Algoritmos genéticos: alguns experimentos com os operadores de cruzamento (“Crossover”) para o problema do caixeiro viajante assimétrico

Anderson Freitas Silva (DIE/UFPI) cadfreitas@yahoo.com.br

Antonio Costa de Oliveira (DIE/UFPI) costa@ufpi.br

Resumo

Os algoritmos genéticos são métodos de busca probabilística inspirados na evolução natural. Este método heurístico mostra-se eficiente e robusto na resolução de diversos problemas de otimização combinatória. Uma das aplicações dos algoritmos genéticos em tais problemas é na resolução do Problema do Caixeiro Viajante (PCV). Um algoritmo genético para o PCV requer modificações do algoritmo genético padrão proposto por John Holland na década de 70. Codificações apropriadas para lidar com as permutações inerentes ao problema foram necessárias, e conseqüentemente operadores de cruzamentos foram desenvolvidos sobre essas codificações para manipular a população de cromossomos e garantir a geração de proles viáveis.Este artigo visa comparar, em termos de desempenho computacional e qualidade da solução obtida, os principais operadores de cruzamento desenvolvidos para o PCV.

Palavras-chave: Problema Caixeiro Viajante, Algoritmos Genéticos e Operadores de Cruzamento.

1. Introdução

O Problema do Caixeiro Viajante (PCV) é um exemplo clássico de problema de otimização combinatória. Dado um conjunto de *n* cidades e a distância (ou custo da viagem) entre elas; o problema consiste em determinar uma rota que percorra cada uma das *n* cidades uma única vez e retorne à cidade de origem, de tal forma que a distância total percorrida seja mínima.

O interesse por este problema deve-se a sua dificuldade de resolução e também, a sua enorme aplicabilidade na engenharia e nas ciências. Diversos problemas reais podem ser modelados como um PCV: roteamento de veículos, programação de tarefas em máquinas, perfuração de placas de circuitos integrados, mapeamento do DNA humano, dentre outras aplicações (LENSTRA & RINNOOY KAN, 1975) e (TSP Homepage, 2005).

Há inúmeros algoritmos exatos e heurísticos para resolver este problema. Os algoritmos exatos encontram soluções ótimas, mas sua aplicação em problemas de grande dimensão torna-se proibitiva em virtude do elevado esforço computacional. Isto se deve ao fato do PCV pertencer a classe de problemas NP-Completo (GAREY & JOHNSON, 1979).

Os algoritmos heurísticos constituem uma alternativa computacionalmente viável encontrada pelos pesquisadores para resolver o PCV. A limitação desta alternativa é a qualidade da solução obtida, que não necessariamente é a ótima. Dentre os métodos heurísticos que encontram “boas” soluções de forma eficiente estão os algoritmos genéticos.

Algoritmos genéticos são métodos de busca e otimização baseados nos princípios de seleção e evolução natural dos organismos biológicos. Foram propostos por John Holland e seus

colaboradores da Universidade de Michigan na década de 70. Inicialmente, foram utilizados no aprendizado de máquinas e otimização de funções numéricas.

O algoritmo genético canônico, idealizado por Holland, opera sobre uma população de cromossomos inicial codificados em cadeias de strings 0's e 1's. Cada cromossomo representa uma potencial solução do problema. A cada geração, os cromossomos são avaliados de acordo com uma função de adaptação, também chamada de *fitness*. Os indivíduos mais aptos (com melhores valores de adaptação) possuem maior probabilidade de reproduzir-se (seleção) mediante cruzamentos com outros indivíduos da população, produzindo descendentes com características de ambas as partes. Também é possível modificar características da população através de mutações. A seleção, o cruzamento e a mutação são operadores genéticos que transformam a população através de sucessivas gerações. Após a aplicação desses operadores, ao longo das gerações, espera-se que os indivíduos da população convirjam para uma boa solução, não necessariamente a ótima (GOLDBERG, 1989).

O operador de cruzamento e a forma de codificação do espaço de soluções (conjunto de todas as soluções possíveis) têm um papel fundamental no desempenho dos algoritmos genéticos.

