O Problema Da Diversidade Máxima de Grupos: uma abordagem de programação linear inteira

Carlos V. D. Araujo[1], **Tatiane F. Figueiredo**[2]

[1]Universidade Federal do Ceará (UFC) – Campus Russas
CEP – 62.900-000 – Russas – CE

victor-araujo@alu.ufc.br, tatianefernandes@lia.ufc.br

Abstract. *In the Maximally Diverse Grouping Problem, are given as input data a set of n elements and a function that provides the diversity degree between pair of elements. The problem is NP-hard and it aims at partitioning the set of elements into smaller subsets, called groups, in such a way that the diversity between the elements belonging to each group is as large as possible. In this paper, we present an integer linear programming model for this problem. The results from computational experiments show that the proposed model performs better when compared with the quadratic one from the literature.*

Resumo. *No Problema da Diversidade Máxima de Grupos, são dados como entrada um conjunto de n elementos e uma função que informa o valor da diversidade entre pares de elementos. O problema é NP-difícil e tem como objetivo particionar o conjunto de elementos em subconjuntos menores, chamados grupos, de maneira que a diversidade entre os elementos em cada grupo seja a maior possível. Neste artigo, apresentamos um modelo em programação linear inteira o problema. Os resultados de experimentos computacionais mostram que o modelo proposto tem um melhor desempenho em comparação ao modelo quadrático existente na literatura.*

1. Introdução

O *Problema da Diversidade Máxima de Grupos* (PDMG) consiste em encontrar um particionamento para um conjunto de n elementos, ou seja, m subconjuntos, de modo que a *diversidade* entre os elementos em cada subconjunto seja a maior possível. A *diversidade* entre os elementos de um subconjunto é calculada como a soma da distância individual entre cada par de elementos, sendo os m subconjuntos, em muitas aplicações, também denominados *grupos* e seus respectivos elementos denominados *indivíduos*. O objetivo do PDMG é maximizar a diversidade geral, definida como o somatório do resultado da diversidade dos m grupos.

Várias são as aplicações do PMDG, como por exemplo, projetos de circuitos VLSI [Chen 1986] e armazenamento de memória paginada [Kral, J. 1965]. Uma das mais populares aplicações do PDMG aparece no contexto acadêmico, quando formamos grupos de estudantes [O'Brien and Mingers 1995]. A situação também pode ser adaptada para a criação de grupos de revisores para publicações científicas ou para avaliação de projetos em agências de fomento à ciência [Hettich and Pazzani 2006], criando diversos grupos onde pessoas com conhecimentos e experiências diferentes lidem com a heterogeneidade facilitando a comunicação entre elas [Bhadury et al. 2000].

https://proceedings.science/p/85427?lang=pt-br

O problema da Diversidade Máxima de Grupos também é conhecido como Problema das k-partições(*k-partition Problem* [Feo et al. 1992]), ou Problema de Partição Equilibrada (*Equitable Partition Problem* [O'Brien and Mingers 1995]). Sendo o Problema da Diversidade Máxima (*Maximum Diversity Problem*, [Kuo et al. 1993]), onde o número de subconjuntos a serem encontrados é sempre igual a 1, o caso mais explorado da literatura. Em [Feo T. and Khellaf 1990] foi provado que o PDMG é NP-Difícil. Problemas de otimização desta natureza são caracterizados por grandes espaços de soluções, dificultando a obtenção de uma solução de custo ótimo em tempo computacional razoável, motivo pelo qual encontra-se na literatura um número extenso de soluções baseadas apenas em algoritmos heurísticos.

No melhor do nosso conhecimento, somos os primeiros a formular um modelo de programação linear inteira para o PDMG, existindo na literatura apenas um modelo quadrático para o problema, proposto por [Fan et al. 2011]. Neste trabalho, apresentamos o modelo de programação linear inteira para o PDMG e uma análise dos resultados obtidos através de extensos experimentos computacionais realizados em instâncias da literatura. Os resultados da análise indicam que nosso modelo supera ou é competitivo com o modelo quadrático existente para o PDMG.

2. Trabalhos Anteriores

Diversas abordagens heurísticas e metaheurísticas têm sido propostas para obter soluções para o PDMG. [Fan et al. 2011] abordaram o PDMG através de um algoritmo híbrido baseado em algoritmos genéticos. Esta abordagem mostrou-se satisfatória encontrando-se bons resultados para instâncias de tamanho máximo n = 240.

[Gallego et al. 2013] propuseram uma nova heurística utilizando mecanismos de busca, com base na metodologia busca tabu para o PDMG, incluindo uma estratégia de oscilação, que permitiu o cruzamento de caminhos viáveis de busca. Foram avaliados procedimentos de construção e de busca para configurar um procedimento de solução que quando comparado ao estado da arte para o PDMG. Extensos experimentos computacionais com médias e grandes instâncias foram apresentados para demonstrar as vantagens do método de solução que inclui estratégica de oscilação.

