UM MODELO MATEMÁTICO PARA O PROBLEMA DE ORIENTAÇÃO EM *CLUSTERS* COM REGRA DE PRIORIDADE d-RELAXADA

Júlio Cesar Vasconcelos Conserva
Universidade Federal de Campina Grande
Rua Luiz Grande, s/n – Bairro Frei Damião, Sumé – PB, CEP 58540-000
julio.conserva@outlook.com

Walton Pereira Coutinho
Universidade Federal de Pernambuco
Campus do Agreste, Av. Marielle Franco, s/n – Nova Caruaru – PE, CEP 55014-900
walton.coutinho@ufpe.br

RESUMO

O *Orienteering Problem* (OP) é um problema de otimização combinatória bastante conhecido. O presente artigo apresenta uma nova variante do OP, chamada de Problema de Orientação em *Clusters* com Regra de Prioridade d-Relaxada (CluOP-d). Neste problema, os locais a serem visitados são agrupados em clusters, com diferentes níveis de urgência, tal que os locais com maior prioridade devem ser visitados antes dos menos prioritários. Um prêmio é associado a cada localidade e um valor *d* é atribuído para balancear o compromisso entre a eficiência operacional e o cumprimento da prioridade. O objetivo é maximizar o prêmio coletado em um tempo de viagem limitado. Neste trabalho, é apresentado um modelo matemático para o CluOP-d baseado em programação inteira mista. Experimentos computacionais preliminares atestam a efetividade do modelo desenvolvido e o impacto da regra de prioridade d-relaxada.

PALAVRAS CHAVE. Orienteering Problem, Regra de Prioridade d-Relaxada, Programação Inteira Mista.

Tópicos: Logística e Transporte, Otimização Combinatória.

ABSTRACT

The Orienteering Problem (OP) is a well known combinatorial optimization problem. This article presents a new variant of the OP, the so-called Clustered Orienteering Problem with d-Relaxed priority rule (CluOP-d). In this problem, locations are grouped into clusters with different priority levels, such that higher priority locations must be visited before lower priority ones. A score is associated with each location and a value *d* is defined with the purpose of balancing the trade-off between operational efficiency and respecting the location's priorities. In the CluOP-d the objective is to maximize the total collected score provided a limited travel time. In this work, a mathematical formulation based on mixed-integer programming is presented for the CluOP-d. Preliminary computational experiments show the effectiveness of the proposed model and the impact of the d-relaxed priority rule.

KEYWORDS. Orienteering Problem, d-Relaxed Priority Rule, Mixed-Integer Programing.

Paper topics: Logistics and Transport, Combinatorial Optimization.

1. Introdução

Existem diversos problemas na logística que envolvem a tomada de decisão no transporte, podendo-se encontrar diversas variantes na literatura. Um desses é o Problema do Caixeiro Viajante (TSP, do inglês *Travelling Salesman Problem*) que possui inúmeras aplicações. De acordo com [Applegate et al. 2007], o TSP consiste em estabelecer uma rota, iniciando seu percurso em uma cidade de origem, passando por todas as demais cidades uma única vez e retornando à origem percorrendo a menor distância possível.

No TSP, o conjunto de locais a serem visitados têm o mesmo grau de urgência, podendo estes ser visitados em qualquer ordem em uma solução viável. Contudo, no mundo real existem diversas ocasiões em que diferentes níveis de prioridades de visitação devem ser levados em consideração, como nos casos de ajuda humanitária em desastres naturais, ou na distribuição de suprimentos a populações vulneráveis. Assim diferentes locais com mesmo nível de urgência podem ser agrupados e os que possuem maior prioridade devem ser visitados antecipadamente. Na literatura, este problema é conhecido como o Problema do Caixeiro Viajante Clusterizado (CTSP, do inglês *Clustered Traveling Salesman Problem*) [Hà et al. 2020].

O fato de obrigar o veículo a visitar clusters com prioridade mais alta antes dos que possuem prioridade mais baixa pode acabar não sendo eficiente do ponto de vista operacional. Para que se possa equilibrar o compromisso entre a eficiência operacional e o cumprimento das prioridades, pode-se utilizar a regra de prioridade d-relaxada, onde o veículo pode visitar um local com prioridade *d* menor antes de visitar um local de prioridade mais alta. Para a regra de prioridade 0-relaxada o veículo é obrigado a atender os locais com prioridade *p*, e se no seu caminho encontra um com prioridade $p + 1$ teria de voltar depois para atendê-lo, precisando assim terminar o atendimento de todos os locais com prioridade *p* antes de visitar locais com prioridade menor que *p*. Já se tivermos a regra de prioridade 1-Relaxado permitiria a visita em um local $p + 1$ antes de encerrar todos os locais *p* [Panchamgam 2011].

Nesse sentido [Hà et al. 2020], propuseram o Problema do Caixeiro Viajante Clusterizado com regra de Prioridades Relaxadas (CTSP-d, do inglês *Clustered Traveling Salesman Problem with d-relaxed priority rule*), em que os *clusters* possuem diferentes níveis de urgência, ou diferentes prioridades, e o valor de d é definido tal que exista uma compensação entre a emergência da viagem e o custo. A aplicação de tal método é importante por considerar um *trade-off* no planejamento das rotas, respeitando até certo nível a prioridade das localidades, mas identificando oportunidades para a diminuição do custo total da rota.

Dentre as diversas variantes do CTSP presentes na literatura existe ainda o *Clustered Team Orienteering Problem* (CluTOP). Para [Yahiaoui 2019], nesta variante, é atribuído um prêmio para cada *cluster*, sendo obtidos quando ocorre a visita ao cliente. Então o veículo tem como objetivo maximizar o prêmio total coletado e minimizar o custo em um determinado tempo limitado de viagem.

Diante desses cenários é possível observar que em diversos problemas de logística humanitária, por exemplo, no gerenciamento de equipes de emergência, é necessário decidir quais locais críticos visitar e em que ordem, considerando a urgência do cluster, respeitando a regra de prioridade e sem deixar de lado a eficiência logística. Com isso, essa pesquisa propõe um novo problema, chamado de Problema de Orientação em *Clusters* com Regra de Prioridade d-Relaxada (CluOP-d, do inglês *Clustered orienteering problem with d-relaxed priority rule*). Este problema divide os clientes em *clusters*, fazendo o uso da regra de prioridades d-relaxada, que pode ser atenuada a partir do valor definido *d*, além disso, a cada cluster está associado um prêmio que tem como intuito a priorização das visitas a locais mais urgentes, sendo a estes atribuído prêmios maiores. O objetivo deste problema é definir a melhor sequência de visita para que a soma desses prêmios seja maximizada.

Neste trabalho propõe-se uma formulação matemática baseada em Programação Inteira Mista (MIP, do inglês *Mixed-Integer Programming*) para o CluOP-d, que foi construída a partir de adaptações de modelos presentes na literatura para problemas semelhantes, e posteriormente foi

realizada uma análise dos resultados preliminares obtidos por meio de experimentos computacionais, com o intuito de verificar o desempenho do modelo para os diferentes parâmetros estabelecidos, avaliando a efetividade da regra de prioridade d-relaxada e o impacto nos resultados que tempo disponível para a viagem apresenta.