Para o PCV, diversas formas de codificação e seus respectivos operadores de cruzamento foram desenvolvidos, dentre os principais operadores estão PMX (*Partially Matched Crossover*), OX (*Order Crossover*), CX (*Cicle Crossover*) e HX (*Heuristic Crossover*). O objetivo deste artigo é fazer uma análise comparativa do desempenho computacional e a qualidade da solução obtida por estes operadores, utilizando uma das formas mais naturais de codificação, a representação por caminho. Inicialmente, na seção 2, serão expostas as razões pela escolha desta codificação e em seguida, na seção 3, serão descritas as formas de codificação para o PCV. Na seção 4, será descrito o funcionamento de cada operador de cruzamento utilizado no trabalho. E finalmente, na seção 5, serão mostrados os resultados computacionais obtidos utilizando instâncias assimétricas do PCV em um grafo completo, geradas de forma aleatória ou retiradas da literatura.

2. Algoritmos genéticos para o PCV

No contexto do PCV, cada cromossomo codifica uma solução, isto é, uma rota. A função de adaptação (*fitness*), que avalia cada cromossomo, está relacionada com o tamanho da rota, o qual depende da ordem em que as cidades aparecem na mesma.

Para construir novos cromossomos (prole), o algoritmo genético deve combinar as características de duas soluções previamente selecionadas. Essas características a serem transmitidas são os arcos de cada solução.

O algoritmo canônico de Holland não é adequado para o PCV, por dois motivos:

a) Exige excessivo uso de memória, à medida que aumenta o tamanho do PCV. Supondo um problema com n cidades, cada cidade é codificada em uma seqüência de 0's e 1's com tamanho $\log_2^n$. Assim, o cromossomo inteiro terá o tamanho $n\log_2^n$.

b) O operador de cruzamento de um corte produz proles inviáveis, requerendo uma técnica auxiliar para reparar estas soluções, aumentando, assim, o esforço computacional.

Desta maneira, foram pesquisadas e desenvolvidas novas formas de codificação para o PCV, assim como, elaborados sofisticados operadores de cruzamento para manipular estas representações e criar soluções viáveis.

3. Representação genética das soluções

Para o PCV, foram propostas várias formas de codificação do espaço de busca. Entre as principais representações estão: a ordinal, por caminho (ou inteiros) e por adjacência. Além destas, há outras formas baseada em matrizes booleanas (MICHALEWICZ, 1996).

Nas subseções a seguir, será detalhada cada uma destas codificações.

3.1 Representação ordinal

Nesta forma de codificação, cada solução é representada como uma lista de *n* cidades, onde o *i-ésimo* elemento da lista é um número entre 1 e $n - i + 1$. Existe uma lista ordenada de cidades que serve como referencia para construir a representação. A lista ordenada *l*, varia de 1 a *n* cidades. O cromossomo é um vetor de posicionamento da cidade na lista. Por exemplo, seja o cromossomo $p = (1, 1, 2, 3, 1, 1)$ e a lista $l = (1, 2, 3, 4, 5, 6)$.

O primeiro elemento do cromossomo *p* é 1. Ele corresponde ao primeiro elemento da lista *l* que é o 1 (Rota: 1). O próximo elemento do cromossomo *p* é 1. Ele corresponde ao primeiro elemento da lista *l* atualizada (após remoção do 1) que é o 2 (Rota: 1 → 2). O próximo número do cromossomo *p* é 2. Tal número corresponde ao segundo elemento da lista *l* atualizada (sem o 1 e o 2) que é o 4 (Rota: 1 → 2 → 4). O próximo elemento do cromossomo é 3, correspondendo ao 6 na lista *l* atualizada (Rota: 1 → 2 → 4 → 6). Seguindo este raciocínio, chega-se à rota final: 1 → 2 → 4 → 6 → 3 → 5. Essa é a solução representada por este cromossomo.

3.2 Representação por caminho

Esta representação é talvez a mais natural de uma solução para o PCV. O cromossomo é formado pela seqüência dos nós na solução. Por exemplo, o cromossomo $p = (1, 4, 2, 3, 6, 5)$ representa diretamente a solução 1 → 4 → 2 → 3 → 6 → 5.