[Palubeckis et al. 2011] propuseram três heurísticas para o PDMG, sendo elas: Multistart simulated annealing (MSA), Hybrid steady-state genetic algorithm (HGA) e Variable Neighborhood search (VNS). O desempenho das heurísticas foi comparado computacionalmente através de dois grupos de instâncias: o primeiro, instâncias do Problema da Diversidade Máxima, caso particular do PDMG e o outro, com instâncias de até 2000 elementos. Dentre as três abordagens, a que se mostra mais eficiente é o HGA.

[Uroševié 2014] desenvolveu uma heurística híbrida baseada em VNS onde o processo de busca utilizado para melhoria de soluções baseou-se em um algoritmo VND composto por duas vizinhanças. Os resultados apresentados mostram que a metodologia proposta é melhor que os resultados obtidos com o VNS clássico. Um algoritmo Artificial Bee Colony (ABC), metodologia baseada no comportamento alimentar das abelhas, foi proposto por [Rodriguez et al. 2013]. A eficiência do novo algoritmo foi comprovada para instâncias de até 2000 elementos e, comparado aos seus antecessores, este algoritmo apresentou as melhores soluções até o presente momento.

Diante da análise feita a partir de trabalhos anteriores, é notório a extensa

https://proceedings.science/p/85427?lang=pt-br

utilização de abordagens heurísticas para obtenção de soluções para o PDMG. Fato que motiva a criação de modelos de programação linear inteira para o problema, uma vez que, além de inexistentes na literatura, algoritmos exatos garantem a obtenção de soluções ótimas, ao contrário dos algoritmos heurísticos. Em contrapartida, em termos gerais, algoritmos heurísticos obtém soluções consideradas aceitáveis utilizando um tempo razoavelmente menor em comparação com modelos exatos.

3. Problema da Diversidade Máxima de Grupos

Na literatura é possível encontrar duas variantes do PDMG. O problema clássico define que, dado um conjunto de n indivíduos, deve-se encontrar m grupos, todos possuindo exatamente k indivíduos, tal que $k = \lfloor n/m \rfloor$, garantindo que a diversidade destes m grupos seja a máxima possível. A segunda variação, foco deste trabalho, permite que os grupos tenham tamanhos variados, sendo portanto, uma generalização do PDMG clássico.

Considerando a definição formal do PDMG, podemos representar parte de sua entrada através de um grafo. Seja $G = (V, E)$ um grafo não-orientado, onde o conjunto de vértices V representa o conjunto de indivíduos; e conjunto de arestas E representa a diversidade de cada par de indivíduos do conjunto V, definimos a entrada do PDMG como uma 5-upla $(G(V, E), d, a, b, m)$, onde:

- $G = (V, E)$ é um grafo não-orientado, onde V representa os indivíduos que serão agrupados e E as diversidades entre pares indivíduos pertencentes a V;
- $d : E \to \mathbb{N}$ é uma função que retorna o valor da diversidade para cada par de indivíduos $(u, v) \in E$;
- m é uma constante que indica a quantidade de grupos que devem ser formados;
- $a : g \to \mathbb{N}$ é uma função, onde a_g é o valor mínimo de indivíduos permitidos no grupo g, para todo $g = \{1, 2, ..., m\}$;
- $b : g \to \mathbb{N}$ é uma função, onde b_g é o valor máximo de indivíduos permitidos no grupo g, para todo $g = \{1, 2, ..., m\}$.

3.1. Modelo de Programação Linear Inteira

Dada uma instância $I_{MDGP} = (G(V, E), d, a, b, m)$ para o problema PDMG, considere as seguintes variáveis.

Para cada indivíduo $i \in V(G)$ e para cada grupo $g = \{1, ..., m\}$, introduzimos uma variável binária X_{ig}, relacionando o indivíduo i ao grupo g, onde:

$$x_{ig} = \begin{cases} 1, & \text{se o indivíduo } i \text{ está no grupo } g, \\ 0, & \text{caso contrário.} \end{cases} \quad \forall i \in V(G), \forall g = \{1, ..., m\}. \quad (1)$$

Para cada aresta $ij \in E(G)$ e para cada grupo $g = \{1, ..., m\}$, definimos uma variável binária y_{ijg} para indicar os indivíduos i e j estão no mesmo grupo g , onde:

$$y_{ijg} = \begin{cases} 1, & \text{se } i \text{ e } j \text{ pertecem ao mesmo grupo g,} \\ 0, & \text{caso contrário.} \end{cases} \quad \forall ij \in E(G), \forall g = \{1, ..., m\}. \quad (2)$$

