

Problema de Roteamento em Arcos Dominante

João Pedro Ramos Lopes

Instituto de Computacao - UNICAMP
Av. Albert Einstein 1251, CEP 13083-852, Campinas/SP - Brasil
jpedrorl@gmail.com

Rafael Kendy Arakaki

Instituto de Computacao - UNICAMP
Av. Albert Einstein 1251, CEP 13083-852, Campinas/SP - Brasil
rafael.arakaki@ic.unicamp.br

Fábio Luiz Usberti

Instituto de Computação - UNICAMP
Av. Albert Einstein 1251, CEP 13083-852, Campinas/SP - Brasil
fusberty@ic.unicamp.br

RESUMO

Este trabalho apresenta um novo problema de otimização combinatória NP-difícil denominado Problema de Roteamento em Arcos Dominante (em inglês, Dominating Arc Routing Problem - DARP). O objetivo do DARP é encontrar um circuito de custo mínimo tal que todas as arestas do grafo sejam dominadas pelas arestas do circuito, ou seja, estejam a uma determinada distância máxima das arestas do circuito. Neste trabalho, será investigada a metodologia heurística de Algoritmos Genéticos para a obtenção de soluções heurísticas para o problema. Este problema possui aplicações práticas em áreas de monitoramento wireless como, por exemplo, coleta de valores de consumo de água e energia de clientes por um veículo com dispositivo leitor sem fio. Este trabalho é candidato ao prêmio de Iniciação Científica do Simpósio Brasileiro de Pesquisa Operacional de 2018. O aluno foi responsável pela idealização, implementação e testes das metodologias propostas, assim como pela elaboração deste artigo.

PALAVRAS CHAVE. roteamento em arcos, algoritmo genético, conjunto dominante, otimização combinatória.

Otimização Combinatória, Metaheurística

ABSTRACT

This work presents a new NP-hard combinatorial optimization problem named Dominating Arc Routing Problem (DARP). The DARP is defined as finding the minimum cost circuit where each edge in the graph is dominated by an edge in the circuit, which means that each edge must be within a given distance from the circuit. This project will explore Genetic Algorithm methodologies aimed at finding heuristic solutions for the problem. The solutions proposed for this problem has applications on monitoring systems that use wireless devices to gather information (e.g., water and energy meter reading). This work is a 2018 SBPO Undergraduate Scientific Research nominee. The student was responsible for the idealization, implementation and testing of the proposed methodologies, as well as for the writing of this paper.

KEYWORDS. arc routing, genetic algorithm, dominating sets, combinatorial optimization.

Combinatorial Optimization, Metaheuristics

Introdução e Justificativa

Conceitos Introdutórios

Um grafo $G = (V, E)$ é definido por um conjunto V de vértices e um conjunto E de arestas. Uma sequência de vértices $(v_1, \dots, v_n)$, $v_i \in V$, é dita um passeio em G se, dado $\forall i \in [1, n)$, todas as arestas (v_i, v_{i+1}) pertencem a E . Um passeio é denominado um circuito em G se os vértices inicial e final são os mesmos, ou seja, $v_1 = v_n$. Seja atribuído um custo associado a cada aresta $w(u, v)$. O custo de um circuito $(v_1, \dots, v_n)$ é então definido pelo somatório dos custos de suas arestas: $\sum_{i=1}^{n-1} w(v_i, v_{i+1})$ [Gross e Yellen, 2004].

Dentre os problemas de otimização combinatória, pode-se estabelecer duas classes importantes: problemas que podem ser resolvidos com complexidade polinomial em uma máquina de Turing determinística (classe P) e os que podem ser resolvidos em tempo polinomial em uma máquina de Turing não-determinística (classe NP) [Gary e Johnson, 1979]. Supondo que $P \neq NP$, não há algoritmo em tempo polinomial para os problemas pertencentes à classe NP e, por isso, o uso de metodologias heurísticas é de especial interesse para essa classe de problemas.