3.3 Representação por adjacência

Cada cromossomo é representado como uma lista de *n* cidades. A cidade *j* da rota é listada na posição *i* se, e somente se, a rota vai da cidade *i* a cidade *j*. Por exemplo, um cromossomo $p = (3, 1, 5, 6, 4, 2)$ apresenta os seguintes arcos em uma rota associado a *p*: 1→ 3; 2 → 1; 3 → 5; 4 → 6; 5 → 4; 6 → 2. Assim, o cromossomo representa a seguinte solução: 1 → 3 → 5 → 4 → 6 → 2.

4. Operadores de cruzamento

Existem diversos operadores de cruzamento desenvolvidos para manipular cada uma das representações descritas acima. Cada representação possui seu próprio conjunto de operadores. Neste artigo, foram selecionados os principais operadores para a codificação por caminho. Eles são classificados, de acordo com Potvin (1996), em: operadores que preservam a posição absoluta das cidades e operadores que preservam a ordem relativa.

Dentre os operadores de cruzamento que mantêm a posição absoluta estão: o PMX (*Partially Matched Crossover*) proposto por Goldberg e Lingle (1985 apud POTVIN, 1996, p. 351) e o CX (*Cycle Crossover*) proposto por Oliver (1987 apud POTVIN, 1996, p. 353). E entre os operadores que mantêm a ordem relativa está o OX (*Order Crossover*) desenvolvido por Davis (1985 apud MICHALEWICZ, 1996, p. 217).

Também, foi utilizada para o estudo comparativo uma versão adaptada do HX (*Heuristic Crossover*) que utiliza representação por caminho. Além destes operadores, existem outros que foram desenvolvidos para outras formas de codificação, como o operador ER (*Edge*

Recombination) para a representação por adjacência e o operador de um corte para a representação ordinal, os quais são citados por Michalewicz (1996) e Potvin (1996).

A seguir, será descrito como cada um destes operadores manipula uma solução do PCV.

4.1 Operador PMX

Ele é descrito pelo seguinte procedimento: considere um par de cromossomos p_1 e p_2 (figura 1), e sobre eles realizam-se dois pontos de corte aleatórios. Os filhos f_1 e f_2 herdam integralmente as seqüências parciais entre os dois pontos de corte respectivamente de p_2 e p_1, ou seja, há uma troca de subseqüências. Para isso, deve-se preservar a ordem e a posição de cada cidade nas subseqüências.

$$\left.\begin{array}{l} p_1 = (1, 2, \mid 3, 4, 5, \mid 6) \\ p_2 = (6, 3, \mid 1, 4, 5, \mid 2) \end{array}\right\} \rightarrow \left\{\begin{array}{l} f_1 = (3, 2, \mid 1, 4, 5, \mid 6) \\ f_2 = (6, 1, \mid 3, 4, 5, \mid 2) \end{array}\right.$$

Figura 1 – Cruzamento aplicando operador PMX

Esta troca também define o seguinte mapeamento: 1 ↔ 3, 4 ↔ 4, 5 ↔ 5, para reparar rotas inviáveis. Cada gene de f_1, ainda não conhecido após a colocação das seqüências parciais, é preenchido a partir de seu pai p_1, e cada gene de f_2 não conhecido é preenchido a partir de seu pai p_2, desde que não forme uma rota inviável.

Observa-se que se o filho f_1 herdar a cidade '1' (já presente em sua seqüência parcial) do cromossomo p_1 cria-se uma rota inviável. Assim pelo mapeamento definido anteriormente, o filho f_1 herda a cidade '3'. Da mesma forma, o filho f_2 herda a cidade '1' ao invés da cidade '3' de p_2.