Um modelo de programação linear inteira para o PDMG é então:

$$\max \quad d(x,y) = \sum_{g=1}^{m} \sum_{\forall i \in V} \sum_{\substack{\forall j \in V \\ i \neq j}} d_{ij} y_{ijg} \tag{3}$$

$$\text{s.a:} \quad \sum_{g=1}^{m} x_{ig} = 1, \qquad \forall i \in V \tag{4}$$

$$\sum_{\forall i \in V} x_{ig} \geq a_g, \qquad \forall g = \{1, 2, \ldots, m\} \tag{5}$$

$$\sum_{\forall i \in V} x_{ig} \leq b_g, \qquad \forall g = \{1, 2, \ldots, m\} \tag{6}$$

$$x_{ig} + x_{jg} - 1 \leq y_{ijg}, \qquad \forall g = \{1, 2, \ldots, m\}, \forall ij \in E, i \neq j \tag{7}$$

$$\sum_{\substack{\forall j \in V \\ i \neq j}} y_{ijg} \geq (a_g - 1) x_{jg}, \qquad \forall j \in V, \forall g = \{1, 2, \ldots, m\} \tag{8}$$

$$\sum_{\substack{\forall i \in V \\ i \neq j}} y_{ijg} \leq (b_g - 1) x_{jg}, \qquad \forall j \in V, \forall g = \{1, 2, \ldots, m\} \tag{9}$$

$$x_{ig} \in \{0, 1\}, \qquad \forall i \in V(G), \forall g = \{1, ..., m\} \tag{10}$$

$$y_{ijg} \in \{0, 1\}, \qquad \forall ij \in E(G), \forall g = \{1, ..., m\} \tag{11}$$

As restrições 4 garantem que todo indivíduo $i \in V$ deverá estar em exatamente uma equipe g, $\forall g = \{1, ..., m\}$, as restrições 5 e 6 obrigam que o número de indivíduos, em uma equipe g, seja no mínimo a_g e no máximo b_g para todo $g = \{1, ..., m\}$, respectivamente. Restrições do tipo 7 exigem que caso dois indivíduos i e j estejam em uma mesma equipe g então o valor da variável y_{ijg} deverá ser igual valor 1, ou seja, a variável y_{ijg} relaciona os pares de indivíduos de uma equipe g. As restrições 8 e 9 garantem que cada indivíduo estará se relacionando em uma equipe g com no mínimo a_g indivíduos e no máximo b_g indivíduos, respectivamente. Por fim, as restrições 10 e 11 garantem que as variáveis sejam binárias.

4. Resultados Computacionais

Nessa seção são descritos os experimentos computacionais realizados para testar a eficiência do modelo linear inteiro proposto neste artigo em comparação com o modelo quadrático proposto por [Fan et al. 2011]. A implementação dos algoritmos exatos faz uso do pacote de otimização ILOG CPLEX versão 12.6.1. Neste pacote está presente a biblioteca denominada *Concert Technology for C++*, que permite o desenvolvimento de modelos matemáticos utilizando a linguagem de programação C++. Os experimentos realizados foram executados utilizando uma máquina com processador Intel Core I5-4570 (3.20GHz) 4 threads com 14.6 GB RAM e sistema operacional Linux Mint 18 Cinnamon 64 bits.

Os experimentos foram realizados sobre uma amostra de 240 instâncias, esse conjunto de instâncias de *benchmark*, referido como MDGPLIB, é disponibilizado em `http://www.optsicom.es/mdgp`. A seguir apresentamos os principais dados das 240 instâncias que foram divididas em três subconjuntos:

https://proceedings.science/p/85427?lang=pt-br

- *RanReal:* 80 instâncias compostas de matrizes $D_{n\times n}$ em que os valores d_{ij} da matriz D representam a diversidade entre um par de indivíduos (i, j), sendo d_{ij} valores reais gerados utilizando uma distribuição uniforme $U(0, 100)$;
- *RanInt*: Possui a mesma estrutura das instâncias do tipo *RanReal*, porém d_{ij} são valores inteiros gerados por uma distribuição uniforme *U*(0, 100);
- *Geo*: Também possui a mesma estrutura das instâncias dos outros dois conjuntos já mencionados, entretanto, como proposto por [Glover et al. 1998], os valores de d_{ij} são calculados como distância Euclidiana entre pares de pontos com coordenadas aleatoriamente geradas no intervalo $[0, 10]$. O número de coordenadas para cada instância é aleatoriamente gerada no intervalo $[2, 21]$.

