cada sendo executada 30 vezes com um tempo de 10 minutos.

TSPLIB		AG			AGIV			AGIV/AG	
Instâncias	x^*	μ	σ	$(\mu-x^*)/x^*$	μ	σ	$(\mu-x^*)/x^*$	μ	σ
br17.atsp	39,00	39,00	0,00	0,00	39,00	0,00	0,00	1,00	0,00
swiss42.tsp	1273,00	1381,03	50,68	0,08	1343,63	45,69	0,06	0,97	0,90
p43.atsp	5620,00	5629,37	5,72	0,00	5626,67	3,62	0,00	1,00	0,63
brazil58.tsp	23395,00	26534,30	598,26	0,13	26323,37	516,92	0,13	0,99	0,86
ftv70.atsp	1950,00	2190,30	102,79	0,12	2129,97	99,17	0,09	0,97	0,96
kro124p.atsp	36230,00	42276,17	1620,37	0,17	41563,17	1287,94	0,15	0,98	0,79
ftv170.atsp	2755,00	5899,00	309,28	1,14	4644,37	249,38	0,69	0,79	0,81
brg180.tsp	1950,00	11371,33	4276,54	4,83	3710,33	885,13	0,90	0,33	0,21
rbg323.atsp	1326,00	2592,17	39,74	0,95	2012,70	36,94	0,52	0,78	0,93
rbg403.atsp	2465,00	4267,90	37,79	0,73	3282,37	36,59	0,33	0,77	0,97

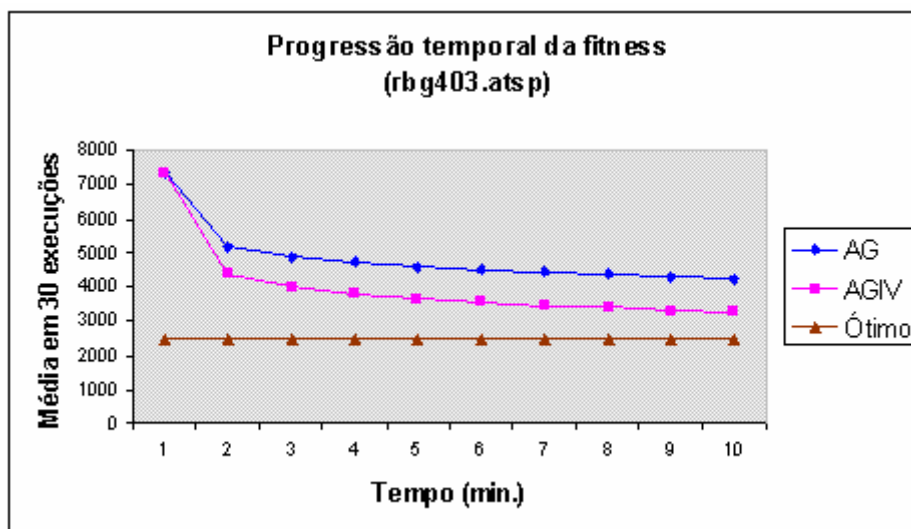


Figura 1. Gráfico comparativo da média dos resultados alcançados pelo AG padrão e com a Infecção Viral (AGIV).

Conclusões

Conforme os dados apresentados, o método proposto demonstrou uma maior eficiência, em relação ao AG convencional. Esse ganho foi obtido não só na qualidade das soluções, mas também no tempo despendido para encontrá-las. Com isso, podemos inferir que a utilização de uma população extra, contendo trechos de boa qualidade, causa a melhoria e a aceleração do Algoritmo Genético padrão.

Referências

- APPLEGATE, D.; BIXBY, R.; CHVATAL, V; COOK, W. Finding cuts in the TSP (A preliminary report). **DIMACS Technical Report** 95-05, New Brunswick, mar.1995.
- CAMPELLO, R. E.; MACULAN, N. **Algoritmos e Heurísticas**. 1. ed. Niterói. EDUFF, 1994.
- CHRISTOFIDES, N. Worst-case analysis of a new heuristic for the travelling salesman problem, **Report 388**, Pittsburgh, Graduate School of Industrial Administration, CMU, 1976.
- CROWDER, H.; PADBERG, M. Solving Large Scale Symmetric Travelling Salesman Problem to Optimality. **Management Science**, Evanston, v. 26, n. 5, p. 495-509, mai. 1980.
- DANTZIG, G. B.; FULKERSON, D. R.; JOHNSON, S. M. Solution of a Large Scale Travelling Salesman Problem. **Operations Research**, Linthicum, v. 2, p. 393-410, 1954.
- GAREY, M. R.; JOHNSON, D. S. **Computers and Intractability: a Guide to the Theory Of NP_Completeness**. 1. ed. New York: W. H. Freeman, 1979.
- GLOVER, F. Scatter Search and Path Relinking. In: CORNE, D. (Ed.); DORIGO, M. (Ed.); GLOVER, F. (Ed.). **New Ideas in Optimization**. Maidenhead: McGraw Hill, p. 297-319, 1999.
- GLOVER, F; KOCHENBERGER, G. A. **Handbook of Metaheuristic**, 1. ed. Norwell: Kluwer Academic Publishers, 2002.
- GOLDEN, B.; BODIN, L.; DOYLE, T; W. STEWART JR., W. Approximate Travelling Salesman Algorithms, **Operations Research**, Linthicum, v. 28, n. 2, p. 694-711, 1980.
- GRÖTSCHEL, M.: On the symmetric travelling salesman problem: Solution of a 120-city problem. **Mathematical Programming**. Studies 12, 61-77, 1980.
- GRÖTSCHEL, M.; HOLLAND, O.; Solution of Large Scale Symmetric Travelling Salesman Problems. **Mathematical Programming**, n. 51, p. 141-202, 1991.
- GUTIN, G. (Ed.); PUNNEN, A. P. (Ed.). **The Traveling Salesman Problem and Its Variations**. 1. ed. Boston: Springer, 2002.