4.2 Operador CX

Este operador gera filhos que preservam a posição absoluta das cidades provenientes dos cromossomos pais. Ele trabalha sobre um subconjunto de cidades que ocupa um mesmo subconjunto de posições em ambos os pais. Estas cidades são copiadas de um dos pais (por exemplo p_1), nas mesmas posições, para um filho (f_1). As posições remanescentes são completadas com as cidades do outro pai (p_2). Assim, cada cidade e sua posição são herdadas de um dos pais. Veja figura 2:

$$\begin{array}{l} p_1 = (\mathbf{3}, \mathbf{2}, \mathbf{1}, \mathbf{4}, \mathbf{5}, \mathbf{6}) \\ \underline{p_2 = (6, 3, 4, 1, 5, 2)} \\ f_1 = (\mathbf{3}, \mathbf{2}, 4, 1, 5, \mathbf{6}) \end{array}$$

Figura 2 – Cruzamento aplicando operador CX

4.3 Operador OX

Este operador gera filhos a partir da escolha de uma seqüência parcial de cidades de um dos pais e preservando a ordem relativa das cidades do outro pai. Observe a figura 3:

$$\left.\begin{array}{l} p_1 = (1, 6, \mid 3, 2, 5, \mid 4) \\ p_2 = (6, 2, \mid 1, 4, 3, \mid 5) \end{array}\right\} \rightarrow \left\{\begin{array}{l} f_1 = (1, 4, \mid 3, 2, 5, \mid 6) \\ f_2 = (2, 5, \mid 1, 4, 3, \mid 6) \end{array}\right.$$

Figura 3 – Cruzamento aplicando operador OX

Os filhos f_1 e f_2 herdam as faixas entre os cortes de seus respectivos pais p_1 e p_2. A seguir, partindo do segundo corte de um pai (por exemplo p_2), copia-se as cidades na mesma ordem em que aparecem, removendo aquelas contidas entre os dois cortes do outro pai (no caso p_1). Assim, a partir da seqüência 5, 6, 2, 1, 4 e 3, obtêm-se a seqüência 6, 1, 4 (removendo 3, 2 e 5 já presentes) a ser inserida em f_1 a partir de seu segundo ponto de corte. Da mesma maneira, a partir da seqüência 4, 1, 6, 3, 2 e 5 de p_1, têm-se a seqüência 6, 2 e 5 (removendo 1, 4 e 3 de p_2) a ser inserida em f_2 a partir de seu segundo corte . Dessa forma, obtêm-se os filhos ilustrados na figura acima.

4.4 Operador HX

Este operador utiliza uma informação heurística não explorada pelos operadores anteriores, a distância entre as cidades, isto é, o tamanho dos arcos.

O operador HX pode ser descrito da seguinte forma:

a) Escolha uma cidade inicial aleatória de um dos pais;

b) Compare os arcos deixando a atual cidade em ambos os pais e selecione o menor;

c) Se o menor arco escolhido formar um ciclo na rota parcial, então escolha um arco aleatório que não introduza um ciclo;

d) Repita os passos “b” e “c” até que todas cidades sejam incluídas na rota.

5. Testes e resultados computacionais

Um algoritmo genético para o PCV foi implementado em linguagem C++. Os testes computacionais foram realizados em um microcomputador com processador Intel® Pentium® IV 3.0 GHz e 512 MB de RAM.

Para realizar a comparação dos operadores explanados nas seções anteriores, foram resolvidos 9 (nove) PCV’s assimétricos. Dentre estes, 5 (cinco) foram gerados de forma aleatória com instâncias de 100, 200, 300, 400 e 500 cidades. E os 4 (quatro) problemas restantes foram retirados do TSPLIB (1995), com as seguintes instâncias: 17, 71, 171 e 443 cidades. Cada problema foi resolvido 5 (cinco) vezes para cada operador, obtendo-se a solução média e o tempo computacional médio das execuções, assim como o registro da melhor solução encontrada por cada operador.

Os valores utilizados na matriz representativa das distâncias entre as cidades, para os problemas gerados aleatoriamente, pertencem ao intervalo inteiro [0,1000].

No algoritmo implementado, o operador de seleção utilizado foi o método da roleta, no qual os indivíduos a serem selecionados são representados em uma roleta proporcionalmente ao seu *fitness*, para posteriormente serem submetidos ao operador de cruzamento, de acordo com uma taxa de cruzamento (GOLDBERG, 1989). Esta taxa indica a probabilidade de aplicação do operador de cruzamento sobre duas soluções selecionadas na roleta. De acordo com Coley (1999) e Goldberg (1989), o valor desta probabilidade está compreendido entre 0,4 e 0,95. No algoritmo, a taxa foi fixada em 0,85 para todos os operadores utilizados.