O número de elementos n, o número de grupos m e os limites de cada grupo a_g e b_g são apresentados na Tabela 1, existindo 20 instâncias para cada combinação de parâmetros, onde 10 instâncias representam casos em que todos os grupos devem ter o mesmo tamanho e 10 instâncias representam casos em que os grupos possuem tamanhos diferentes. Sendo assim, no primeiro caso, $a_g = b_g = \lfloor n/m \rfloor$. Já no segundo caso, os limites para o tamanho de cada grupo (a_g e b_g) foram gerados aleatoriamente em um intervalo pré-definido onde: a_g é gerado no intervalo [${a_g}^{min}$, ${a_g}^{max}$] e b_g no intervalo [${b_g}^{min}$, ${b_g}^{max}$]. Este conjunto de instâncias foi introduzido por [Fan et al. 2011], com valores de n variando entre 10 a 240. [Gallego et al. 2013] apresentam instâncias maiores, com valores de n variando de 480 a 960, utilizando o mesmo padrão criado por [Fan et al. 2011].

Tabela 1. Sumário dos parâmetros das 240 instâncias

n	m	$a_g = b_g$	${a_g}^{min}$	${a_g}^{max}$	${b_g}^{min}$	${b_g}^{max}$
10	2	5	3	5	5	7
12	4	3	2	3	3	5
30	5	6	5	6	6	10
60	6	10	7	10	10	14

Utilizando as 240 instâncias apresentadas, efetuamos experimentos computacionais para duas implementações de algoritmos exatos. O primeiro, denominado *Lin*, é a implementação do modelo linear inteiro, proposto neste artigo. O segundo algoritmo exato, denominado *Quad*, é a implementação do modelo quadrático proposto por [Fan et al. 2011]. O objetivo dessa comparação é apresentar qual das implementações possui um melhor desempenho, considerando o tempo de execução e utilização de memória. O tempo limite para resolução de cada instância, em ambos algoritmos, foi pré-definido com o valor de 3600 segundos, limitando a utilização de memória em 26000 MB.

As tabelas 2 e 3 apresentam os resultados dos experimentos para as instâncias onde ambos algoritmos obtiveram a solução ótima dentro do tempo e utilização de memória pré-definidos. Mais especificamente, a tabela 2 apresenta os resultados para as instâncias onde os grupos possuem tamanhos diferentes (60 instâncias) e a tabela 3 os resultados dos experimentos para instâncias onde os grupos possuem tamanhos iguais (60 instâncias). Nestas tabelas, estão presentes a coluna de identificação da instância solucionada (*RanReal*, *RanInt* ou *Geo*), assim como o valor da solução ótima para cada algoritmo (*Lin*, *Quad*) é apresentado: o tempo de execução (em segundos) e o número de nós gerados na árvore de solução, que é informado pelo *solver*.

