colloquium exactarum

ESTUDO COMPARATIVO DO DESEMPENHO DE DIFERENTES OPERADORES GENÉTICOS NA RESOLUÇÃO DO PROBLEMA DO CAIXEIRO VIAJANTE

Comparative Study of Different Genetic Operators to the Resolution of the Traveling Salesman Problem

Fábio Portela da Silva[1;] Almir Olivette Artero[1;] Marco Antônio Piteri[1;] Francisco Assis Silva[2;] Danillo Roberto Pereira[2]

Faculdade de Ciências e Tecnologia, FCT/UNESP[1]; Faculdade de Informática de Presidente Prudente, Fipp/Unoeste[2;]
fabio_ps84@yahoo.com.br; almir@fct.unesp.br; piteri@fct.unesp.br; chico@unoeste.br; danilopereira@unoeste.br

RESUMO - Este trabalho apresenta um estudo comparativo do desempenho de diferentes combinações de operadores genéticos comumente utilizados na resolução do Problema do Caixeiro Viajante. A avaliação considerou somente os custos obtidos pelas combinações e foi realizada com seis instâncias da versão simétrica do problema. Os testes foram feitos com o uso do elitismo para todas as possíveis combinações. Os resultados obtidos mostraram que, individualmente, os operadores Torneio (seleção), CX (Cruzamento) e Inversão (Mutação) foram os mais eficientes.
Palavras-chave: algoritmos genéticos; inteligência artificial; otimização, problema do caixeiro viajante.

ABSTRACT - This work proposes a comparative study of genetic operators commonly used to find a solution for the Traveling Salesman Problem. It was considered only the costs obtained by the combinations and six different instances of the symmetric version of the problem were used. The experiments were performed with elitism for all the possible combinations. The results demonstrated that, individually, the Tournament (selection), CX (crossover) and Inversion (mutation) operators had the best results.
Keywords: genetic operators; artificial intelligence; optimization, the salesman problem.

Recebido em: 16/06/2016
Revisado em: 13/12/2016
Aprovado em: 11/02/2017

1. INTRODUÇÃO

O Problema do Caixeiro Viajante (PCV) (PAPADIMITRIOU, 1977), classificado como NP-Difícil, figura como um dos problemas mais conhecidos e estudados na Ciência da Computação. Consiste em encontrar um ciclo Hamiltoniano de mais baixo custo em um grafo completo não direcionado. Em outras palavras, resume-se em encontrar um percurso que visite cada vértice do grafo, uma única vez, de maneira que o custo total do percurso seja mínimo. No grafo, os vértices representam as cidades e os pesos, associados às arestas, são as distâncias separando as cidades. A versão do PCV avaliada neste trabalho caracteriza-se por apresentar o mesmo custo entre os vértices, ou cidades, independentemente do sentido escolhido para percorrê-las, ou seja, é a versão simétrica do problema (APPLEGATE, 2006). Porém, vale lembrar que há também a versão assimétrica que apresenta custos distintos entre os vértices, dependendo do sentido tomado na aresta. O PCV é bastante estudado por estar relacionado a problemas práticos como a identificação de itinerários, rotas para pacotes de dados transmitidos por roteadores, perfuração de placas de circuito impresso, entre outros (CALADO; LADEIRA, 2011). Trata-se de um problema para o qual não há um algoritmo conhecido capaz de resolvê-lo em um intervalo de tempo polinomial (LAWYER et al., 1985). Assim, métodos de força bruta, de um modo geral, são considerados impraticáveis para problemas dessa natureza, levando os pesquisadores a buscar formas alternativas para resolvê-lo.

O Algoritmo Genético (AG), cujo desenvolvimento foi baseado na teoria da evolução de Darwin (GOLDBERG, 1989) e na genética de Mendel (MICHALEWICZ, 1996), se apresenta como uma alternativa que simula um ambiente de competição entre possíveis soluções para o problema. A partir de uma população com soluções iniciais aleatórias, operadores de seleção, cruzamento e mutação são aplicados, para gerar populações com níveis superiores de aptidão (MAN; TANG; KWONG, 1999).