Já o operador de mutação implementado foi o de inversão. Dois pontos no cromossomo são selecionados aleatoriamente e a substring entre estes pontos é invertida (MICHALEWICZ, 1996). Este operador foi aplicado para cada cromossomo da prole com uma dada probabilidade também fixada, no valor de 0,02.

Outros parâmetros importantes de um algoritmo genético também foram fixados. O tamanho da população foi limitado a 100 cromossomos. E número máximo de iterações (critério de parada do algoritmo) foi 150000 gerações. Além disso, o algoritmo implementado utiliza uma estratégia elitista, de maneira que a melhor solução da geração atual seja transferida automaticamente para a geração seguinte.

A tabela 1 resume os resultados computacionais obtidos para os 9 (nove) PCV's propostos.

Nº de Cidades	PMX		OX		CX		HX		Melhor Solução Conhecida
	Média	Melhor Solução	Média	Melhor Solução	Média	Melhor Solução	Média	Melhor Solução	
17	39	39	39	39	39	39	39	39	39
71	2660,2	2401	2161,6	2012	3008,6	2816	2152,8	2111	1950
100	9606,2	9091	2967,6	2619	8616,6	8223	2285,8	2239	-
171	5680,6	5496	5764,8	5417	6407,6	6002	3341,2	3180	2755
200	20242,3	19028	9631,2	8810	15152,6	13655	2776,8	2689	-
300	32150,4	30261	21288,6	20286	21933,6	21293	3676	3494	-
400	44884,2	42091	35352,2	33982	31868,6	30260	3959,4	3708	-
443	4578,2	4493	4726,8	4655	3276,8	3226	3659,8	3608	2720
500	56285	54124	52646	50517	40084,2	36913	4412	4106	-

Tabela 1 – Comparação de quatro operadores de cruzamento

A partir da tabela acima, observa-se que o operador de cruzamento HX destaca-se sobre os demais operadores, sobretudo quando o número de cidades aumenta. Uma exceção foi o problema com 443 cidades, onde o CX foi melhor.

Analisando os resultados obtidos para os problemas com a melhor solução conhecida, observa-se que, à medida que o número de cidades cresce, a qualidade da melhor solução obtida para cada problema distancia-se da melhor solução conhecida. Para o problema com 17 cidades, todos os operadores encontraram a melhor solução conhecida. Já no problema com 71 cidades, a melhor solução, obtida pelo operador OX, ficou por volta de 3,1 % da melhor solução conhecida. No problema com 171 cidades, o percentual foi de 15,4 %, que foi obtido pelo operador HX. E no problema com 443 cidades, a melhor solução, obtida pelo operador CX, foi 18,6 % da melhor solução conhecida.

As figuras 4 e 5 mostram a evolução do *fitness* médio para alguns experimentos realizados com relação aos 4 (quatro) operadores implementados e duas diferentes instâncias de cidades.

Para o PCV com 171 cidades (figura 4), o operador HX obteve uma solução de melhor qualidade em relação aos demais operadores.