https://proceedings.science/p/85427?lang=pt-br

Tabela 2. Comparação entre os algoritmos para instâncias onde os grupos possuem tamanhos diferentes e $n \leq 12$

n	Instância	Valor ótimo	Tempo(s)		Nós	
			LIN	QUAD	LIN	QUAD
10	**RanReal_01**	1437,81	0,24	0,1	0	0
	RanReal_02	1202,09	0,22	0,08	0	0
	RanReal_03	1292,79	0,05	0,19	0	0
	RanReal_04	1285,91	0,24	0,1	0	0
	RanReal_05	1574,6	0,24	0,09	0	0
	RanReal_06	1153,93	0,06	0,13	0	0
	RanReal_07	1402,68	0,19	0,08	0	0
	RanReal_08	1267,58	0,09	0,17	0	0
	RanReal_09	1420,13	0,15	0,14	0	0
	RanReal_10	1381,34	0,16	0,07	0	0
12	**RanReal_01**	1050,35	19,61	3,5	1469	5713
	RanReal_02	1093,06	3,8	15,02	3060	14728
	RanReal_03	1011,77	0,15	4,63	0	5343
	RanReal_04	1248,46	4,15	12,63	3467	14726
	RanReal_05	1225,48	10,64	12,84	5737	13216
	RanReal_06	1036,79	5,09	15,99	4417	16004
	RanReal_07	1207,33	16,46	13,21	12027	18702
	RanReal_08	921,411	1,65	14,26	890	13421
	RanReal_09	980,793	16,96	10,48	10770	10715
	RanReal_10	1235,08	6,29	15,3	5959	19426
10	**RanInt_01**	1325	0,19	0,08	0	0
	RanInt_02	1204	0,25	0,06	0	0
	RanInt_03	1374	0,06	0,12	0	0
	RanInt_04	1308	0,2	0,1	0	0
	RanInt_05	1349	0,27	0,06	0	0
	RanInt_06	1228	0,14	0,18	0	0
	RanInt_07	1221	0,18	0,11	0	0
	RanInt_08	1208	0,12	0,16	0	0
	RanInt_09	1278	0,08	0,09	0	0
	RanInt_10	1268	0,22	0,07	0	0
12	**RanInt_01**	1059	26,92	2,74	21313	4465
	RanInt_02	978	2,83	16,56	2260	15616
	RanInt_03	993	0,29	12,36	0	11785
	RanInt_04	1139	12,81	12,59	9415	17640
	RanInt_05	1199	14,58	20,82	9737	23745
	RanInt_06	1112	2,38	3,66	1466	5850
	RanInt_07	1248	14,18	7,8	9359	8180
	RanInt_08	1010	1,46	2,32	267	3088
	RanInt_09	1080	24,82	12,47	21169	17115
	RanInt_10	1173	2,09	2,79	859	4262
10	**Geo_01**	3864,69	0,28	0,29	0	893
	Geo_02	1404,8	0,24	0,16	0	576
	Geo_03	1662,12	0,15	0,12	0	403
	Geo_04	3678,98	0,32	0,23	0	918
	Geo_05	2452,55	0,24	0,2	0	765
	Geo_06	1161,77	0,11	0,1	0	342
	Geo_07	3206,54	0,26	0,24	0	926
	Geo_08	2243,15	0,18	0,14	0	548
	Geo_09	3828,65	0,09	0,17	0	476
	Geo_10	4559,18	0,16	0,16	0	645
12	**Geo_01**	807,677	75,76	11,2	74671	17778
	Geo_02	921,548	21,12	25,31	23559	46512
	Geo_03	2368,84	0,34	28,2	0	43977
	Geo_04	2041,37	79,86	51	79223	105225
	Geo_05	2684,17	60,53	117,9	64312	232736
	Geo_06	1612,75	51,25	44,94	49000	83448
	Geo_07	1821,88	148,26	78,87	184539	185217
	Geo_08	2481,76	103,38	36,07	109453	58074
	Geo_09	1940,36	212,51	72,17	226563	139568
	Geo_10	984,986	24,12	26,2	23774	43802