O Problema de Roteamento em Arcos Dominante (DARP)

Seja $G = (V, E)$ um grafo não-direcionado, uma aresta $(u, v) \in E$ possui um conjunto de arestas k -dominadas denotado por $D_k(u, v)$. Seja $d(u, v)$ o custo do caminho mínimo do vértice u ao vértice v , o conjunto $D_k(u, v)$ é definido pela união de arestas dos seguintes casos:

- Caso 1: $\{(i, j) \in E \mid d(i, u), d(j, u) \leq k\}$ (Figura 1)
- Caso 2: $\{(i, j) \in E \mid d(i, v), d(j, v) \leq k\}$ (Figura 2)
- Caso 3: $\{(i, j) \in E \mid d(i, v), d(j, u) \leq k\}$ (Figura 3)
- Caso 4: $\{(i, j) \in E \mid d(i, u), d(j, v) \leq k\}$ (Figura 3)

Os casos de dominância encontram-se ilustrados nas figuras de 1 a 3, onde são apresentadas tracejadas as arestas dominadas por (u, v) , onde $k = 2$ e $w(i, j) = 1, \forall (i, j) \in E$. Dado que uma aresta sempre se domina, tal caso foi omitido das imagens. A figura 4 apresenta todas as arestas dominadas pela aresta (u, v) . Para simplificação do texto, deste ponto em diante o termo *dominado* refere-se à definição k -dominado.

Dessa forma, um roteamento em arcos dominante em um grafo pode ser definido como um circuito, $C \subseteq E$, tal que as arestas desse circuito dominam as arestas do grafo, ou seja, $\{\forall e \in E, \exists c \in C \mid e \in D_k(c)\}$. Na figura 5 é apresentado um circuito dominante para $k = 1$ e, na Figura 6, para $k = 2$. O DARP é um problema NP-difícil, uma vez que é uma generalização de um problema NP-difícil: o problema do conjunto k -dominante (Seção 2.2).