O restante deste trabalho está organizado da seguinte maneira. Na Seção 2 apresenta-se os últimos trabalhos realizados na literatura relacionada ao tema desta pesquisa. Na Seção 3, propõe-se uma formulação matemática para o CluOP-d. Na seção 4, apresenta-se os resultados preliminares de experimentos computacionais. A seção 5 conclui este trabalho e direciona pesquisas futuras.

2. Revisão de Literatura

Na literatura, diversas abordagens podem ser encontradas para problemas relacionados com o CluOP-d, como o Problema de Roteamento de Veículos em *Cluster* (CluVRP, do inglês *Clustered Vehicle Routing Problem*), CTSP-d e *Team Orienteering Problem* (TOP), dentre outros. Com o intuito de auxiliar na resolução desses e outros problemas relacionados, alguns algoritmos exatos podem ser encontrados na literatura. Estes são baseados em estratégias de enumeração implícita (e.g. *Branch-and-bound*), e também em programação linear inteira mista (PLIM), onde o emprego de técnicas como geração de cortes ou geração de colunas são utilizadas, geralmente em conjunto com um método de *branch-and-bound* [Wolsey 1998].

Apesar dos recentes avanços na resolução exata, as estratégias heurísticas ainda são as mais empregadas na prática por explorarem o problema de forma inteligente de modo a gerar soluções rápidas e de boa qualidade, mas sem garantir a otimalidade. Uma outra forma de resolver problemas combinatórios é por meio de metaheurísticas, as quais possuem ferramentas que possibilitam escapar das armadilhas dos ótimos locais, permitindo a busca em outras regiões do espaço de busca. A utilização dessas técnicas possibilitou um grande avanço com relação à qualidade das soluções finais obtidas com os procedimentos heurísticos clássicos [Chaves et al. 2007].

Em [Battarra et al. 2014] é apresentada uma formulação compacta de programação inteira para o CluVRP e dois algoritmos de solução exata resultantes. A formulação baseia-se em um algoritmo de pré-processamento que simplifica o problema original, gerando um número polinomial de colunas em tempo exponencial. O primeiro é um algoritmo de corte que usa todas as colunas, enquanto o segundo algoritmo é um algoritmo de ramificação e corte e preço em que as colunas são geradas dinamicamente. Já [Vidal et al. 2015] apresentam para o mesmo problema, adaptações de bem conhecidos conceitos metaheurísticos, explorando os conhecimentos em Pesquisa Local Iterada (ILS, do inglês *Iterated Local Search*) e algoritmos genéticos híbridos.

Com a finalidade de resolver pequenas e médias instâncias do CTSP-d, [Hà et al. 2020] expõem em sua pesquisa um método exato, testando-o em instâncias de até 50 clientes. Os pesquisadores ainda propuseram uma metaheurística para lidar eficientemente com grandes instâncias de até 200 clientes, combinando ideias de *Greedy Randomized Adaptive Search Procedure* (GRASP), ILS e *Variable Neighborhood Descent* (VND).

Em sua pesquisa [El-Hajj et al. 2016] propuseram um algoritmo exato para resolver o TOP. Esse é baseado em uma formulação linear com um número polinomial de variáveis binarias. Ele é um algoritmo de plano de corte que explora soluções inteiras de modelos sucessivos, com as restrições de eliminação de subtour relaxadas no início e depois reforçadas iterativamente.

Para a solução do CluTOP, o trabalho de [Yahiaoui et al. 2019] apresenta um algoritmo exato baseado em planos de corte. Esse algoritmo inclui a implementação de um procedimento de pré-processamento juntamente com a consideração de desigualdades válidas. Esses componentes são introduzidos para reduzir a carga computacional do método exato. Os autores estenderam a heurística proposta inicialmente em [Yahiaoui et al. 2017]. Eles ainda propõem uma heurística hibrida que combina uma ALNS (do inglês, *Adaptive Large Neighborhood Search*) e um procedimento de divisão eficaz.

Conforme o acima exposto, verifica-se que o CluOP-d ainda não foi formalmente definido. Observa-se também que não existem abordagens na literatura para sua resolução, mas o uso das abordagens presentes nos problemas semelhantes pode auxiliar nesse processo por meios de adaptações desses modelos já existentes. Nas seções a seguir, apresenta-se uma formulação matemática de Programação Inteira para o CluOP-d, permitindo assim a aplicação prática deste problema.

3. Formulação matemática para o CluOP-d

Nesta pesquisa, utilizou-se uma adaptação do modelo proposto por [Hà et al. 2020], para o Problema do Caixeiro Viajante Clusterizado com regra de Prioridade Relaxada (CTSP-d). Neste modelo, foram realizadas adaptações quanto a implementação do uso de scores que são atribuídos a cada local e ainda uma limitação de tempo para que o percurso seja realizado, conforme o trabalho de [Yahiaoui et al. 2019].

Dado um conjunto de localidades e um limite de tempo para a viagem, o CluOP-d tem como objetivo estabelecer um trajeto ideal de modo a maximizar o prêmio total coletado. Nesse problema, os clientes são agrupados em subconjuntos chamados de clusters de acordo com a sua prioridade de visita e é definido um valor de *d* para que haja uma flexibilização entre a eficiência logística e as prioridades. Um prêmio é associado a cada localidade e esse será obtido apenas se o cliente for atendido.

O CluOP-d é definido formalmente da seguinte maneira: Seja $G = (V, A)$ um grafo direcionado, onde *V* é um conjunto de *n* localidades $V = \{0, 1, \dots, n\}$, e $V' = V \setminus V\{0\}$, em que 0 representa o depósito, cada aresta $e \in A$, com os dois pontos de extremidade definidos em V reproduzem as conexões entre os locais e tem um tempo de viagem t_{ij} associado, ou seja, $A = \{(i,j); i,j \in V, i \neq j\}$. O conjunto V é dividido em grupos disjuntos $K = \{V_1, V_2, \dots, V_m\}$, cada grupo V_m está associado a uma prioridade m, em que $V_i \subset V' : V_i \cap V_j = \emptyset; i,j = 1, \dots, m; i \neq j$. Um prêmio s_i é associado a cada localidade em V_m. A rota definida para o atendimento dos clientes deve ser percorrida durante um tempo de viagem limitado *T*.

Os parâmetros apresentados foram organizados a seguir:

T	→	Tempo total disponível;
$s_i, i \in V'$	→	Score da localidade $i \in V'$;
$\bar{t}_{ij}, (i,j) \in A$	→	Tempo de viagem do nó *i* para o nó *j*;
$M = \sum_{(i,j)\in A} \bar{t}_{ij}$	→	Constante Big-M;
d	→	Parâmetro que controla a regra d-Relaxada
ε	→	Número muito pequeno (ex.: 10-6);

Considere ainda as seguintes variáveis de decisão:

x_{ij}	→	Variável binaria igual a *1* se o veículo vai de i para j e *0* caso contrário;
y_i	→	Variável binaria igual a *1* se a localidade $i \in V'$ é visitada e *0* caso contrário;
$t_i \in R \geq 0$	→	Tempo de chegada na localidade $i \in V$.