Este artigo apresenta os resultados de um estudo realizado com vários operadores de seleção (Roleta, Torneio e Ranking Linear), cruzamento (PMX, OX e CX) e mutação (Inversão, Substituição e Troca) na resolução de diferentes instâncias do PCV. Para cada uma das etapas do AG (Seleção-Cruzamento-Mutação) foram propostas várias combinações, gerando 27 configurações, além de uma combinação aleatória, ou seja, que em cada execução, escolhia um operador diferente para cada etapa do processo. Durante os experimentos foram avaliadas as qualidades das rotas encontradas em cada uma das combinações, além dos custos

computacionais. Os resultados obtidos mostraram que as melhores combinações de operadores conseguem em torno de 60% de redução no tamanho das rotas, merecendo destaque o operador de seleção torneio, que obtém rotas mais curtas e também apresenta um custo computacional menor e o operador de mutação inversão, presente nas melhores soluções.

As demais seções deste artigo estão organizadas da seguinte maneira: Na Seção 2 são apresentados alguns trabalhos relacionados ao tema desta pesquisa. Na Seção 3 é introduzida a definição formal do PCV, sendo apresentadas suas principais características. Na Seção 4 apresenta o algoritmo genético e descreve seu funcionamento. Na Seção 5 é descrita a técnica do elitismo, avaliada neste trabalho. Na seção 6, é detalhada a metodologia adotada, especificando os materiais utilizados. A Seção 7 apresenta os resultados dos experimentos realizados. Por fim, a Seção 8 apresenta as conclusões e trabalhos futuros.

2. TRABALHOS RELACIONADOS

Há diversas pesquisas relacionadas ao estudo de algoritmos genéticos para a resolução do PCV, a fim de comparar determinados operadores genéticos. Em Abdoun e Abouchabaka (2011), por exemplo, é detalhado um estudo em que seis operadores de cruzamento são implementados e testados para solucionar a variante mais simples do PCV. Neste trabalho, os resultados mostraram que o operador OX foi capaz de encontrar as melhores soluções para o problema.

Em Silva e Oliveira (2006), é descrito um estudo que visa comparar operadores de cruzamento para a variante assimétrica do PCV. Esta pesquisa mostrou que o operador Hybrid Crossover (HX) foi o melhor em termos de qualidade da solução, enquanto que o CX foi o que apresentou menor tempo de processamento.

Castro e Masutti (2009), por sua vez, apresentam um estudo comparativo de duas classes de algoritmos evolucionários aplicados ao PCV. Uma delas é representada por um algoritmo genético, no qual foi implementado um método de seleção elitista, dois tipos de operadores de cruzamento e cinco operadores de mutação. A outra classe é representada por um algoritmo inspirado em sistemas imunológicos de seres vertebrados. Os resultados mostraram que o algoritmo imunológico gerou soluções mais próximas da solução ótima, porém, o tempo de execução do algoritmo genético mostrou-se menor.

Em Yamakami e Carvalho (2008), é descrita uma pesquisa em que foi implementado um algoritmo híbrido para a solução do PCV simétrico. Neste

procedimento, meta-heurísticas de colônia de formigas e um algoritmo genético são combinados a fim de alcançar uma melhora na busca, na qualidade das soluções encontradas e na sua convergência. Os resultados mostraram que esta combinação de algoritmos apresentou uma média de soluções ótimas, bem próxima do ideal, apesar do tempo computacional ter sido muito alto.

Em Calado e Ladeira (2011), é realizado um estudo comparativo envolvendo um algoritmo genético, redes neurais e um algoritmo heurístico na solução de três diferentes instâncias do PCV, selecionadas da base TSPLIB (REINELT, 1995). Os resultados mostraram que o algoritmo genético implementado apresentou melhor desempenho.

No trabalho de Razali e Geraghty (2011), foi desenvolvido um algoritmo genético, utilizando o MATLAB, a fim de comparar o desempenho dos operadores de seleção Roleta, Torneio e Ranking Linear. Como operadores de cruzamento e mutação, foram implementados os operadores OX e Inversão, respectivamente. Os resultados mostraram que a seleção por Ranking foi a que apresentou as melhores soluções, seguida pelos operadores Torneio e Roleta. Contudo, o Torneio mostrou-se mais rápido que os outros.

3. PROBLEMA DO CAIXEIRO VIAJANTE

A variante do PCV (simétrica), abordada neste trabalho, consiste em encontrar o trajeto de menor custo em um grafo totalmente conectado não direcionado $G(V, E)$, em que cada vértice c do conjunto $V = 1,...,v$ é uma cidade, e cada aresta a pertencente ao conjunto $E = 1,...,e$, uma conexão entre duas cidades (GUTIN; PUNNEN, 2002). Para cada aresta é atribuído um custo $d(ij)$ (peso), onde i e j são as cidades interligadas pela aresta. O objetivo é minimizar a somatória dos custos associados às arestas percorridas, ou seja, tem como função objetivo em 1.

$$Minc = \sum_{(i,j)\in R} d_{ij} \qquad (1)$$

sendo R o subconjunto das arestas (i, j) que compõem a solução, pertencente ao conjunto E (VIEIRA, 2006). O número de possíveis soluções presentes no espaço de busca do PCV, considerando o vértice de partida como aleatório, é $v!$, sendo v o número de vértices (cidades) que devem ser visitados (APPLEGATE, 2006). Percebe-se que o aumento de v, mesmo que em algumas unidades, é capaz de ampliar consideravelmente as possibilidades de solução para o problema.