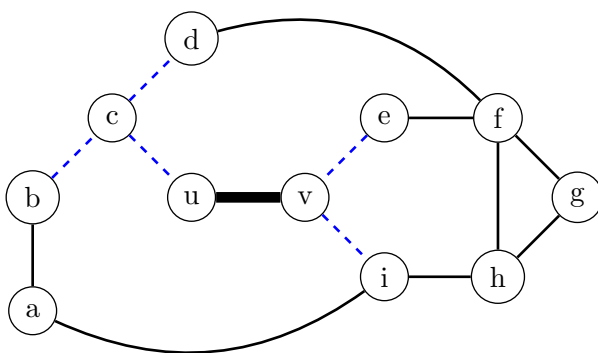


Figura 1: Dominância em arestas - Caso 1

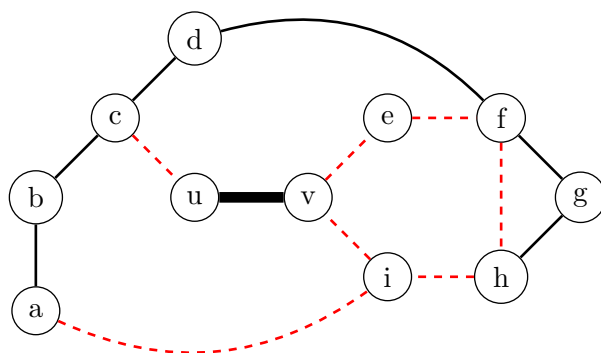


Figura 2: Dominância em arestas - Caso 2

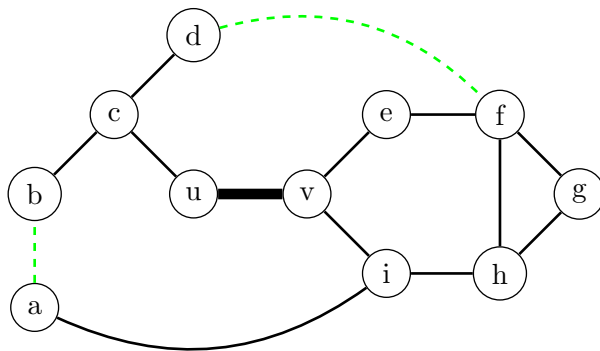


Figura 3: Dominância em arestas - Casos 3 e 4

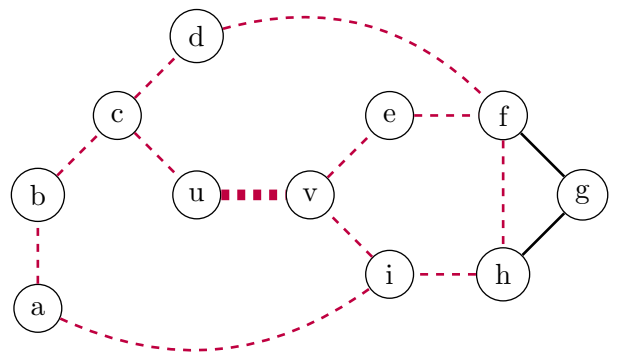


Figura 4: Arestas dominadas por (u, v) para $k = 2$

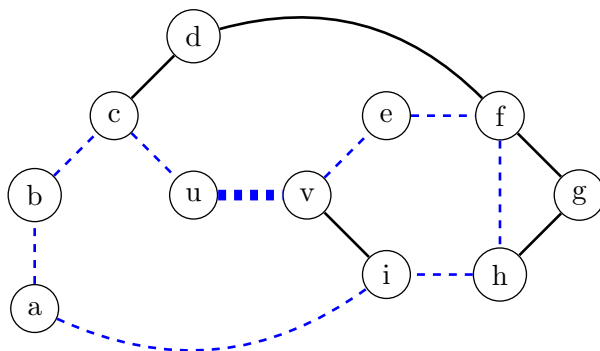


Figura 5: Solução para o DARP para $k = 1$

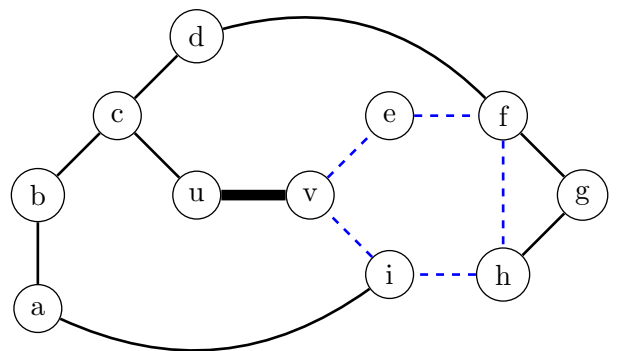


Figura 6: Solução para o DARP para $k = 2$

Aplicação Prática

Uma aplicação prática do problema está relacionada às companhias prestadoras de serviço (água, energia elétrica, gás). Suponha um veículo com um leitor sem-fio e alcance limitado; a companhia deseja fazer a leitura dos consumos das residências para as quais presta serviço. O dispositivo móvel deve se conectar aos diversos dispositivos das residências para troca de dados. No entanto, o veículo deve atender a todas as residências atribuídas a ele e está limitado a transitar somente pelas ruas da área de concessão da companhia. Cabe notar que esse problema é naturalmente modelado através de um grafo, onde as arestas e nós são segmentos e intersecções de ruas.

Trabalhos Relacionados

Problema do Carteiro Chinês (CPP)

O Problema do Carteiro Chinês (Chinese Postman Problem) [Eiselt et al., 1995] é definido em um grafo não-direcionado $G = (V, E)$: onde V é um conjunto de vértices, E é um conjunto de arestas com custos $C = (c_{ij})$. Deseja-se encontrar um circuito de menor custo que contenha todas as arestas do grafo. Jack Edmonds demonstrou que o CPP é da classe P a partir de uma redução ao problema de emparelhamento máximo de custo mínimo [Edmonds e Johnson, 1973].

Problema do Conjunto k -Dominante (k-DSP)

Dado um grafo conexo não-direcionado $G = (V, E)$, um subconjunto de vértices $D \subseteq V$ é um conjunto k -dominante de G se, para cada vértice $v \in V$ existe um vértice $u \in D$ tal que $d(u, v) \leq k$, onde $d(u, v)$ é o custo do caminho mínimo do vértice u ao vértice v . Ou seja, os vértices de D k -dominam todos os vértices do grafo. O objetivo é encontrar um conjunto de tamanho mínimo que k -domine todos os vértices do grafo. Esse problema é NP-difícil, conforme demonstrado por Chen et al. [2004].

Uma generalização desse problema é o Problema do Ciclo k -Dominante, em que se objetiva encontrar o circuito de custo mínimo que k -domine o grafo [Maziero et al., 2016]. Esse problema é semelhante ao DARP, variando na definição de dominância do grafo (dominância em vértices contraposta à dominância em arestas).