A seguir será apresentado o modelo para o problema proposto:

$$max \sum_{i \in V'} s_i y_i \quad (1)$$

$$s.t. \sum_{\substack{i \in V \\ i \neq j}} x_{ij} = y_i, \qquad \forall j \in V \quad (2)$$

$$\sum_{\substack{j \in V \\ j \neq i}} x_{ij} = y_i, \qquad \forall i \in V \quad (3)$$

$$y_0 = 1 \quad (4)$$

$$t_0 = 0 \quad (5)$$

$$\sum_{(i,j) \in A} \bar{t}_{ij} x_{ij} \leq T \quad (6)$$

$$t_i + \bar{t}_{ij} - M(1 - x_{ij}) \leq t_j, \qquad \forall (i,j) \in A \quad (7)$$

$$t_i \leq t_j - \varepsilon, \qquad \forall i \in V_p, \forall j \in V_q; p, q \in K: q > p + d \quad (8)$$

$$x_{ij} \in \{0,1\}, \qquad \forall (i,j) \in A \quad (9)$$

$$y_i \in \{0,1\}, \qquad \forall i \in V \quad (10)$$

$$t_i \in R \geq 0, \qquad \forall i \in V \quad (11)$$

Na formulação acima, (1) representa a função objetivo que consiste em maximizar o prêmio total coletado s_i. (2) e (3) Exigem que o veículo entre e saia de cada localidade exatamente uma vez caso o mesmo seja visitado; (4) garante que o depósito seja visitado; (5) inicializa o serviço a partir do depósito, começando a contagem do tempo de viagem, ou seja, $t = 0$; (6) assegura que o tempo total não exceda o tempo limite disponível para a viagem *T*; (7) representa a relação entre *t* e $\bar{t}$, calculando o tempo de serviço inicial em um nó *j* com base em um nó *i* se o veículo viajar de *i* para *j*. Onde *M* é um número muito grande que deve ser maior que $t_i + \bar{t}_{ij}$, garantindo que o tempo de chegada em *j* seja sempre maior que o registrado para *i*; (8) modela a restrição d-relaxada, garantindo que as prioridades de cada cluster sejam respeitadas a medida em que às flexibilizam a partir do valor definido para *d*; (9), (10) e (11) definem as variáveis de decisão.

4. Experimentos computacionais

Essa seção é dedicada aos experimentos computacionais que foram realizados. Primeiro sendo mostrado como foram criadas as instâncias para o CluOP-d e em seguida os resultados apresentados nesses experimentos, por grupo de instância e posteriormente sendo analisada a influência que os resultados sofrem a partir dos parâmetros estabelecidos nas diferentes instâncias que foram desenvolvidas.

4.1. Criação das instâncias

As instâncias foram desenvolvidas de acordo com as instâncias TSP disponibilizadas em TSPLIB, acrescentando-se os valores dos *scores* atribuídos a cada localidade, o valor de *d* e ainda o tempo total disponível para o passeio. Para essas instâncias, os *scores* foram gerados de três formas distintas, acordo com o modelo proposto por [Fischetti 1998]:

Generation 1: $s_j := 1$;
Generation 2: $s_j := 1 + (7141 \cdot j + 73) mod (100)$;

Generation 3:$s_j := 1 + \left[99 \cdot \frac{t_{1j}}{\theta}\right], onde\ \theta := max_{i \epsilon V(1)} t_{1j}$

Com a finalidade de nomear as instâncias foi utilizado o padrão X-Y-K-*d*-α, onde X é o nome de TSPLIB original da instância, Y é o procedimento de distribuição dos *scores*, podendo ser *gen1*, *gen2* e *gen3*, K é o número de grupos, sendo definido em 5 valores, $K = \{1, 3, 5, 7, 9\}$. As instâncias ainda possuem *d* para modelar a regra de prioridade d-relaxada que pode ter os seguintes valores $d = \{0, 1, 3, 5\}$ e α que representa a base de cálculo do tempo limite de viagem, que pode ser calculado da seguinte forma: $T = CTSP * \alpha$, onde *CTSP* representa o custo da solução do problema original CTSP e α assume como valores possíveis $a = \frac{1}{4}$, $b = \frac{1}{2}$ e $c = \frac{3}{4}$.

A partir da aplicação de todos os parâmetros estabelecidos para o estudo, obteve-se para cada instância original TSPLIB, um quantitativo de 126 novas instâncias para o CluOP-d, o que chega a um total de 1.890 instâncias estudadas.

4.2. Resultados por grupo de instâncias

O modelo para o CluOP-d foi implementado utilizando-se o Pyomo, que é uma biblioteca de modelagem de otimização baseada em Python. Todos os experimentos foram realizados em um computador de 64 bits com processador AMD A8-5500B APU e o software de otimização CPLEX 12.9.0.0 com um tempo limitado de 1 hora de execução, para cada instância.

Primeiro analisou-se o desempenho da formulação em um conjunto de instâncias *gen1*, e posteriormente *gen2* e *gen3* sucessivamente. Os resultados preliminares dos experimentos encontram-se na Tabela 1, onde estão representados na coluna 1 os nomes dos grupos de instâncias, na 2 o percentual de soluções ótimas obtidas pelo grupo, nas colunas 3 e 4 estão os limites inferior e superior encontrados pelo CPLEX, sucessivamente, sendo expostas as médias para a execução das 42 variações de cada geração do grupo de instâncias. Ainda pode ser analisado na coluna 5, o GAP, que representa a distâncias entre os limites apresentados pelo solucionador, sendo calculado da seguinte forma, $GAP = \frac{UB-LB}{UB} \cdot 100$, e por fim, na coluna 6 é apresentado o tempo de execução em segundos. Os parâmetros analisados se repetem para cada geração de instâncias.