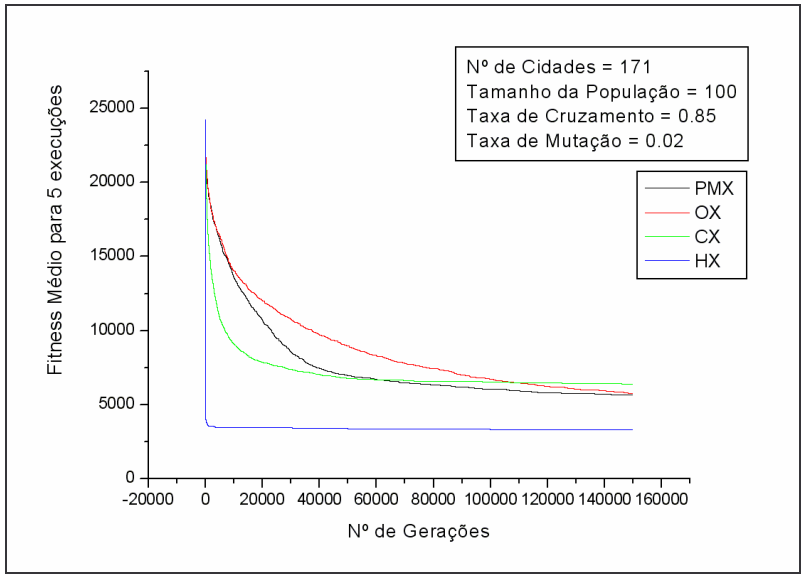

Figura 4 - Evolução do *fitness* médio da melhor solução para os diferentes operadores de cruzamento num PCV assimétrico com 171 cidades

Já, com relação ao PCV com 500 cidades (figura 5), o operador HX também obteve uma solução de melhor qualidade que os demais operadores.

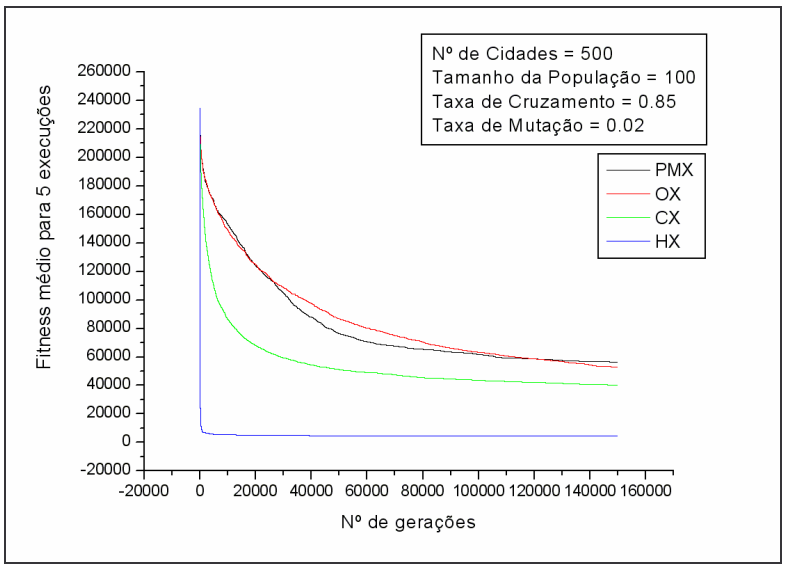

Figura 5 - Evolução do *fitness* médio da melhor solução para os diferentes operadores de cruzamento num PCV assimétrico com 500 cidades

De acordo com o gráfico da figura 6, observa-se que, para os PCV's com tamanho até 100 cidades, todos os operadores têm um tempo médio computacional semelhante. Mas, para os PCV's com tamanho maior que 100 cidades, o tempo computacional do operador HX destaca-se em relação aos demais. Ele obtém soluções melhores a um esforço computacional um pouco maior. Em contrapartida, o operador CX tem um desempenho quase linear, exigindo pouco tempo de execução. Os outros operadores PMX e OX têm um desempenho intermediário, entre o CX e o HX.

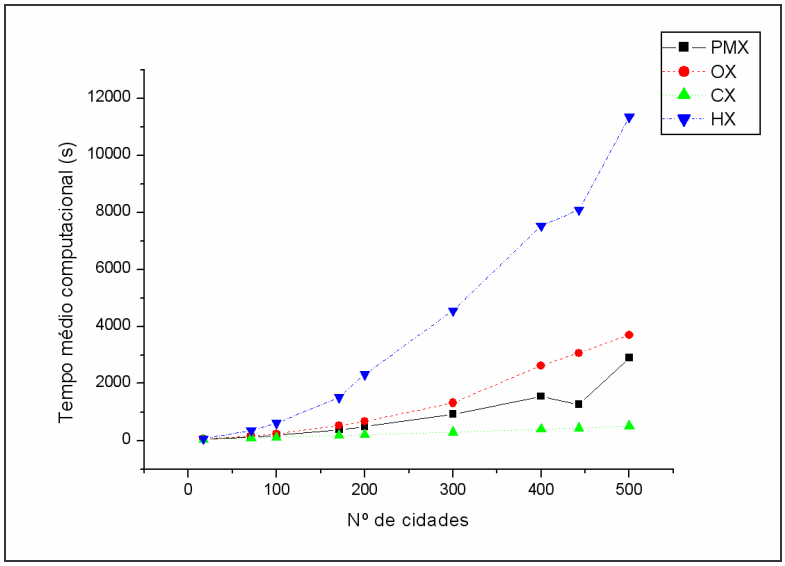

Figura 6 – Tempo computacional médio dos operadores de cruzamento para diferentes instâncias do PCV