Metodologia

Neste projeto, é proposta para a solução do DARP utilizando Algoritmos Genéticos (Sessão 4) com dois métodos de decodificação de um cromossomo: (i) dominância utilizando apenas arestas presentes explicitamente no cromossomo; (ii) dominância obtida por todas as arestas presentes no caminho mínimo existente entre duas arestas do cromossomo.

Algoritmos Genéticos

Algoritmos Genéticos (AGs) são metaheurísticas fundamentadas nos conceitos genéticos da biologia evolutiva. Explicando resumidamente, inicialmente é escolhido um conjunto de soluções viáveis denominado população inicial. A cada iteração o algoritmo cruza as soluções da população, obtendo novas soluções cruzadas, possivelmente realizando mutações, e então seleciona uma população sobrevivente para formar a próxima geração Gendreau e Potvin [2010]. O Algoritmo 1 apresenta um pseudo-código geral do modelo de Algoritmos Genéticos.

Algorithm 1: pseudo-código para algoritmos genéticos

```

início
    //Inicialização
    pre_processamento()
    pais = populacao_inicial()
    //Evoluir população
    repita
        repita
            //Geração
            cromossomos_pais = seleciona_pais(pais)
            cromossomo_filho = crossover(cromossomos_pais)
            mutacoes(cromossomo_filho)

            //Seleção
            se viavel(cromossomo_filho) então
                 $\sqcup$  filhos = filhos  $\cup$  cromossomo_filho
        até tamanho(filhos) suficiente ;
        //Atualiza nova geração
        pais = filhos
        filhos = {}
    até condicao_de_parada;
fim

```

Pré-processamento

Para auxiliar as operações do AG são apresentadas variáveis que são previamente computadas e posteriormente utilizadas pelo GA: (i) C_e : custo de uma aresta e ; (ii) $D_{u,v}$: menor custo entre vértices u e v ; (iii) DS_e : conjunto de arestas dominadas por uma aresta e ; (iv) $PDS_{u,v}$: conjunto de arestas dominadas por um caminho qualquer de custo mínimo entre vértices u e v .

Codificação

No DARP deseja-se estudar a relação de dominância de arestas, assim os cromossomos foram codificados como permutações das arestas do grafo. Um gene é composto pelo par ordenado $e = (u, v)$, onde u representa o vértice de entrada e v o vértice de saída da aresta, ou seja, apesar de o grafo G ser não-direcionado, a ordem das arestas em um gene define a sequência em que os vértices devem ser percorridos no circuito da solução.

Fitness

A fitness de um cromossomo é dada pelo custo do menor circuito dominante gerado a partir da visitação ordenada dos vértices pertencentes as arestas na ordem em que aparecem no cromossomo. Usualmente, AGs em problemas de dominância utilizam apenas a dominância dos genes (vértices ou arestas) presentes no cromossomo (algoritmo 2). É proposto um método alternativo, que considera também a dominância atrelada a um caminho mínimo entre dois genes adjacentes, representada pelo conjunto *PDS* (algoritmo 3).

Algorithm 2: cálculo de custo com dominância simples

```
Entrada: cromossomo
Saída: custo atrelado ao cromossomo
início
     $S = \{\}$  // Conjunto de arestas dominadas
     $aresta\_anterior = cromossomo.aresta[0]$ 
     $custo = C_{aresta\_anterior}$ 
    para  $e = (u, v) \in cromossomo[1 : ]$  e  $|S| < |cromossomo|$  faça
         $(p\_u, p\_v) = aresta\_anterior$ 
         $custo = custo + D_{p\_v, u}$  // Adiciona custo de menor caminho
         $custo = custo + C_e$  // Adiciona custo da aresta
         $S = S \cup DS_e$ 
         $aresta\_anterior = e$ 
     $e = cromossomo.aresta[0]$  // Adiciona custo de retorno
     $custo = custo + C_e$ 
    retorna custo
fim
```