Tabela 1. Desempenho por grupo de instâncias.

	gen1					*gen2*					*gen3*				
Grupo de Instâncias	**Ótimas (%)**	**UB**	**LB**	**GAP (%)**	**Tempo (s)**	**Ótimas (%)**	**UB**	**LB**	**GAP (%)**	**Tempo (s)**	**Ótimas (%)**	**UB**	**LB**	**GAP (%)**	**Tempo (s)**
burma14	100%	10,38	10,38	0	0,68	100%	522,95	522,95	0	1,74	100%	483,55	483,55	0	1,20
ulysses16	100%	12,57	12,57	0	40,13	100%	661,86	661,86	0	30,26	100%	347,55	347,55	0	24,17
ulysses22	100%	18,76	18,76	0	164,41	88%	974,26	970,27	0,38	329,30	90%	489,92	488,02	0,34	421,25
att48	76%	36,59	36,06	1,19	1299,87	60%	1948,27	1915,65	1,53	1495,01	60%	1810,40	1748,36	3,51	1571,68
eil51	88%	37,16	36,97	0,37	518,41	81%	2000,86	1966,27	0,03	768,60	62%	2034,97	2027,00	0,38	1273,42
berlin52	81%	40,72	40,16	1,11	1101,91	62%	2258,76	2230,64	1,21	1578,39	48%	1643,76	1591,00	3,12	2360,43
st70	52%	54,52	50,70	6,19	2407,10	38%	3014,12	2882,91	4,17	2569,61	33%	3443,08	3348,07	2,63	2547,54
eil76	50%	56,42	53,93	4,09	1723,01	43%	3112,17	3022,20	1,67	1841,95	31%	3404,58	3226,89	4,87	2161,18
pr76	24%	63,32	53,45	14,29	2786,58	14%	3292,25	2812,75	13,64	2987,13	19%	3584,10	3139,72	11,90	3209,71
rat99	24%	84,28	61,47	25,24	3372,58	12%	4218,22	3445,26	17,18	3418,10	7%	5185,52	4098,81	20,04	3352,73
kroa100	0%	84,44	53,09	35,03	3603,48	2%	4535,76	3159,60	29,20	3620,80	0%	4891,81	3504,90	27,58	3602,34
krob100	7%	85,47	55,58	32,44	3369,27	7%	4566,67	3146,48	29,77	3419,88	2%	4241,70	3133,07	25,56	3579,40
kroc100	14%	83,97	54.03	32,54	3238,19	7%	4432,83	2995,54	31,68	3481,51	2%	5233,03	3726,71	28,03	3550,24
krod100	12%	79,84	54,15	28,12	3268,27	5%	4548,33	3139,82	30,04	3510,56	10%	4970,48	3438,44	29,68	3380,80
kroe100	10%	82,18	55,76	28,78	6240,17	2%	4473,52	3180,95	27,71	3562,12	0%	4595,80	3455,58	24,14	3602,35

Podemos observar que no geral as instâncias *gen1* são executadas mais rapidamente que as demais, ao mesmo tempo em que apresentam um percentual maior de soluções ótimas encontradas pelo solver. Isso pode ser explicado a partir das características de cada tipo de instância, enquanto *gen1* produz instâncias em que o objetivo é cobrir tantas localidades quanto possível, *gen2* produz instâncias com scores pseudoaleatórios no intervalo [1, 100], e *gen3* tem instâncias mais difíceis por atribuir grandes scores a localidades mais distantes do depósito.

Ao analisar as instâncias por seu tamanho, pode-se notar ainda que as que possuem até 52 localidades apresentam resultados satisfatórios com relação ao percentual de soluções ótimas encontradas, apresentando percentuais entre 48% e 100%, enquanto nas instâncias com número de cidades entre 70 e 99, vemos uma queda importante, quando é mostrado resultados no intervalo de 7% a 52%. Já as instâncias de 100 localidades apresentam índices baixíssimos de soluções ótimas encontradas pelo solver, com resultados variando entre 0% e 14%.

4.3. Influência dos parâmetros nos resultados

Com o intuito de possibilitar uma análise mais detalhada dos resultados obtidos nos experimentos computacionais preliminares, as instâncias foram separadas por tamanho, considerando-as como pequenas, médias ou grandes, de acordo com o número de cidades que possuem. Em seguida, foram escolhidas um grupo de instância para os tamanhos pequeno e médio, a fim de otimizar a visualização, sendo ulysses22 selecionada como pequena e eil51 como média, posteriormente, elaborou-se as Tabelas 2, 3 e 4. Essas tabelas apresentam os resultados alcançados para cada uma das 42 variações de cada grupo, separado por geração de instância escolhida, possibilitando a comparação dos resultados de acordo com a mudança dos parâmetros criados. Nelas os resultados estão reproduzidos de forma agrupada para cada valor possível de α disponível para o passeio, sendo na coluna 1 descrito os nomes das instâncias, na 2 o status da solução. Já as colunas seguintes estão dispostas de forma idêntica ao apresentado na Tabela 1.

Tabela 2. Influência dos parâmetros em instâncias *gen1*.

	$\alpha = 1/4$					$\alpha = 1/2$					$\alpha = 3/4$				
Instâncias	**Status**	**UB**	**LB**	**GAP (%)**	**Tempo (s)**	**Status**	**UB**	**LB**	**GAP (%)**	**Tempo (s)**	**Status**	**UB**	**LB**	**GAP (%)**	**Tempo (s)**
ulysses22-gen1-1-0	Ótimo	20	20	0	1346.12	Ótimo	22	22	0	0.75	Ótimo	22	22	0	0.34
ulysses22-gen1-3-0	Ótimo	11	11	0	1.91	Ótimo	18	18	0	15.73	Ótimo	21	21	0	1.20
ulysses22-gen1-3-1	Ótimo	14	14	0	23.95	Ótimo	21	21	0	111.12	Ótimo	22	22	0	4.63
ulysses22-gen1-5-0	Ótimo	12	12	0	1.00	Ótimo	19	19	0	0.63	Ótimo	21	21	0	0.68
ulysses22-gen1-5-1	Ótimo	15	15	0	10.57	Ótimo	20	20	0	134.31	Ótimo	22	22	0	22.44
ulysses22-gen1-5-3	Ótimo	17	17	0	2350.06	Ótimo	21	21	0	80.82	Ótimo	22	22	0	2.36
ulysses22-gen1-7-0	Ótimo	13	13	0	1.08	Ótimo	18	18	0	1.75	Ótimo	21	21	0	1.12
ulysses22-gen1-7-1	Ótimo	14	14	0	3.28	Ótimo	20	20	0	15.68	Ótimo	21	21	0	97.47
ulysses22-gen1-7-3	Ótimo	15	15	0	166.52	Ótimo	21	21	0	6.60	Ótimo	22	22	0	15.80
ulysses22-gen1-7-5	Ótimo	17	17	0	1836.50	Ótimo	21	21	0	64.08	Ótimo	22	22	0	1.10
ulysses22-gen1-9-0	Ótimo	13	13	0	0.31	Ótimo	17	17	0	0.33	Ótimo	20	20	0	0.31
ulysses22-gen1-9-1	Ótimo	14	14	0	1.04	Ótimo	19	19	0	3.84	Ótimo	21	21	0	2.55
ulysses22-gen1-9-3	Ótimo	16	16	0	8.06	Ótimo	21	21	0	10.30	Ótimo	22	22	0	14.33
ulysses22-gen1-9-5	Ótimo	17	17	0	532.35	Ótimo	21	21	0	8.95	Ótimo	22	22	0	3.41
eil51-gen1-1-0	Ótimo	18	18	0	3,12	Ótimo	34	34	0	3,21	Ótimo	47	47	0	37,14
eil51-gen1-3-0	Ótimo	19	19	0	32,85	Ótimo	33	33	0	23,42	Ótimo	44	44	0	22,60
eil51-gen1-3-1	Ótimo	22	22	0	71,81	Ótimo	17	17	0	21,38	T. Max	51	47	7,84	3600,87
eil51-gen1-5-0	Ótimo	23	23	0	27,36	Ótimo	37	37	0	59,93	Ótimo	48	48	0	119,04
eil51-gen1-5-1	Ótimo	27	27	0	56,01	-	-	-	-	-	Ótimo	51	51	0	3277,46
eil51-gen1-5-3	Ótimo	29	29	0	61,06	T. Max	51	48	5,88	3837,79	Ótimo	51	51	0	614,35
eil51-gen1-7-0	Ótimo	23	23	0	29,12	Ótimo	36	36	0	42,59	Ótimo	46	46	0	37,21
eil51-gen1-7-1	Ótimo	26	26	0	248,46	Ótimo	43	43	0	1326,65	Ótimo	51	51	0	464,78
eil51-gen1-7-3	Ótimo	30	30	0	314,10	-	-	-	-	-	Ótimo	51	51	0	913,39
eil51-gen1-7-5	Ótimo	31	31	0	229,75	-	-	-	-	-	Ótimo	51	51	0	39,92
eil51-gen1-9-0	Ótimo	22	22	0	6,13	Ótimo	35	35	0	8,29	Ótimo	45	45	0	19,81
eil51-gen1-9-1	Ótimo	26	26	0	40,76	Ótimo	43	43	0	481,25	Ótimo	51	51	0	262,06
eil51-gen1-9-3	Ótimo	29	29	0	1660,55	-	-	-	-	-	Ótimo	51	51	0	478,00
eil51-gen1-9-5	Ótimo	32	32	0	583,23	-	-	-	-	-	Ótimo	51	51	0	125,62