https://proceedings.science/p/85427?lang=pt-br

Tabela 3. Comparação entre os algoritmos para instâncias onde os grupos possuem tamanhos iguais e $n \leq 12$

n	Instância	Valor ótimo	Tempo(s)		Nós	
			LIN	QUAD	LIN	QUAD
10	**RanReal_01**	1427,85	0,04	0,14	0	0
	RanReal_02	1202,09	0,04	0,08	0	0
	RanReal_03	1292,79	0,05	0,16	0	0
	RanReal_04	1263,77	0,05	0,18	0	0
	RanReal_05	1385,65	0,05	0,08	0	0
	RanReal_06	1153,93	0,05	0,12	0	0
	RanReal_07	1236,48	0,18	0,15	0	0
	RanReal_08	1267,58	0,05	0,16	0	0
	RanReal_09	1420,13	0,2	0,17	0	0
	RanReal_10	1195,92	0,03	0,13	0	0
12	**RanReal_01**	956,43	0,26	2,48	0	1384
	RanReal_02	972,55	0,37	2,4	0	1387
	RanReal_03	1011,77	0,13	2,43	0	1119
	RanReal_04	986,692	0,26	2,09	0	1652
	RanReal_05	993,349	0,13	2,23	0	1110
	RanReal_06	937,348	0,5	2,22	0	1489
	RanReal_07	977,853	0,32	2,87	0	860
	RanReal_08	869,724	0,21	2,03	0	1025
	RanReal_09	881,723	0,32	1,88	0	847
	RanReal_10	1031,01	0,08	2,46	0	1295
10	**RanInt_01**	1292	0,06	0,14	0	0
	RanInt_02	1204	0,1	0,14	0	0
	RanInt_03	1374	0,05	0,1	0	0
	RanInt_04	1201	0,06	0,12	0	0
	RanInt_05	1208	0,04	0,14	0	0
	RanInt_06	1228	0,15	0,15	0	0
	RanInt_07	1140	0,14	0,16	0	0
	RanInt_08	1208	0,05	0,16	0	0
	RanInt_09	1278	0,06	0,13	0	0
	RanInt_10	1112	0,1	0,13	0	0
12	**RanInt_01**	985	0,18	2,28	0	1030
	RanInt_02	856	0,22	2,29	0	996
	RanInt_03	993	0,39	2,17	0	1103
	RanInt_04	937	0,46	2,18	0	1238
	RanInt_05	1003	0,24	1,64	0	714
	RanInt_06	1000	0,13	1,92	0	700
	RanInt_07	993	0,34	2,08	0	999
	RanInt_08	863	0,37	1,58	0	494
	RanInt_09	989	0,14	2,14	0	961
	RanInt_10	923	0,26	1,72	0	772
10	**Geo_01**	3660,67	0,12	0,16	0	504
	Geo_02	1404,8	0,15	0,14	0	554
	Geo_03	1662,12	0,11	0,11	0	403
	Geo_04	3504,88	0,07	0,13	0	541
	Geo_05	1994,6	0,12	0,14	0	577
	Geo_06	1161,77	0,1	0,1	0	342
	Geo_07	3031,27	0,1	0,12	0	509
	Geo_08	2243,15	0,08	0,14	0	534
	Geo_09	3828,65	0,09	0,12	0	476
	Geo_10	3752,03	0,08	0,12	0	490
12	**Geo_01**	716,463	0,39	3,18	71	4204
	Geo_02	840,198	0,17	4,21	63	4559
	Geo_03	2368,84	0,19	13,73	0	14633
	Geo_04	1657,54	0,26	14,84	0	16120
	Geo_05	2120,14	0,43	14,2	22	14532
	Geo_06	1482,28	0,48	8,88	10	11052
	Geo_07	1445,27	0,45	14,76	0	15495
	Geo_08	2261,4	0,41	13,33	0	14652
	Geo_09	1734,33	0,5	10,11	19	14511
	Geo_10	783,734	0,54	3,33	0	4406