Diversas restrições podem ser aplicadas ao PCV, com variações envolvendo limites de carga, horário, tempo, distância e várias outras. O Problema do Roteamento de

Veículos Capacitados (PRVC) é um exemplo destas variações, em que há *n* clientes com demandas para serem atendidas por m veículos com certo limite de carga (NEIA et al., 2013). Outra é a variante conhecida como Problema do Caixeiro Viajante com Demandas Heterogêneas (PCVDH) em que há um custo constante entre as cidades, e um custo de natureza variável cuja base de cálculo é a demanda dos produtos que devem ser entregues (VIEIRA, 2006).

4. ALGORITMOS GENÉTICOS

Os algoritmos genéticos pertencem ao ramo da Computação evolutiva. São técnicas de busca que apresentam elevada capacidade de encontrar boas soluções para um problema e, ao mesmo tempo, evitar situações de estagnação em ótimos locais (ARTERO, 2009). Trata-se de um algoritmo capaz de explorar espaços de busca contendo um variado conjunto de soluções possíveis. Um algoritmo genético típico consegue, por meio de seus operadores, produzir sucessivas gerações de soluções candidatas e, eventualmente, encontrar uma que possa ser considerada ótima. A escolha pela técnica em questão se justifica pela sua importância no meio científico, que tem se mostrado crescente ao longo dos últimos anos com diversos livros e artigos publicados a respeito.

A versão mais simples do algoritmo genético é composta dos seguintes passos:

1. *Geração de uma população inicial aleatória;*
2. *Cálculo da aptidão de cada cromossomo;*
3. *Seleção de cromossomos de acordo com o grau de aptidão apresentado;*
4. *Cruzamento dos cromossomos;*
5. *Mutação da prole gerada;*
6. *Repetição dos passos 2 à 5, até que a condição de parada ser satisfeita;*

O primeiro passo consiste em gerar uma população inicial, que pode ser criada a partir de alguma informação que se tenha sobre o problema a ser resolvido ou, então, produzida de forma aleatória. No caso da solução do Problema do Caixeiro Viajante, é preciso gerar soluções que sejam coerentes, ou seja, que contenham todos os nós a serem visitados, exatamente uma vez.

Os indivíduos que compõem a população inicial são soluções em potencial para o problema. Após esta primeira etapa, o grau de aptidão, ou fitness, é calculado para cada um deles. Se dentre os cromossomos da geração atual não houver algum que possa ser considerado ótimo, uma nova geração deverá ser criada. A criação da nova geração é feita por etapas, e procura eliminar cromossomos com características indesejadas, ao mesmo tempo que se tenta preservar aqueles que possam ser considerados promissores. O AG busca simular o que ocorre em um ambiente

natural, no qual os seres vivos competem entre si para sobreviver. Assim, como ocorre na natureza, os mais adaptados são naturalmente selecionados para o cruzamento, podendo ocorrer mutações nos descendentes. No algoritmo, os operadores genéticos são os responsáveis por executar as tarefas de seleção, cruzamento e mutação dos cromossomos. A seleção é realizada de acordo com o grau de aptidão calculado anteriormente. Os cromossomos mais aptos terão uma maior probabilidade de serem selecionados, transmitindo seu "código genético" para a próxima geração. O cruzamento consiste em uma troca de genes entre pares de cromossomos que foram selecionados na etapa anterior. A prole gerada pode, ainda, passar por uma mutação. No algoritmo genético clássico, probabilidades de cruzamento e mutação podem ser associadas aos cromossomos. Finalmente, a nova geração substitui a população original e as aptidões dos cromossomos são calculadas. Verifica-se, então, se a condição de parada foi satisfeita. Caso ainda não tenha sido, o algoritmo produzirá uma nova geração passando por todas as etapas descritas anteriormente;

5. ELITISMO

Trata-se de uma técnica complementar para os métodos de seleção, que procura garantir a permanência dos melhores cromossomos nas gerações seguintes (MITCHELL, 1999). Ela tem por objetivo melhorar o desempenho do algoritmo ao tentar impedir a perda de cromossomos de boa aptidão. É fixado um valor *k*, que representa uma porcentagem da população, como sendo a quantidade dos indivíduos mais bem adaptados que surgiram na geração *t*, e que devem ser passados para a geração *t*+1 (LINDEN, 2012). Com isto, consegue-se garantir que seus genes não serão extintos com o término da sua geração, o que poderia levar a uma queda no nível geral da aptidão da população de cromossomos. Esta técnica foi implementada e executada em conjunto com as combinações avaliadas em todas as instâncias selecionadas.