Algorithm 3: cálculo de custo com dominância de caminhos

```
Entrada: cromossomo
Saída: custo atrelado ao cromossomo
início
     $S = \{\}$  // Conjunto de arestas dominadas
     $aresta\_anterior = cromossomo.aresta[0]$ 
     $custo = C_{aresta\_anterior}$ 
    para  $e = (u, v) \in cromossomo[1 : ]$  e  $|S| < |cromossomo|$  faça
         $(p\_u, p\_v) = aresta\_anterior$ 
         $custo = custo + D_{p\_v, u}$  // Adiciona custo do menor caminho
         $S = S \cup PDS_{p\_v, u}$ 
         $custo = custo + C_e$  // Adiciona custo da aresta e
         $S = S \cup DS_e$ 
         $aresta\_anterior = e$ 
     $e = cromossomo.aresta[0]$  // Adiciona custo de retorno
     $custo = custo + C_e$ 
    retorna custo
fim
```

População

A população ou geração de um algoritmo genético forma seu escopo de busca. Soluções não representadas por uma população podem ser geradas a partir de soluções presentes na população por meio dos operadores de *crossover* e de mutação. Na população inicial, cada cromossomo é formado a partir de uma permutação aleatória das arestas pertencentes ao grafo. O tamanho da população é uma função logarítmica do número de arestas no grafo, $10 \log_2(|E|)$.

Crossover

O operador de *crossover* é responsável por recombinar dois cromossomos visando a gerar um indivíduo com fitness possivelmente melhor. Diversidade é fundamental para

a efetividade desse operador, visto que em geral essa operação quando submetida a dois progenitores próximos gera um filho similar a seus pais.

Visando explorar a codificação de uma solução, é proposta a utilização dos operadores **C1** e **PMX**, que ocorrem com a mesma probabilidade.

Crossover C1

O operador C1 pode ser definido conforme: dados dois pais, p_1 e p_2 , é escolhido um ponto de corte, c , aleatório em p_1 ; o filho será idêntico a p_1 do início do cromossomo até c . Em seguida, são inseridos os genes faltantes ao filho na sequência em que aparecem em p_2 .

Crossover PMX

Dados dois pais, p_1 e p_2 , no operador PMX é escolhido um trecho aleatório em p_1 , $[c_1, c_2]$, no qual o filho será idêntico a p_1 . No cromossomo p_2 , nesse mesmo trecho, podem existir genes que não estão presentes no filho. Para cada um desses genes g_i , deve-se utilizar a posição correspondente k_i para: (i) encontrar o gene g_c que está em p_1 na mesma posição, k_i , que g_i está em p_2 ; (ii) encontrar a posição, k_c , em que g_c aparece em p_2 ; (iii) se k_c está no trecho original, deve-se ir para o primeiro item utilizando k_c ; (iv) senão, g_i deve ser inserido no filho na posição encontrada.

Em seguida, os itens restantes que estão fora do trecho inicial e não foram copiados para o filho de p_2 são copiados para o filho.

Seleção

Foi utilizado o método de seleção por torneio de tamanho fixo N . São selecionados aleatoriamente N possíveis indivíduos para o torneio e o melhor dentre esses indivíduos é selecionado como um dos progenitores. O tamanho do torneio foi fixado em 4 indivíduos, sendo gerado um torneio para a seleção dos progenitores de cada cruzamento.

Foi adotado elitismo de 1 indivíduo, ou seja, o melhor indivíduo de uma população certamente estará na geração seguinte. Isso garante que o melhor indivíduo nunca seja descartado de uma geração para a seguinte.

Mutação

O operador de mutação visa manter a diversidade em uma população, inserindo alterações aleatórias em cromossomos após o *crossover*. Foram implementados dois operadores de mutação: (i) *swap range*: são escolhidos aleatoriamente dois genes de um cromossomo e para todo esse intervalo os genes têm suas posições trocadas, onde o primeiro gene troca de posição com o último, o segundo troca com o penúltimo, etc. Tal operador ocorre com probabilidade de 3%; (ii) *invert range*: são escolhidos aleatoriamente dois genes de um cromossomo e para todo esse intervalo os vértices de cada aresta são invertidos, ou seja, o vértice de entrada passa a ser o vértice de saída. Esse operador ocorre com probabilidade de 7%.