https://proceedings.science/p/85427?lang=pt-br

Tabela 4. Comparação entre os algoritmos para instâncias onde os grupos possuem tamanhos diferentes e $n > 12$

n	Instância	Valor atingido		Tempo(s)		GAP(%)		Nós	
		LIN	QUAD	LIN	QUAD	LIN	QUAD	LIN	QUAD
30	**RanReal_01**	5572,07	5610,92	3427,15	2754,55	0,401187	1,561726	278458	708052
	RanReal_02	5770,78	5726,85	3437,65	2739,55	0,313523	1,632044	273619	736255
	RanReal_03	5400,11	5438,2	3430,375	2693,175	0,2518	1,514818	278889	762342
	RanReal_04	5836,04	5887,12	3442,275	2755,325	0,435701	1,600357	254371	753584
	RanReal_05	5605,7	5697,27	3415,15	2758,275	0,294399	1,388156	280157	761404
	RanReal_06	5528,61	5526,67	3455,35	2543,75	0,331399	1,469318	255472	720621
	RanReal_07	5464,87	5420,11	3437,5	2606,775	0,453604	1,691517	269167	660768
	RanReal_08	5449,63	5425,34	3438,55	2637,05	0,627209	1,295388	245638	849503
	RanReal_09	5275,75	5369,27	3403,9	2803,725	0,431934	1,628226	321210	722312
	RanReal_10	5503,69	5288,2	3444,725	2688,5	0,226862	1,389842	265405	778164
60	**RanReal_01**	16893,8	16778,5	3403,425	3306,8	0,676009	3,74434	7975	36453
	RanReal_02	17697,1	17277,7	3444,525	3406,525	0,510353	3,672919	8866	26102
	RanReal_03	17353,1	17122,3	3410,725	3407,575	0,627381	3,688596	7591	30528
	RanReal_04	18184,4	17595,9	3419,925	3459,725	0,521435	3,580332	8093	19501
	RanReal_05	17306,2	17136,1	3387,2	3452,6	0,676641	3,802309	5537	25727
	RanReal_06	17487,4	16364	3397,3	3431,275	0,589636	3,884106	7662	24498
	RanReal_07	17027,2	16693,4	3428,6	3399,825	0,686762	3,828435	7792	39999
	RanReal_08	17449,5	16668,8	3433	3298,475	0,442618	3,826434	9921	35211
	RanReal_09	17823,7	16723,9	3355,025	3431,125	0,616263	3,894392	4871	27005
	RanReal_10	17412,1	17565,6	3377,825	3387,625	0,6476	3,635606	6923	35203
30	**RanInt_01**	5568	5444	3431,525	2652,325	0,396245	1,683134	295903	748856
	RanInt_02	5821	5625	3427,225	2744,85	0,336756	1,664921	294089	791146
	RanInt_03	5458	5426	3446,65	2775,25	0,248961	1,512422	258183	793683
	RanInt_04	5607	5496	3453,075	2718,475	0,408032	1,518285	254302	672660
	RanInt_05	5741	5781	3440,425	2520,8	0,278144	1,399128	292590	790649
	RanInt_06	5309	5050	3456,475	2118,455	0,322642	1,64327	264059	805426
	RanInt_07	5364	5261	3437,025	2750,525	0,472484	1,667506	273698	749064
	RanInt_08	5201	5324	3419,775	2737,225	0,681152	1,208304	277303	884080
	RanInt_09	5734	5615	3412	2686,325	0,368748	1,650429	298629	740332
	RanInt_10	5689	5670	3419,1	2782,075	0,256412	1,416731	270094	790311
60	**RanInt_01**	17320	16506	3396,925	3428,975	0,663068	3,888346	6164	28074
	RanInt_02	17402	16859	3425,875	3446,825	0,519376	3,645453	7990	28130
	RanInt_03	17041	16613	3424,875	3455,4	0,655182	3,717504	7818	28429
	RanInt_04	17423	16834	3440,475	3441,75	0,56604	3,600648	8548	28712
	RanInt_05	17459	17220	3410,7	3455,45	0,653147	3,712768	8675	26324
	RanInt_06	17545	17289	3404,525	3395,4	0,620732	3,658811	5845	34077
	RanInt_07	17430	16791	3380,3	3397,25	0,652521	3,732033	6884	37906
	RanInt_08	17971	16546	3434,6	3447,725	0,42649	3,883408	9294	27926
	RanInt_09	17365	16991	3423,775	3421	0,668874	3,807087	7439	35002
	RanInt_10	17211	16589	3423,625	3450,6	0,635858	3,780145	5760	27731
30	**Geo_01**	14333,5	14293,6	3405,2	3625,59	0,428673	0,237401	311237	596378
	Geo_02	13821,8	13730,9	3401,8	3293,96	0,367844	0,214202	287162	598014
	Geo_03	11302,5	11098,5	3405,475	3256,98	0,244391	0,540964	286946	608249
	Geo_04	6078,55	5864,42	3442,7	3302,66	0,495114	1,04613	251940	568930
	Geo_05	15125,1	14956,7	3414,525	3102,52	0,232202	0,344185	271055	510757
	Geo_06	14844,7	14809,5	3406,175	3089,37	0,349589	0,428564	273528	519966
	Geo_07	7932,88	7864,79	3426,075	3064,97	0,516234	0,681997	257500	489928
	Geo_08	13282	13339	3391,975	3661,82	0,720243	0,122277	238627	657515
	Geo_09	11142,1	11040,5	3403,225	3140,16	0,411765	0,361417	281159	525223
	Geo_10	10822,6	10792,9	3384,85	3126,93	0,29503	0,375937	328285	515931
60	**Geo_01**	46760,4	47354,7	11838,675	2932,78	0,360579	0,24074	11630	103574
	Geo_02	45674,6	45319,4	11329,85	2960,18	0,241171	0,223006	10300	102951
	Geo_03	54774,3	53588,4	13397,1	2981,06	0,299867	0,391432	10871	103012
	Geo_04	48593,9	47965,3	11991,325	2886,82	0,276145	0,252746	10220	101168
	Geo_05	54510,7	53409,2	13352,3	2898,18	0,284586	0,349373	11987	100881
	Geo_06	26010,8	26387,5	6596,875	2727,69	0,446521	0,334473	8500	100318
	Geo_07	39282,4	39528,2	9882,05	2817,27	0,394176	0,232483	10037	103307
	Geo_08	42551,1	42446,5	10611,625	2872,13	0,175865	0,281753	13123	101322
	Geo_09	38103,6	37841,2	9460,3	2825,21	0,386043	0,655149	10701	105061
	Geo_10	54056,1	54183,6	13545,9	3261,66	0,322221	0,303709	12352	116127