6. METODOLOGIA

Para os experimentos, foram selecionadas seis instâncias simétricas para os testes, com quatro delas selecionadas a partir da biblioteca TSPLIB, que são: *Bays29*, *Swiss42*, *Berlin52* e *eil76*, com 29, 42, 52 e 76 vértices respectivamente. Outras duas, com 14 e 60 vértices, foram geradas de forma randômica pela própria aplicação desenvolvida para este trabalho. Todos os pesos associados às arestas são valores inteiros. As combinações de operadores genéticos estão relacionadas na Tabela 1. Além delas, também foi testada uma forma

aleatória de combinação para a qual, a cada iteração, é escolhido um operador diferente para seleção, cruzamento e mutação. A implementação foi feita usando a linguagem Java e, assim como as instâncias usadas nos experimentos, está disponível para download[1]. Inicialmente, foram realizadas 30 execuções para todas as combinações de operadores genéticos ao longo de 1.000 gerações, com uma população de 100 cromossomos e taxa de mutação de 10%. Os testes foram realizados com elitismo fixo em 10%. Em seguida, as cinco melhores combinações foram selecionadas para um rápido comparativo com o ótimo conhecido das instâncias do TSPLIB, assim como uma breve análise da variância das soluções para instância *Bays29*. Logo depois, ainda com a instância *Bays29*, e selecionando-se as três melhores combinações, testes com diferentes tamanhos de população e diferentes valores de elitismo foram realizados a fim de avaliar o impacto que a variação destes parâmetros pode ter nas soluções. Por fim, com base nos resultados obtidos, foi feita uma avaliação dos operadores individualmente. Ao todo, foram 12240 execuções com os parâmetros descritos acima em uma máquina com processador Intel Core 2 de 2.4 GHz, 4GB de RAM e sistema operacional OpenSuse (Linux).

Tabela 1. Combinações de operadores avaliadas neste trabalho.

ID	Seleção/Cruzamento/Mutação
C1	Roleta, PMX, Inversão
C2	Roleta, PMX, Substituição
C3	Roleta, PMX, Troca
C4	Torneio, PMX, Inversão
C5	Torneio, PMX, Substituição
C6	Torneio, PMX, Troca
C7	Ranking Linear, PMX, Inversão
C8	Ranking Linear, PMX, Substituição
C9	Ranking Linear, PMX, Troca
C10	Roleta, OX, Inversão
C11	Roleta, OX, Substituição
C12	Roleta, OX, Troca
C13	Torneio, OX, Inversão
C14	Torneio, OX, Substituição
C15	Torneio, OX, Troca
C16	Ranking Linear, OX, Inversão
C17	Ranking Linear, OX, Substituição
C18	Ranking Linear, OX, Troca
C19	Roleta, CX, Inversão
C20	Roleta, CX, Substituição
C21	Roleta, CX, Troca
C22	Torneio, CX, Inversão
C23	Torneio, CX, Substituição
C24	Torneio, CX, Troca
C25	Ranking Linear, CX, Inversão
C26	Ranking Linear, CX, Substituição
C27	Ranking Linear, CX, Troca
C28	Aleatório

[1] www.dropbox.com/s/q4t32v3z zrr27uq/GA.tar.gz?dl=0

7. EXPERIMENTOS E RESULTADOS

A Tabela 2 apresenta os custos, em unidades de comprimento, das rotas obtidas com todas as combinações para as instâncias com menor número de vértices (instâncias com 14, 29 e 42 vértices), além dos tempos de processamento, em segundos, de cada uma delas. As dez melhores combinações foram C1, C4, C10, C11, C12 C13, C19, C22, C25 e C28. Estas combinações possuem a predominância do operador Roleta na etapa de seleção, do operador OX na etapa de cruzamento e do operador inversão na mutação.

Tabela 2: Custo médio das rotas e tempos de processamento.