6. Conclusões

Neste trabalho, foram realizados alguns experimentos com os operadores de cruzamento implementados (PMX, OX, CX e HX), utilizando representação por caminho. Estes operadores obtiveram "boas" soluções, para os problemas gerados aleatoriamente e retirados da literatura, em pouco tempo de execução, demonstrando eficiência para lidar com um problema NP-Completo como o PCV (GAREY & JOHNSON, 1979).

Com relação à comparação de desempenho entre os operadores, o HX foi o operador que apresentou melhor desempenho na qualidade da solução, considerando a totalidade dos resultados. Isto ocorre, talvez, pelo fato do operador utilizar uma informação heurística não explorada pelos operadores anteriores, a distância entre as cidades, como dito em seções anteriores.

Já o CX foi o operador que exigiu o menor tempo computacional em relação aos outros operadores. Isto se deve ao fato da aplicação do operador, sobre duas soluções previamente selecionadas, não exigir mapeamentos e muitas verificações de viabilidade na prole obtida, como ocorre no PMX, no OX e no HX.

Considerando somente o conjunto de operadores PMX, CX e OX, observou-se um maior distanciamento das melhores soluções obtidas por estes operadores em relação à solução obtida pelo HX, para PCV's de tamanho maior que 300 cidades. Alguns fatores podem ter influenciado no desempenho desses operadores, como os valores fixados dos parâmetros utilizados (tamanho da população, taxas de cruzamento e mutação) e o operador de mutação. Isto requer um maior aprofundamento sobre a influência desses parâmetros para os operadores PMX, CX e OX.

7. Trabalhos futuros

Como trabalhos futuros, sugere-se uma análise comparativa destes mesmos operadores de cruzamento (PMX, OX, CX e HX) para PCV's simétricos e em um grafo qualquer.

Também, esta análise pode ser estendida, abrangendo outros operadores de cruzamento, utilizando outras formas de codificação.

Outra sugestão é a inserção de busca local na prole, a fim de melhorar os resultados obtidos. Existem diversos algoritmos híbridos (algoritmos genéticos com busca local) que melhoram a eficiência dos algoritmos genéticos sem comprometer sua flexibilidade (KENNEDY, 1993).

8. Referências

COLEY, David A. *An Introduction to Genetic Algorithms for Scientists and Engineers.* World Scientific Pub Co, Singapure, 1999.

GAREY, M. R. & JOHNSON, D. S. *Computers and Intractability: a guide to the theory of NP-Completeness.* San Francisco, Freeman, 1979.

GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, Reading, MA, 1989.

KENNEDY, Scott A. *Five Ways to a Smarter Genetic Algorithm.* AI Expert. December 1993.

LENSTRA, J. K. & RINNOOY KAN, A. H. G. *Some Simple Applications of the Traveling Salesman Problem.* Operational Research Quarterly, v. 26, n. 4, p. 717-733, 1975.

MICHALEWICZ, Z. *Genetic Algorithms + Data Structures = Evolution Programs.* 3rd edition. Springer-Verlag. Berlin, Germany (1996).

POTVIN, J. Y. *Genetic algorithms for the traveling salesman problem.* Annals of Operations Research 6, p. 339-370, 1996.

TSP HomePage, 2005. *TSP Applications.* Janeiro 2005. Disponível em: http://www.tsp.gatech.edu/apps/index.html. Acesso em: 06/05/2006.

TSPLIB, 1995. *Traveling Salesman Problem Library.* Novembro 2004. Disponível em: http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95. Acesso em: 06/05/2006.