- HANSEN, P.; RIBEIRO, C., **Essays and Surveys in Metaheuristics**. Boston: Springer, 2001.
- HOFFMAN, A. J.; WOLFE, P. History. In: LAWLER, E. L (Ed.); LENSTRA, J. K. (Ed.); RINNOOY KAN, A. H. G. (Ed.); SHMOYS, D. B. (Ed.). **The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization**. New York: John Wiley & Sons, 1985. p. 1-16.
- HOLLAND, J. H., **Adaptation in Natural and Artificial Systems**. Ann Arbor: University of Michigan Press, 1975.
- IBARAKI, T.; NONOBE, K.; YAGIURA, M. **Metaheuristics: Progress as real problem Solvers**. 1. ed. Boston: Springer, 2005.
- JOHNSON, D. S.; MCGEOCH, L.A. The Traveling Salesman Problem: A Case Study in Local Optimization, **Local Search in Combinatorial Optimization**, E. H. L. Aarts and J.K. Lenstra (ed), John Wiley and Sons Ltd, p. 215-310, 1997.
- KANO, H.; HASEGAWA, K.; KATO, N. Solving Constraint Satisfaction Problems by a Genetic Algorithm Adopting Viral Infection. In: INTERNATIONAL JOINT SYMPOSIA ON INTELLIGENCE AND SYSTEMS, 1996. Proceedings. Washington: IEEE Computer Society, 1996, p.67-73.
- KANO, H.; NAKAMURA, N. Route Guidance with Unspecified Posts Using Genetic Algorithm for Car Navigation Systems. In: INTELLIGENT TRANSPORTATION SYSTEMS, Dearborn, 2000. Proceedings...[S.l.]: IEEE, Out. 2000. p.119-124.
- KANO, H.; NAKAMURA, T. Knowledge Based Genetic Algorithm for Dynamic Route Selection. In: INTERNATIONAL CONFERENCE ON KNOWLEDGE-BASED INTELLIGENT SYSTEMS & ALLIED TECHNOLOGIES, 4, 2000, Brighton. Proceedings... [S.l.]: IEEE, 2000. p.616-619.
- LAPORTE, G.; ASEF-VAZIRI, A.; SRISKANDARAJAH, C. (1996), Some Applications of the generalized Travelling Salesman Problem, **Journal of Operations Research Society**, 47, 1461-1467, 1996.
- LARRAÑAGA, P.; KUIJPERS, C.M.H.; MURGA, R.H.; INZA, I.; DIZDAREVIC, S. Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators. **Artificial Intelligence Review**, v. 13, p. 129-170, 1999.
- LIN, S.; KERNIGHAN, B. An Effective Heuristic Algorithm for the Traveling Salesman Problem. **Operations Research**, n. 21, p. 498-516, 1973.
- MELAMED, I. I.; SERGEEV, S. I.; SIGAL, I. Kh. The Traveling Salesman Problem. **Surveys**. [S.l.]: Plenum Publishing Corporation, 1990. p. 1147-1173.

- PAPADIMITRIOU, C. H.; STEIGLITZ, K. **Combinatorial Optimization: Algorithms and Complexity**. 1. ed. New York: Prentice Hall, 1982.
- PADBERG, M. W.; RINALDI, G.: Optimization of a 532-city symmetric travelling salesman problem. **Oper. Res. Letters** v .6, 1-7, 1987.
- PADBERG, M. W.; RINALDI, G. A Branch-and-cut Algorithm for the Resolution of Large Scale Symmetric Traveling Salesman Problem. **SIAM Review**, n. 33, p. 60-100, 1991.
- REEVES, C. R. (Ed.). **Modern Heuristic Techniques for Combinatorial Problems**. 1. ed. Oxford: Blackwell Scientific Publication, 1993.
- ROSENKRANTZ, R.; STEARNS, R.; LEWIS, P. An Analysis of Several Heuristics for the Traveling Salesman Problem. **SIAM Journal on Computing**, Philadelphia, 6(3):563-581, 1977.

Allison da Costa Batista Guedes

Endereço eletrônico: allison@lcc.ufrn.br

Base de Pesquisa: Otimização Combinatória e Desenvolvimento de Sistemas Computacionais de Apoio à Decisão

Endereço postal : Departamento de Informática e Matemática Aplicada, Universidade Federal do Rio Grande do Norte, Campus Universitário, Natal/RN 59078-970 – Brasil