Instância 14			
Pos	Comb	Custo da rota	Tempo de Proces.
1	C1	6.023,0	33.4
2	C10	6.023,0	33.5
3	C13	6.023,0	33.4
4	C28	6.023,0	18.3
5	C12	6.061,2	18.1
6	C11	6.061,5	18.2
7	C19	6.083,3	21.6
8	C22	6.083,3	21.6
9	C25	6.083,3	22.1
10	C4	6.113,4	34.7
11	C2	6.121,5	34.1
12	C15	6.182,5	33.9
13	C14	6.312,3	18.3
14	C26	6.360,4	18.0
15	C16	6.397,4	18.0
16	C5	6.406,4	21.2
17	C7	6.408,7	21.3
18	C8	6.466,3	21.2
19	C20	6.476,0	33.6
20	C17	6.507,7	34.0
21	C18	6.515,3	35.8
22	C9	6.667,6	18.2
23	C27	6.736,3	18.1
24	C3	6.773,3	18.1
25	C21	6.798,3	21.3
26	C23	7.284,7	21.3
27	C24	7.343,6	21.3
28	C6	7.362,8	25.4

Instância 29			
Pos	Comb	Custo da rota	Tempo de Proces.
1	C1	2.298,3	49.4
2	C13	2.318,3	51.0
3	C25	2.329,6	50.8
4	C4	2.330,6	22.1
5	C19	2.346,3	22.3
6	C22	2.350,0	22.2
7	C28	2.350,4	24.6
8	C10	2.355,9	24.6
9	C12	2.434,2	24.4
10	C11	2.435,0	50.8
11	C15	2.496,9	50.7
12	C2	2.564,4	50.6
13	C20	2.569,9	20.7
14	C14	2.582,8	21.2
15	C5	2.646,9	21.0
16	C26	2.759,4	22.5
17	C23	2.827,9	22.4
18	C27	3.029,2	22.4
19	C21	3.034,1	49.3
20	C3	3.036,6	48.9
21	C24	3.065,6	48.8
22	C17	3.066,0	20.3
23	C16	3.073,4	20.3
24	C18	3.129,3	20.4
25	C6	3.143,7	22.2
26	C7	3.261,4	22.2
27	C9	3.282,1	21.9
28	C8	3.457,3	32.2

Instância 42			
Pos	Comb	Custo da rota	Tempo de Proces.
1	C13	1.403,7	65.4
2	C1	1.417,5	66.0
3	C4	1.419,1	65.2
4	C22	1.421,3	24.9
5	C19	1.442,6	24.8
6	C10	1.476,3	24.6
7	C28	1.484,5	27.6
8	C25	1.511,0	27.6
9	C11	1.584,4	27.2
10	C12	1.667,0	62.6
11	C2	1.742,9	60.4
12	C14	1.778,2	60.8
13	C5	1.835,4	22.3
14	C15	1.876,0	21.9
15	C20	2.031,8	22.1
16	C26	2.155,2	22.8
17	C23	2.197,3	23.3
18	C6	2.204,7	23.0
19	C3	2.215,5	60.3
20	C24	2.232,3	60.7
21	C21	2.235,4	60.4
22	C27	2.273,5	21.8
23	C16	2.569,6	22.0
24	C17	2.696,7	21.3
25	C18	2.710,4	22.1
26	C9	2.761,4	22.5
27	C8	2.972,4	21.9
28	C7	2.986,8	38.1

A Tabela 3 apresenta os custos, em unidades de comprimento, das rotas obtidas com todas as combinações para as instâncias com maior número de vértices (instâncias com 52, 60 e 76 vértices), além dos tempos de processamento, em segundos, de cada uma delas As 10 primeiras posições foram ocupadas praticamente pelas mesmas combinações descritas acima. A única exceção ocorreu na instância de 60 vértices,

com a combinação C14 (Torneio, OX, Substituição) na décima posição. Considerando apenas as cinco primeiras posições, percebe-se que as combinações C13, C4, C22, C1 e C19 foram as mais frequentes em todas as instâncias.

Tabela 3. Custo médio das rotas e tempos de processamento com elitismo.