Critério de parada

Foi utilizado como critério de parada baseado em tempo de execução: após o processamento de cada geração, se o tempo de processamento for superior a um limite pré-estabelecido, deve-se parar a execução do método e retornar a melhor solução obtida.

Experimentos computacionais

O presente experimento se objetiva obter os primeiros resultados de soluções viáveis para o DARF e também comparar os resultados obtidos pelo algoritmo genético utilizando dois esquemas diferentes de decodificação da dominância: (i) a dominância simples (Algoritmo 2); e (ii) a dominância por caminhos (Algoritmo 3).

Tabela 1: Resultados experimentais

instância	V	E	$k = 50$					$k = 100$					$k = 150$					média		
			GA_1	CPU_1	GA_2	CPU_2	ΔGA	GA_1	CPU_1	GA_2	CPU_2	ΔGA	GA_1	CPU_1	GA_2	CPU_2	ΔGA			
UR532	298	597	352962	1797	244574	1777	0.443	187447	1782	59118	1697	2.171	138728	1793	34107	1668	3.067	1.894		
UR535	458	812	471809	1806	145675	1809	2.239	161149	1821	34706	1420	3.643	74004	1469	21858	1439	2.386	2.756		
UR537	493	868	483541	1801	140115	1748	2.451	152235	1728	79424	1792	0.917	59013	1573	24983	1751	1.362	1.577		
UR542	343	862	460693	1767	338338	1802	0.362	192929	1821	60728	1757	2.177	73895	1832	21735	1443	2.400	1.646		
UR545	476	1104	615220	1818	130552	1783	3.712	111213	1772	36694	1284	2.031	53371	1799	23230	1345	1.298	2.347		
UR547	498	1112	535607	1808	182320	1810	1.938	104659	1838	33969	1036	2.081	44531	1679	21603	1413	1.061	1.693		
UR552	388	1135	575368	1812	405464	1808	0.419	158521	1784	58031	1168	1.732	49424	1833	22490	1762	1.198	1.116		
UR555	490	1305	677014	1816	326254	1816	1.075	94208	1769	28464	973	2.310	44922	1889	14365	1373	2.127	1.837		
UR557	498	1310	663618	1807	167334	1821	2.966	116035	1835	44330	907	1.618	41010	865	13625	1469	2.010	2.198		
UR562	416	1403	710290	1827	515010	1821	0.379	201022	1871	45499	1698	3.418	42956	1882	13166	1788	2.263	2.020		
UR565	496	1513	663092	1817	260150	1812	1.549	94641	1880	31524	1715	2.002	39718	1771	13131	1872	2.025	1.859		
UR567	499	1526	711127	1811	167002	1819	3.258	110635	1890	46231	1574	1.393	39808	1524	14046	1900	1.834	2.162		
UR732	452	915	638005	1809	476102	1803	0.340	350621	1824	95287	1792	2.680	128547	1814	40005	1181	2.213	1.744		
UR735	662	1200	688187	1801	301931	1802	1.279	193689	1803	59620	1229	2.249	60868	1838	30916	1431	0.969	1.499		
UR737	744	1315	568531	1798	172153	1806	2.302	161665	1834	66193	1663	1.442	80616	1786	35925	1576	1.244	1.663		
UR742	538	1325	683122	1827	414117	1802	0.650	264016	1825	68129	1185	2.875	54932	1742	22746	1860	1.415	1.647		
UR745	713	1616	687802	1806	201563	1820	2.412	138086	1838	42080	1758	2.282	52117	1868	29190	1927	0.785	1.826		
UR747	745	1659	747171	1838	173145	1799	3.315	144846	1827	42624	1865	2.398	54877	1844	19133	1772	1.868	2.527		
UR752	580	1735	828644	1775	558522	1815	0.484	304984	1835	103282	1878	1.953	56174	1685	16705	1966	2.363	1.600		
UR755	724	1966	951376	1827	201271	1830	3.727	123941	1738	39374	1856	2.148	51299	1869	23677	1859	1.167	2.347		
UR757	748	1969	922587	1847	233113	1840	2.958	114894	1880	40251	1830	1.854	49567	1744	16184	1907	2.063	2.292		
UR762	593	2089	1027092	1829	737354	1864	0.393	153475	1928	48087	1701	2.192	52554	1790	20976	2021	1.505	1.363		
UR765	741	2278	838761	1840	212992	1872	2.938	107301	1906	34511	1850	2.109	47420	1945	21445	1897	1.211	2.086		
UR767	749	2314	1097286	1867	227873	1875	3.815	123334	1547	43510	1924	1.835	53413	1240	19597	1948	1.726	2.459		
média							1.892						2.146						1.732	1.923

GA_1, GA_2 : custo das soluções obtidas utilizando-se respectivamente dominância simples e dominância de caminhos.