Tabela 3. Influência dos parâmetros em instâncias *gen2*.

	$\alpha = 1/4$					$\alpha = 1/2$					$\alpha = 3/4$				
Instâncias	**Status**	**UB**	**LB**	**GAP (%)**	**Tempo (s)**	**Status**	**UB**	**LB**	**GAP (%)**	**Tempo (s)**	**Status**	**UB**	**LB**	**GAP (%)**	**Tempo (s)**
ulysses22-gen2-1-0	Ótimo	1013	1013	0	900,67	Ótimo	1099	1099	0	0,37	Ótimo	1099	1099	0	0,17
ulysses22-gen2-3-0	Ótimo	672	672	0	2,21	Ótimo	968	968	0	4,02	Ótimo	1026	1026	0	20,88
ulysses22-gen2-3-1	Ótimo	799	799	0	15,39	Ótimo	1015	1015	0	58,23	Ótimo	1099	1099	0	4,24
ulysses22-gen2-5-0	Ótimo	706	706	0	1,66	Ótimo	965	965	0	0,97	Ótimo	1049	1049	0	1,04
ulysses22-gen2-5-1	Ótimo	801	801	0	61,42	Ótimo	1013	1013	0	238,28	Ótimo	1099	1099	0	20,73
ulysses22-gen2-5-3	-	-	-	-	-	T. Max	1075,62	1042	3,13	3603,12	Ótimo	1099	1099	0	2,99
ulysses22-gen2-7-0	Ótimo	756	756	0	0,59	Ótimo	955	955	0	1,18	Ótimo	1039	1039	0	1,06
ulysses22-gen2-7-1	Ótimo	781	781	0	4,54	Ótimo	1013	1013	0	10,16	Ótimo	1097	1097	0	90,12
ulysses22-gen2-7-3	Ótimo	847	847	0	269,80	Ótimo	1015	1015	0	122,08	Ótimo	1099	1099	0	14,02
ulysses22-gen2-7-5	T. Max	958,97	904	5,73	3600,53	T. Max	1097	1026	6,40	3600,66	Ótimo	1099	1099	0	0,59
ulysses22-gen2-9-0	Ótimo	767	767	0	0,39	Ótimo	930	930	0	0,40	Ótimo	1040	1040	0	0,31

ulysses22-gen2-9-1	Ótimo	814	814	0	0,60	Ótimo	991	991	0	4,54	Ótimo	1082	1082	0	5,38
ulysses22-gen2-9-3	Ótimo	874	874	0	31,73	Ótimo	1015	1015	0	40,41	Ótimo	1099	1099	0	2,69
ulysses22-gen2-9-5	Ótimo	905	905	0	425,07	-	-	-	-	-	Ótimo	1099	1099	0	8,75
eil51-gen2-1-0	Ótimo	1018	1018	0	76,54	Ótimo	1948	1948	0	7,10	Ótimo	2479	2479	0	54,35
eil51-gen2-3-0	Ótimo	1198	1198	0	3,74	Ótimo	1848	1848	0	48,87	Ótimo	2357	2357	0	43,18
eil51-gen2-3-1	Ótimo	1296	1296	0	165,98	-	-	-	-	-	-	-	-	-	-
eil51-gen2-5-0	Ótimo	1353	1353	0	35,07	Ótimo	2109	2109	0	46,25	Ótimo	2442	2442	0	66,96
eil51-gen2-5-1	Ótimo	1414	1414	0	1252,91	Ótimo	2365	2365	0	2266,51	-	-	-	-	-
eil51-gen2-5-3	Ótimo	1670	1670	0	45,88	-	-	-	-	-	Ótimo	2549	2549	0	196,53
eil51-gen2-7-0	Ótimo	1283	1283	0	27,56	Ótimo	1987	1987	0	44,49	Ótimo	2404	2404	0	38,25
eil51-gen2-7-1	Ótimo	1426	1426	0	484,27	T. Max	2271,68	2248	1	3600,68	Ótimo	2549	2549	0	220,43
eil51-gen2-7-3	Ótimo	1656	1656	0	2460,44	-	-	-	-	-	Ótimo	2549	2549	0	164,99
eil51-gen2-7-5	Ótimo	1830	1830	0	100,90	T. Max	2549	2542	0,27	4784,42	Ótimo	2549	2549	0	98,89
eil51-gen2-9-0	Ótimo	1280	1280	0	3,23	Ótimo	1964	1964	0	9,48	Ótimo	2409	2409	0	24,21
eil51-gen2-9-1	Ótimo	1462	1462	0	57,17	Ótimo	2288	2288	0	947,63	Ótimo	2549	2549	0	266,70
eil51-gen2-9-3	Ótimo	1576	1576	0	436,67	-	-	-	-	-	Ótimo	2549	2549	0	560,08
eil51-gen2-9-5	Ótimo	1787	1787	0	3401,89	Ótimo	2549	2549	0	3403,45	Ótimo	2549	2549	0	223,85

Tabela 4. Influência dos parâmetros em instâncias *gen3*.