Instância 52			
Pos	Comb	Custo da rota	Tempo de Proces.
1	C13	8.716,6	81.9
2	C4	8.718,5	80.0
3	C22	8.929,8	79.2
4	C1	9.092,9	27.8
5	C19	9.227,8	28.4
6	C28	9.267,4	27.7
7	C10	9.878,5	31.8
8	C25	9.979,2	31.5
9	C11	10.044,0	31.1
10	C12	10.612,0	74.7
11	C2	11.379,3	74.9
12	C15	11.426,3	77.3
13	C14	11.580,7	24.6
14	C5	12.031,7	24.1
15	C24	13.545,4	24.3
16	C21	13.616,8	25.5
17	C6	13.689,8	26.0
18	C20	13.729,2	24.1
19	C3	14.144,6	78.4
20	C27	14.262,1	75.1
21	C23	14.503,0	73.6
22	C26	15.513,2	22.5
23	C17	18.528,6	22.6
24	C16	18.688,2	22.4
25	C18	18.903,2	22.9
26	C9	19.887,8	23.7
27	C7	20.815,2	23.2
28	C8	21.156,0	44.7

Instância 60			
Pos	Comb	Custo da rota	Tempo de Proces.
1	C13	12.749,3	96.4
2	C22	12.758,6	96.0
3	C4	12.809,1	94.1
4	C1	14.014,5	30.7
5	C19	14.179,8	30.9
6	C28	14.654,7	30.6
7	C10	15.495,4	35.0
8	C25	16.264,3	34.1
9	C11	16.528,7	34.9
10	C14	17.475,3	89.1
11	C12	17.543,2	88.5
12	C5	17.953,7	88.2
13	C2	18.237,9	24.5
14	C15	18.645,3	24.4
15	C23	20.791,1	24.4
16	C20	21.433,1	24.7
17	C24	23.933,9	25.0
18	C26	24.478,2	24.6
19	C6	24.732,2	86.9
20	C3	24.838,0	87.3
21	C27	25.145,8	86.9
22	C21	25.340,1	23.2
23	C16	32.143,4	23.4
24	C17	33.048,5	23.6
25	C18	33.499,9	23.7
26	C8	34.030,2	24.2
27	C7	36.173,1	23.1
28	C9	36.890,7	49.5

Instância 76			
Pos	Comb	Custo da rota	Tempo de Proces.
1	C13	610,4	117.5
2	C4	623,9	111.8
3	C22	636,9	108.5
4	C1	735,8	35.7
5	C19	736,5	36.6
6	C28	759,1	36.8
7	C10	815,0	41.5
8	C11	819,0	41.8
9	C25	832,7	40.3
10	C12	875,2	100.3
11	C14	879,2	97.1
12	C15	890,8	96.9
13	C5	920,2	25.6
14	C2	921,1	25.6
15	C23	1.037,7	25.6
16	C24	1.110,3	25.8
17	C6	1.124,8	26.6
18	C20	1.129,0	25.8
19	C3	1.149,9	102.1
20	C21	1.156,8	98.7
21	C27	1.175,6	94.9
22	C26	1.259,9	23.9
23	C18	1.606,1	24.2
24	C16	1.613,6	23.9
25	C17	1.618,6	23.9
26	C9	1.641,1	25.0
27	C8	1.722,8	23.4
28	C7	1.723,7	55.4

Na Tabela 4, são comparadas as melhores soluções encontradas pelas cinco melhores combinações com as soluções ótimas das instâncias selecionadas da biblioteca TSPLIB. Nota-se que as combinações C22 e C13 tiveram melhores resultados.

Tabela 4. Comparativo das melhores soluções encontradas pelas cinco melhores combinações com a solução ótima conhecida**.**

Instâncias	29	42	52	76
Ótimo	2.020	1.273	7.542	538
C13	2.087	1.313	7.791	563
C4	2.109	1.313	7.548	582
C22	2.087	1.313	7.543	575
C1	2.098	1.313	8.186	628
C19	2.080	1.313	8.193	670

Considerando apenas a instância de 29 vértices (Bays29), na Tabela 5, são mostrados os valores de variância das cinco melhores combinações. A combinação C22 (Torneio, CX, Inversão) apresentou resultados mais próximos da média, ou seja, teve um comportamento mais uniforme na busca pelas rotas. A combinação C1 (Roleta, PMX, Inversão), por sua vez, apresentou um conjunto de soluções mais irregular.

Tabela 5. Variância das soluções encontradas nos testes com a instância *Bays29* (29 vértices)**.**

Combinações	Variância
C13	1.609989e+4
C4	1.278410e+4
C22	1.295423e+4
C1	9.150195556
C19	1.865289e+4

Na Tabela 6, são mostradas as médias obtidas pelas três melhores combinações de operadores nos testes (Combinações 13, 04 e 22) para diferentes taxas de elitismo e diferentes números de cromossomos em cada execução (populações). As médias destas combinações, obtidas nos testes com a instância *Bays29*, foram somadas para avaliar como a variação da população e da taxa de elitismo pode influenciar na qualidade das rotas. O primeiro aspecto observado foi que apenas o aumento do número de cromossomos não é suficiente para melhorar a qualidade das soluções.