CPU_1, CPU_2 : tempo de processamento para atingir os valores de GA_1 e GA_2 , respectivamente (em segundos).

$\Delta GA = (GA_1 - GA_2)/GA_2$: fator de variação entre GA_1 e GA_2 .

Foi utilizado um conjunto de instâncias da literatura de roteamento em arcos apresentado por Corberán et al. [2007]. Para cada instância, foi realizada a execução para três valores distintos de k (distância máxima que define dominância): $k = 50$, $k = 100$ e $k = 150$. A Tabela 1 contém os resultados do experimento. Na Tabela 1, GA_1 e GA_2 são os resultados dados pelo algoritmo genético utilizando respectivamente a dominância simples e a dominância de caminhos. A partir disso, computou-se um fator de variação entre GA_1 e GA_2 dado por $\Delta GA = (GA_1 - GA_2)/GA_2$ que representa a taxa melhoria de GA_2 em relação a GA_1 . Todos os experimentos tiveram como critério de parada o tempo de execução de meia hora para cada instância.

Os resultados apontam para uma melhor performance do método que utiliza dominância de caminhos. Em algumas instâncias como a *UR755* com $k = 50$ o desempenho de GA_2 foi mais de quatro vezes melhor do que GA_1 . Em relação ao fator de distância máxima k , não foi possível apontar um padrão sobre o impacto claro desse parâmetro na dificuldade de solução das instâncias. Por exemplo, embora para *UR752* com $k = 150$ a variação ΔGA é de 2.363, para a mesma instância e $k = 50$ a variação é de apenas 0.484. Porém para a instância *UR747*, por exemplo, notou-se que um comportamento oposto, onde a variação ΔGA para $k = 50$ foi de 3,315 e para $k = 150$ foi de 1,868. Uma das explicações para isso é devido ao fato deste ser um trabalho pioneiro, não havendo ainda valores de limitantes inferiores para o problema, assim como a falta de outros métodos heurísticos para comparação.

Conclusões

O método de codificação de dominância por caminhos mostrou um desempenho muito superior ao método de codificação convencional, de dominância simples. Apesar de terem sido encontrados valores interessantes para essa abordagem, é necessária a experimentação com conjuntos de instâncias mais variados e verificar a possibilidade de melhorias no método, como buscas locais, hibridização com outras metaheurísticas, etc.

A ideia de decodificação por caminhos mínimos utilizada pelo AG pode ser explorada e adaptada para outros problemas de dominância, como o problema do ciclo k -dominante. Além disso, o desenvolvimento de métodos para geração de limitantes inferiores também é uma área de pesquisa interessante.

Referências

- Chen, N., Meng, J., Rong, J., e Zhu, H. (2004). Approximation for dominating set problem with measure functions. *Computing and Informatics*, 23(1):37–49.
- Corberán, A., Plana, I., e Sanchis, J. M. (2007). A branch & cut algorithm for the windy general routing problem and special cases. *Networks*, 49(4):245–257.
- Edmonds, J. e Johnson, E. L. (1973). Matching, euler tours and the chinese postman. *Mathematical programming*, 5(1):88–124.
- Eiselt, H. A., Gendreau, M., e Laporte, G. (1995). Arc routing problems, part i: The chinese postman problem. *Operations Research*, 43(2):231–242.
- Gary, M. R. e Johnson, D. S. (1979). Computers and intractability: A guide to the theory of np-completeness.
- Gendreau, M. e Potvin, J.-Y. (2010). *Handbook of metaheuristics*, volume 2. Springer.
- Gross, J. L. e Yellen, J. (2004). *Handbook of graph theory*. CRC press.
- Maziero, L. P., Usberti, F. L., e Cavellucci, C. (2016). Algoritmos heurísticos e exatos para o problema do ciclo dominante.