	$\alpha = 1/4$					$\alpha = 1/2$					$\alpha = 3/4$				
Instâncias	**Status**	**UB**	**LB**	**GAP (%)**	**Tempo (s)**	**Status**	**UB**	**LB**	**GAP (%)**	**Tempo (s)**	**Status**	**UB**	**LB**	**GAP (%)**	**Tempo (s)**
ulysses22-gen3-1-0	Ótimo	565	565	0	1,32	Ótimo	565	565	0	1,34	Ótimo	565	565	0	0,34
ulysses22-gen3-3-0	Ótimo	407	407	0	23,15	Ótimo	407	407	0	22,95	Ótimo	527	527	0	10,44
ulysses22-gen3-3-1	Ótimo	465	465	0	13,39	Ótimo	465	465	0	13,30	Ótimo	565	565	0	14,24
ulysses22-gen3-5-0	Ótimo	413	413	0	1,05	Ótimo	413	413	0	0,99	Ótimo	529	529	0	1,31
ulysses22-gen3-5-1	Ótimo	461	461	0	686,49	Ótimo	461	461	0	546,94	Ótimo	565	565	0	17,19
ulysses22-gen3-5-3	T. Max	557,27	519	6,87	3600,92	T. Max	556,61	519	6,76	3600,94	Ótimo	565	565	0	0,45
ulysses22-gen3-7-0	Ótimo	392	392	0	1,21	Ótimo	392	392	0	1,22	Ótimo	492	492	0	2,02
ulysses22-gen3-7-1	Ótimo	441	441	0	98,50	Ótimo	441	441	0	98,71	Ótimo	555	555	0	60,45
ulysses22-gen3-7-3	Ótimo	475	475	0	2133,89	Ótimo	475	475	0	2143,51	Ótimo	565	565	0	4,72
ulysses22-gen3-7-5	-	-	-	-	-	-	-	-	-	-	Ótimo	565	565	0	2,25
ulysses22-gen3-9-0	Ótimo	391	391	0	0,57	Ótimo	391	391	0	0,40	Ótimo	491	491	0	0,34
ulysses22-gen3-9-1	Ótimo	419	419	0	1,49	Ótimo	419	419	0	1,53	Ótimo	527	527	0	6,56
ulysses22-gen3-9-3	Ótimo	477	477	0	21,02	Ótimo	477	477	0	20,89	Ótimo	565	565	0	5,69
ulysses22-gen3-9-5	Ótimo	515	515	0	1802,58	Ótimo	515	515	0	1881,97	Ótimo	565	565	0	3,74
eil51-gen3-1-0	Ótimo	1669	1669	0	97,91	Ótimo	1669	1669	0	92,02	Ótimo	2230	2230	0	496,31
eil51-gen3-3-0	Ótimo	1613	1613	0	18,69	Ótimo	1613	1613	0	16,08	Ótimo	2113	2113	0	6,49
eil51-gen3-3-1	Ótimo	1832	1832	0	2449,36	Ótimo	1832	1832	0	2450,07	-	-	-	-	-
eil51-gen3-5-0	Ótimo	1722	1722	0	80,32	Ótimo	1722	1722	0	73,06	Ótimo	2192	2192	0	148,26
eil51-gen3-5-1	T. Max	2109,24	2053	2,67	3600,85	T. Max	2109,34	2048	2,91	3600,86	Ótimo	2369	2369	0	1285,92
eil51-gen3-5-3	-	-	-	-	-	-	-	-	-	-	Ótimo	2369	2369	0	1910,77
eil51-gen3-7-0	Ótimo	1722	1722	0	19,83	Ótimo	1722	1722	0	19,86	Ótimo	2157	2157	0	56,11
eil51-gen3-7-1	T. Max	2008,83	1985	1,19	3600,94	T. Max	2008,83	1986	1,14	3600,92	Ótimo	2369	2369	0	315,61
eil51-gen3-7-3	-	-	-	-	-	-	-	-	-	-	Ótimo	2369	2369	0	277,87
eil51-gen3-7-5	-	-	-	-	-	-	-	-	-	-	Ótimo	2369	2369	0	19,03
eil51-gen3-9-0	Ótimo	1669	1669	0	10,41	Ótimo	1669	1669	0	10,46	Ótimo	2109	2109	0	18,15
eil51-gen3-9-1	T. Max	1986,32	1954	1,63	3600,87	T. Max	1986,36	1954	1,63	3600,86	Ótimo	2369	2369	0	330,80
eil51-gen3-9-3	-	-	-	-	-	-	-	-	-	-	Ótimo	2369	2369	0	3287,95
eil51-gen3-9-5	T. Max	2369	2335	1,43	4002,08	Ótimo	2369	2369	0	2726,26	Ótimo	2369	2369	0	197,90

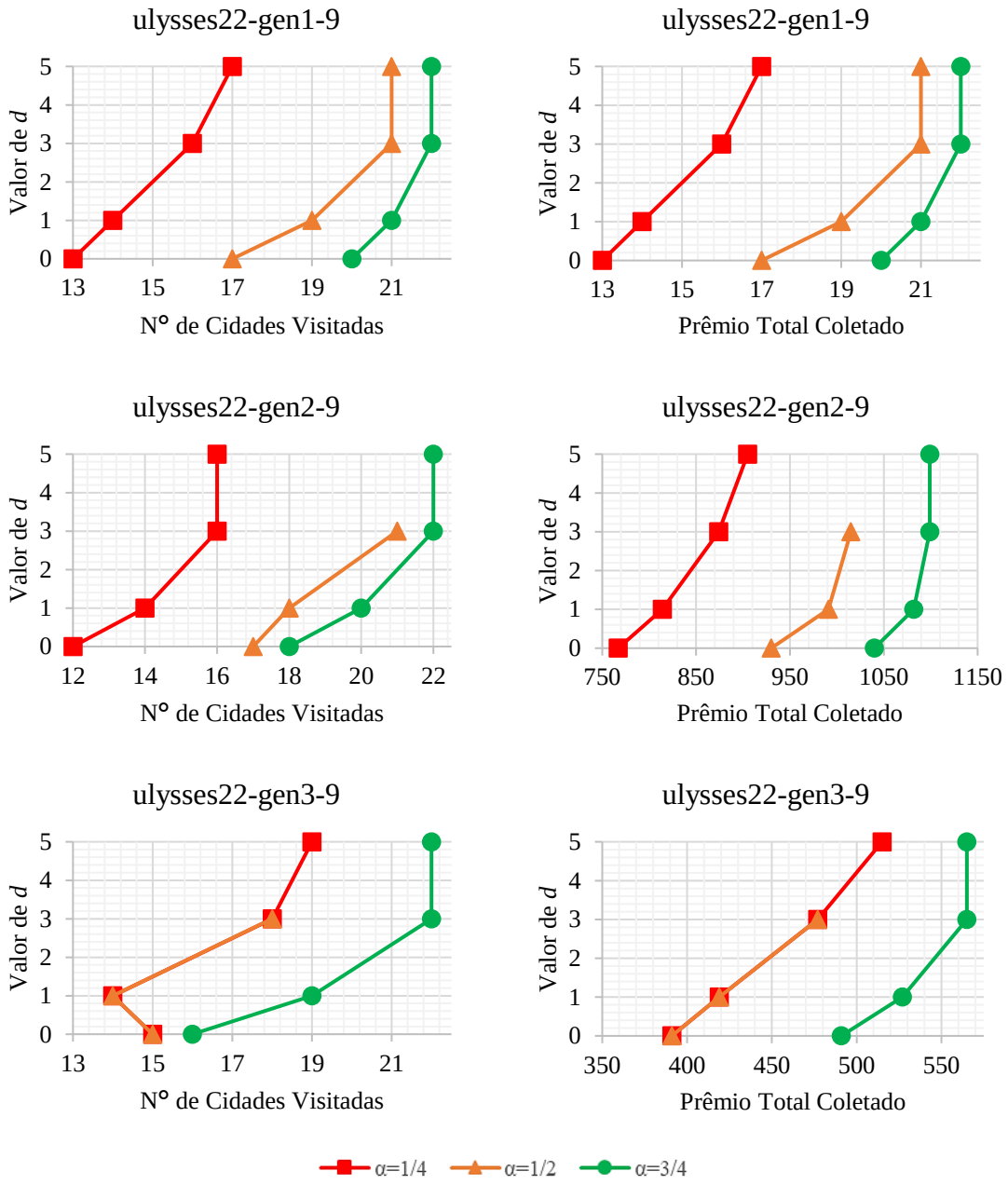
A partir dos resultados expostos nas Tabelas 2, 3 e 4, é possível observar a influência que os resultados sofrem pela mudança dos parâmetros estabelecidos, evidenciando a análise feita na Tabela 1, que mostrava que as instâncias do grupo *gen 1* apresentam um maior número de soluções ótimas. É possível observar ainda, que o α tem grande influência nesses resultados, podendo aumentar o número de cidades visitadas a medida em que ele aumenta. Existe ainda linhas em que não se tem resultados, isso acontece por que o solver não conseguiu resolver, interrompendo assim a execução da instância sem nenhum resultado obtido. Muitos desses casos acontecem justamente em instâncias com α nos valores de $\frac{1}{4}$ e $\frac{1}{2}$, justamente por esses valores limitarem ainda mais o tempo de passeio disponível, tornando assim o problema mais complexo de ser resolvido.