Tabela 6. Soma das médias obtidas pelas três melhores combinações para diferentes valores de elitismo e diferentes populações (Instância Bays29).

Elitismo	Pop30	Pop100	Pop200	Pop300	Pop400	Pop500	Média
0,00%	2.533,91	2.406,47	2.708,07	2.767,00	2.772,56	2781,31	**2.661,5**
1,00%	2.521,88	2.312,64	2.324,63	2.306,63	2.333,76	2306,81	**2.351,0**
10,00%	2.324,42	2.318,71	2.322,57	2.333,64	2.326,19	2323,12	**2.324,7**
20,00%	2.341,19	2.332,29	2.350,26	2.337,17	2.342,36	2316,44	**2.336,6**
Média	**2.430,35**	**2.342,53**	**2.426,38**	**2.436,11**	**2.443,71**	**2.431,92**	

O uso do elitismo, mesmo que em pequenas taxas, é necessário. Além disso, percebeu-se que a taxa de 10% de elitismo apresentou o menor valor nas médias de custo somadas (2.324,78). A população de 100 cromossomos, por sua vez, proporcionou o melhor desempenho, com um custo médio de 2.342,53. Ainda analisando a tabela, algumas taxas de elitismo foram melhores para determinadas populações do que para outras. A população de 300 cromossomos, assim como a de 500 cromossomos,

apresentou uma baixa média de custo com a taxa de elitismo fixada em 1%.

Ao se comparar os operadores individualmente, é possível determinar, de forma mais precisa, seus respectivos desempenhos. Ao analisar os operadores de seleção, tal como mostrado na Tabela 7, verifica-se que o operador Torneio foi superior aos outros dois, em termos de custo, nos testes sem elitismo. Porém, com elitismo, o operador Roleta conseguiu apresentar desempenho próximo do Torneio e o operador Ranking Linear melhorou notavelmente seus resultados. Os operadores de cruzamento, por sua vez, apresentaram poucas diferenças nos testes, porém, com uma pequena superioridade do operador CX em relação aos demais. Quanto aos operadores de mutação, verifica-se que o operador Inversão apresentou o custo médio mais baixo.

Tabela 7. Comparativo individual dos operadores.

Operadores de Seleção	**Roleta**	**Torneio**	**Ranking**
Custo	6.907,07	6.890,13	7.897,17
Operadores de Cruzamento	**PMX**	**OX**	**CX**
Custo	8.612,54	7.832,84	7.551,27
Operadores de Mutação	**Inversão**	**Substituição**	**Troca**
Custo	6.898,38	8.234,26	8.864

8. CONCLUSÕES E TRABALAHOS FUTUROS

Neste artigo, foram detalhados os resultados obtidos com os testes realizados envolvendo nove operadores genéticos para seis diferentes instâncias do PCV clássico. As melhores combinações foram capazes de encontrar soluções bastante próximas do valor ótimo conhecido das instâncias da biblioteca TSPLIB. Assim, esta técnica mostrou-se capaz de contornar a complexidade O (n!) do problema, ao favorecer cromossomos associados a rotas de baixo custo, o que a torna uma alternativa bastante relevante da computação evolutiva. Nos testes, observou-se que as combinações com os operadores Roleta e Torneio tiveram um melhor desempenho na busca por rotas de baixo custo. Observou-se também que as soluções encontradas pelas combinações com o operador Ranking Linear, apresentaram os custos mais elevados. Como este operador consegue evitar que o processo de seleção seja dominado por um cromossomo de alta aptidão, é possível que as combinações com esse operador precisem de um tempo maior para convergir. No trabalho apresentado por Razali e Geraghty (2011), o Ranking Linear apresentou os melhores resultados em comparação aos operadores Torneio e Roleta, porém, é preciso ressaltar que, neste caso, os parâmetros utilizados foram diferentes dos parâmetros adotados no presente trabalho.

Considerando apenas as instâncias com soluções ótimas conhecidas, as

combinações C13 e C22 se destacaram. É possível que, com um maior número de gerações e com algum método de escalonamento para a função de avaliação, estes valores diminuam, aproximando-se ainda mais dos valores ótimos. No comparativo entre operadores, os operadores Torneio, CX e Inversão apresentaram os melhores desempenhos em termos de custo médio das rotas. Em Abdoun e Abouchabaka (2011), dos seis operadores de cruzamento testados, o OX foi o que obteve as melhores rotas.

Nos testes adicionais com diferentes tamanhos de população e taxa de elitismo, as melhores médias de custo foram obtidas com populações de 300 e 500 cromossomos e taxa de elitismo de 1%. Porém, na soma das médias, a população de 100 cromossomos e taxa de 10% foram superiores. Conforme explicado em Diaz-Gomez e Hougen (2007), a escolha do tamanho ideal de população para um GA, assim como a taxa de elitismo, é ainda bastante desafiadora e, tal como explicado em Koljone e Alander (2006), não há um método único capaz de escolher os melhores parâmetros internos de um GA para todas as situações.

Em trabalhos futuros deverão ser feitos testes para diferentes taxas de mutação com as mesmas combinações. Também deverão ser feitos testes variando o número de iterações para o algoritmo, além de atribuir uma probabilidade de ocorrência para a etapa de cruzamento, além de avaliar o desempenho das combinações em arquiteturas usando múltiplos processadores, como é o caso das GPUs.

REFERÊNCIAS

ABDOUN, O.; ABOUCHABAKA., J. A comparative study of adaptive crossover operators for genetic algorithms to resolve the traveling salesman problem. **International Journal of Computer Applications**, v. 31, n. 11, p. 49–57, 2011.

APPLEGATE, D. **The Traveling salesman problem:** a computational study. Princeton University Press, 2006. (Princeton Series in Applied Mathematics).

ARTERO, A. I**nteligencia artificial:** teoria e prática. São Paulo: Livraria da Física, 2009.

CALADO, M. F.; LADEIRA, P. A. Problema do caixeiro viajante: Um estudo comparativo de técnicas de inteligência artificial. **E-xacta**, 2011.

CASTRO, N. L.; MASUTTI, S. A. T. Um algoritmo imunológico para a solução do caixeiro viajante. **Revista Eletrônica de Iniciação Científica**, 2009.

DIAZ-GOMEZ, A. P.; HOUGEN, F. D. T. **Initial population for genetic algorithms:** a metric apporach. University of Oklahoma, 2007.

GOLDBERG, D. **Genetic algorithms in search, optimization, and machine learning**. Addison-Wesley, 1989. (Artificial Intelligence).

GUTIN, G.; PUNNEN, A. **The traveling salesman problem and its variations**. Springer, 2002. (Combinatorial Optimization).

KOLJONEN, J.; ALANDER, T, J. **Effects of population size and relative elitism on**

optimization speed and reliability of genetic algorithms. University of Vaasa, 2006.
LAWYER, E. et al. **The traveling salesman problem**. John Wiley & Sons, 1985.

LINDEN, R. **Algoritmos genéticos**. 3. ed. Rio de Janeiro: Ciência Moderna, 2012.

MAN, K.; TANG, K.; KWONG, S. **Genetic algorithms:** concepts and designs, avec disquette. London: Springer, 1999. (Advanced Textbooks in Control and Signal Processing). https://doi.org/10.1007/978-1-4471-0577-0

MICHALEWICS, Z. **Genetic algorithms + datastructures = evolution programs**. Springer, 1996. (Artificial intelligence).

MITCHELL, M. **An Introduction to genetic algorithms.** 1. ed. London: MIT, 1999.

NEIA, S. S. et al. Roteamento de veículos utilizando otimização por colonia de formigas e algoritmo genético. In: **Meta- heurísticas em pesquisa operacional**. Curitiba: Omnipax, 2013. p. 220–236. https://doi.org/10.7436/2013.mhpo.14

PAPADIMITRIOU, H. C. **Theoretical computer science**. Cambridge: Elsevier, 1977. p. 237-244.

RAZALI, M. N.; GERAGHTY, J. **Genetic algorithm performance with different selection strategies in solving tsp**. In: London: [s.n.], 2011. v. II.

REINELT, G. **TSPLIB 95**. Neuenheimer Feld 294. Heidelberg, 1995.

SILVA, F. A.; OLIVEIRA, C. A. **Algoritmos genéticos:** alguns experimentos com os operadores de cruzamento (crossover) para o problema do caixeiro viajante assimétrico. Fortaleza: [s.n.], 2006.

VIEIRA, E. L. **Algoritmo evolutivo para o problema do caixeiro viajante com demandas heterogêneas.** Santa Maria: [s.n.], 2006.

YAMAKAMI, A.; CARVALHO, B. M. Meta-heurística híbrida de sistema de colônia de formigas e algoritmo genético para o problema do caixeiro viajante. **Sociedade Brasileira de Matemática Aplicada e Computacional**, v. 9, n. 1, p31–40, 2008.