Com a finalidade de verificar a efetividade da regra de prioridade d-relaxada na otimização do resultado obtido pelo modelo, e a influência que o tempo disponível para a viagem tem no resultado, foram criados alguns gráficos que permitem uma melhor interpretação dos resultados. Os gráficos podem ser encontrados nas Figuras 1 e 2. Para o seu desenvolvimento, foram selecionados os grupos de instâncias com 9 clusters por permitirem uma comparação com todos os valores de *d* disponíveis, sendo escolhidas ulysses22 e eil51 por essas apresentarem uma quantidade de soluções ótimas que permitissem essa análise.

Primeiro foram investigadas as influências do valor de *d* e α no número de cidades visitadas e no prêmio total coletado para a solução das instâncias do grupo ulysses22, em todos os tipos de distribuição de scores (*gen1*, *gen2* e *gen3*) com 9 clusters. Os gráficos expõem no eixo

vertical a variação no valor de *d*, já no eixo horizontal na primeira coluna de gráficos encontra-se o número de cidades visitadas e na segunda coluna o prêmio total coletado para a rota estabelecida. Ainda é possível diferenciar o valor de α pela cor da linha de série e o formato do marcador do ponto. A legenda para cada α encontra-se na parte inferior da figura.

Figura 1. Influência de *d* e α em ulysses22.

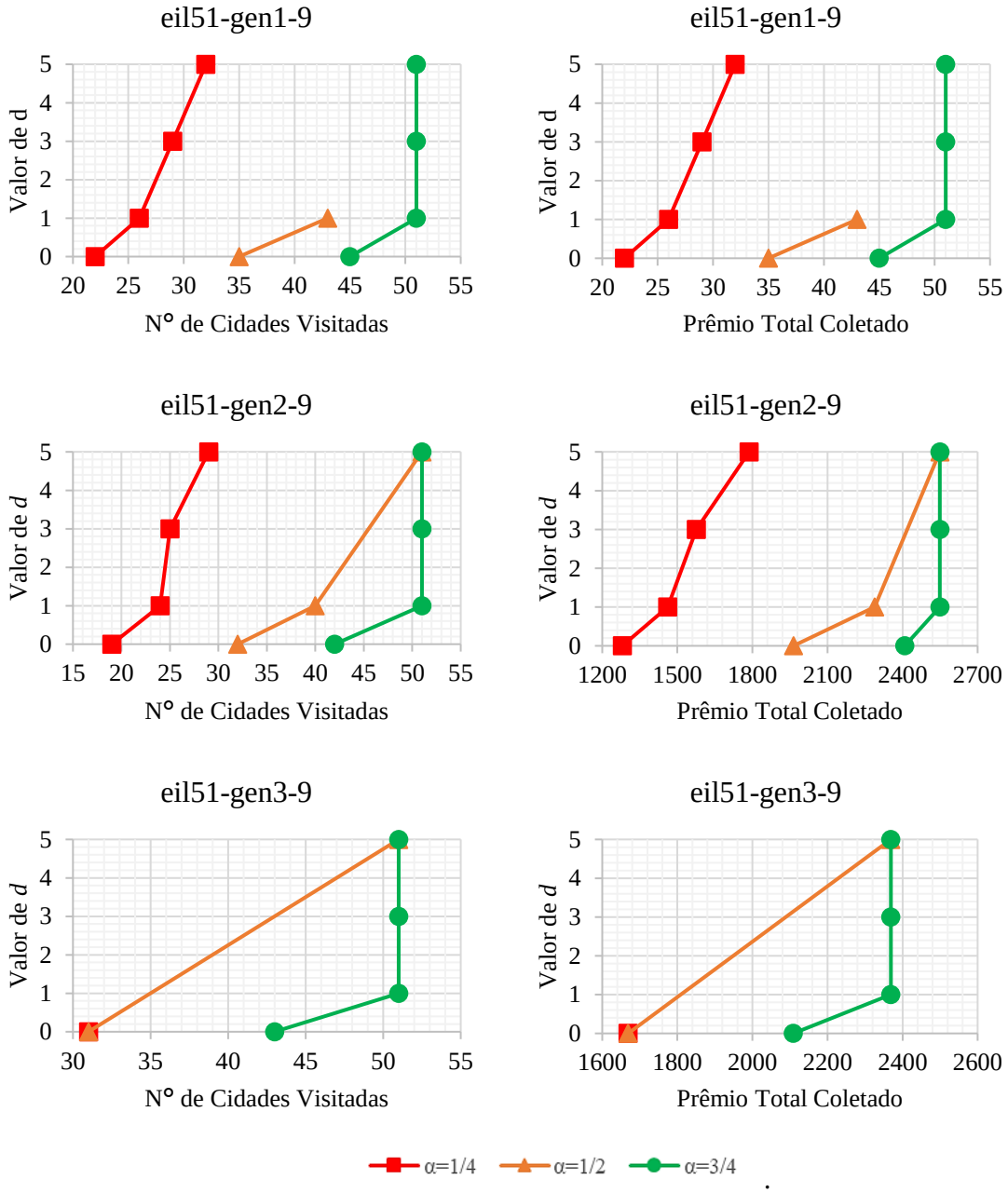


A partir dos gráficos expostos na Figura 1, representando as instâncias pequenas, pode-se observar claramente que no geral a medida em que o valor de *d* e α aumentam, o número de cidades visitadas e o prêmio total coletado tendem a crescer. É possível notar ainda, que no caso do grupo de instâncias ulysses22-gen2-9, com $\alpha = \frac{1}{4}$, quando o valor de *d* subiu de 3 para 5, o

número de cidade visitadas não aumentou, porém o prêmio coletado sim. Isso pode ser facilmente explicado pelo fato de que a medida em que a regra de prioridade é ainda mais relaxada, se expandem as opções de locais que podem ser visitados, ofertando assim uma maior possibilidade de encontrar prêmios superiores, mesmo sem aumentar o número de cidades visitadas, que acaba ficando limitado pelo tempo disponível, melhorando assim de qualquer forma o objetivo do passeio. Levando-se em consideração esses aspectos é possível comprovar a efetividade dos métodos utilizados para a obtenção de eficientes do ponto de vista logístico, ainda sem deixar de lado a prioridade de cada cluster.

Os gráficos na Figura 2, representam os resultados obtidos para as instâncias eil51, para as mesmas condições estabelecidas anteriormente para ulysses22.

Figura 2. Influência de *d* e α em eil51.



Com base na Figura 2, nota-se que apesar de algumas instâncias não terem sido resolvidas, a tendência de crescimento do número de cidades visitadas e do prêmio total coletado se mantem a medida em que se aumenta *d* e α para as instâncias médias. Observa-se ainda, que o aumento do valor de *d* a partir de 1 não é capaz de melhorar a solução quando $\alpha = \frac{3}{4}$. Isso pode ser explicado devido ao tempo disponível para o passeio já possibilitar o melhor resultado, independentemente de o valor de *d* ser maior ou igual a 1.

5. Conclusão

Este trabalho abordou o Problema de Orientação em *Clusters* com Regra de Prioridade d-Relaxada, sendo apresentada uma formulação capaz de resolver o problema. Para tal, foram utilizadas adaptações dos modelos do CluTOP e do CTSP-d. O método de solução utilizado foi o método exato por meio do software CPLEX.

A modelagem proposta contribuiu de forma eficaz no alcance dos resultados, visto que o uso da regra de prioridade d-relaxada se mostrou importante na obtenção de rotas ainda mais eficientes do ponto de vista logístico. Além do estabelecimento das rotas, também foram definidos os prêmios coletados e o tempo da rota percorrida, o que pode contribuir de forma muito importante na aplicação para os problemas de logística do dia-a-dia, como no caso da Logística Humanitária.

Os experimentos computacionais preliminares mostraram um melhor desempenho nas instâncias de até 52 localidades em relação as demais que vão até 100, pois estas obtiveram na sua maioria soluções consideradas ótimas, possibilitando o estabelecimento das rotas e seus respectivos tempo de viagem. Isso permite analisar que ainda há bastante a melhorar com relação à formulação em pesquisas futuras, para que essas instâncias com maior número de localidades que em alguns casos não obtiveram nenhuma solução ótima, possam apresentar resultados mais satisfatórios.

Para um melhor direcionamento de pesquisas futuras, deve-se adicionar desigualdades válidas que venham a ser eficientes, algoritmos de *branch-and-cut*, diminuir os Gaps apresentados e ainda resolver instâncias maiores.

6. Agradecimentos

Ao programa CNPq/PIBIC pelo financiamento do projeto de pesquisa, e à UFCG pela concessão da bolsa de Iniciação Científica.

O presente trabalho foi realizado com apoio do CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico – Brasil, Edital UFCG/CNPQ PIBIC 2019-2020.

Referências

APPLEGATE, David L.; BIXBY, Robert E.; CHVÁTAL, Vašek; COOK, William J.. **The Traveling Salesman Problem:** A Computational Study. Princeton, New Jersey: Princeton University Press, 2007. 608 p.

BATTARRA, Maria; ERDOğAN, Güneş; VIGO, Daniele. Exact Algorithms for the Clustered Vehicle Routing Problem. **Operations Research**, [s.l.], v. 62, n. 1, p. 58-71, fev. 2014. Institute for Operations Research and the Management Sciences (INFORMS). http://dx.doi.org/10.1287/opre.2013.1227.

CHAVES, Antonio Augusto; BIAJOLI, Fabrício Lacerda; MINE, Otávio Massashi; SOUZA, Marcone Jamilson Freitas. Metaheurísticas híbridas para resolução do problema do caixeiro viajante com coleta de prêmios. **Production**, [s.l.], v. 17, n. 2, p. 263-272, ago. 2007. FapUNIFESP (SciELO). http://dx.doi.org/10.1590/s0103-65132007000200004.

EL-HAJJ, Racha; DANG, Duc-cuong; MOUKRIM, Aziz. Solving the team orienteering problem with cutting planes. **Computers & Operations Research**, [s.l.], v. 74, p. 21-30, out. 2016. Elsevier BV. http://dx.doi.org/10.1016/j.cor.2016.04.008.

FISCHETTI, Matteo; GONZÁLEZ, Juan José Salazar; TOTH, Paolo. Solving the Orienteering Problem through Branch-and-Cut. **Informs Journal On Computing**, [S.L.], v. 10, n. 2, p. 133-148, maio 1998. Institute for Operations Research and the Management Sciences (INFORMS). http://dx.doi.org/10.1287/ijoc.10.2.133.

HÀ, Minh Hoàng; PHUONG, Hoa Nguyen; NHAT, Huyen Tran Ngoc; LANGEVIN, André; TRÉPANIER, Martin. Solving the clustered traveling salesman problem with d-relaxed priority rule. **International Transactions In Operational Research**, [S.L.], p. 1-17, 11 abr. 2020. Wiley. http://dx.doi.org/10.1111/itor.12797.

PANCHAMGAM K.V. **Essays in Retail Operations and Humanitarian Logistics**. Ph.D thesis, University of Maryland College Park, 2011.

VIDAL, Thibaut; BATTARRA, Maria; SUBRAMANIAN, Anand; ERDOG˘AN, Güneş. Hybrid metaheuristics for the Clustered Vehicle Routing Problem. **Computers & Operations Research**, [s.l.], v. 58, p. 87-99, jun. 2015. Elsevier BV. http://dx.doi.org/10.1016/j.cor.2014.10.019.

WOLSEY, L. A. **Integer Programming**, John Wiley & Sons, 1998. 288 p.

YAHIAOUI, Ala-eddine; MOUKRIM, Aziz; SERAIRI, Mehdi. The clustered team orienteering problem. **Computers & Operations Research**, [s.l.], v. 111, p. 386-399, nov. 2019. Elsevier BV. http://dx.doi.org/10.1016/j.cor.2019.07.008.
Anna, A. (2006). Artigo em conferência. In *Anais do XXVIII SBPO*, p. 123–134, Rio de Janeiro. SOBRAPO.