https://proceedings.science/p/85427?lang=pt-br

Tabela 5. Comparação entre os algoritmos para instâncias onde os grupos possuem tamanhos iguais e $n > 12$

n	Instância	Valor atingido		Tempo(s)		GAP(%)		Nós	
		LIN	QUAD	LIN	QUAD	LIN	QUAD	LIN	QUAD
30	**RanReal_01**	5503,12	5361,9	3549,9	2709,175	0,015791	1,107376	231245	947363
	RanReal_02	5604,27	5249,14	3535,725	2527,375	0,014382	1,247739	243338	895135
	RanReal_03	5265,5	5093,13	3432,875	2764,7	0,047754	1,151039	308680	936468
	RanReal_04	5559,47	5506,37	3449,025	2722,15	0,054291	1,154807	291540	915564
	RanReal_05	5470,54	5314,89	3441,125	2164,7375	0,046068	1,187771	288260	790034
	RanReal_06	5476,08	5449,21	3531,8	2673,075	0,018328	1,025793	247679	996192
	RanReal_07	5349,54	5321,08	3534,175	2538,8	0,021566	1,123759	239413	912146
	RanReal_08	5464,45	5136,32	3476,775	2142,125	0,037051	1,299048	287285	795271
	RanReal_09	5286,4	5106,97	3439	2565	0,05876	1,164013	285138	926012
	RanReal_10	5276,67	5112,14	3531,325	2089,235	0,027154	1,100992	236359	763921
60	**RanReal_01**	17666,9	15535,8	3458,2	3422,2	0,272695	3,618206	13008	37704
	RanReal_02	17652,1	15715,5	3467,825	3354,3	0,288059	3,787539	12543	48466
	RanReal_03	17710,1	15941,8	3425,6	3415,4	0,274506	3,469625	12071	43213
	RanReal_04	18050,8	15851,2	3452,625	3434,8	0,268737	3,519079	12946	42610
	RanReal_05	17823,9	16695,8	3429,325	3424,05	0,282909	3,370123	12346	42223
	RanReal_06	17063,9	16510,5	3444,575	3429,475	0,309573	3,224064	13712	45743
	RanReal_07	17826,1	16306,4	3435,7	3439	0,272013	3,39646	12123	38124
	RanReal_08	17681,5	15604,3	3442,65	3433,225	0,28114	3,572377	13313	37566
	RanReal_09	17709,4	15906,1	3435,325	3322,375	0,272432	3,634776	13701	54233
	RanReal_10	17589,4	15635,1	3426,675	3315,75	0,305209	3,725054	12043	48722
30	**RanInt_01**	5363	5039	3423,275	2135,085	0,050283	1,290497	313831	825062
	RanInt_02	5652	5389	3453,15	2378,345	0,.051561	1,14623	285388	936372
	RanInt_03	5325	5260	3443,85	2658,275	0,052988	1,115759	305740	930632
	RanInt_04	5303	5136	3470,225	2760,875	0,048868	0,987531	283577	1128329
	RanInt_05	5496	5422	3459,5	2640,25	0,.046303	1,149639	269293	983233
	RanInt_06	5194	5097	3541,625	2746,975	0,020803	1,081295	221437	985374
	RanInt_07	5294	5276	3496,05	2745,075	0,035252	1,078711	261294	972640
	RanInt_08	5337	5182	3468,875	2621,175	0,044519	1,120975	295807	904659
	RanInt_09	5568	5316	3468,925	2314,3225	0,045941	1,176994	273963	984526
	RanInt_10	5499	5445	3463,8	2712,025	0,044729	1,09835	285292	956115
60	**RanInt_01**	17699	16544	3414,575	3377,875	0,292318	3,490964	12938	50075
	RanInt_02	17388	16181	3392,9	3425,65	0,291275	3,327145	11313	48188
	RanInt_03	17512	15899	3436,35	3423,875	0,284327	3,371323	12580	39731
	RanInt_04	17302	16528	3468,925	3408,375	0,298883	3,217912	13228	40187
	RanInt_05	17460	16926	3450,85	3437,1	0,299841	3,222734	13568	40456
	RanInt_06	17457	16905	3442,225	3441,525	0,307832	3,151142	12350	40735
	RanInt_07	17731	16852	3464,4	3437,125	0,285726	3,266286	12034	40521
	RanInt_08	17816	16711	3425,375	3450,475	0,286893	3,301806	12726	40069
	RanInt_09	17742	16982	3444,45	3429,05	0,294098	3,198642	12679	46685
	RanInt_10	17369	15787	3448,7	3404,15	0,292538	3,521172	12167	40859
30	**Geo_01**	13777	13751,6	3437,9	3089,87	0,022081	0,126086	316691	529961
	Geo_02	13269,5	13213	3303,25	3157,27	0,031427	0,124244	404099	536587
	Geo_03	10831	10780,4	2695,1	3410,42	0,040814	0,254234	454325	628913
	Geo_04	5438,85	5431,19	1357,7975	3206,84	0,105664	0,557579	393700	560497
	Geo_05	13918,7	13847,8	3461,95	3143,11	0,022997	0,158526	399080	557184
	Geo_06	14333,7	14342,7	3585,675	3295,92	0,027772	0,216798	389259	591705
	Geo_07	7588,65	7573,68	1893,42	3399,43	0,059765	0,323781	392354	570227
	Geo_08	13348,1	13325,8	3331,45	3246,82	0,034066	0,125029	415990	567866
	Geo_09	10690,1	10654,7	2663,675	3305,82	0,038503	0,207896	279478	610447
	Geo_10	10373,9	10377,2	2594,3	3130,53	0,048663	0,21448	393702	561834
60	**Geo_01**	45245,9	45198,8	11299,7	2830,66	0,095619	0,10944	18954	104597
	Geo_02	43034,5	43046,6	10761,65	2891,88	0,107715	0,100943	17415	107423
	Geo_03	51189,4	51286,3	12821,575	2945,84	0,088021	0,135914	20538	106851
	Geo_04	45971,8	45896,4	11474,1	2884,99	0,095598	0,094198	17993	106335
	Geo_05	49966,5	50040,1	12510,025	2905,54	0,086483	0,124825	19782	104165
	Geo_06	25012,5	25095,9	6273,975	3052,68	0,1883	0,138391	18550	108972
	Geo_07	37860,6	37937,7	9484,425	2809,24	0,116811	0,103038	15901	100358
	Geo_08	40211,8	40152,9	10038,225	2801,3	0,106159	0,178635	18154	107562
	Geo_09	36094,1	36118,4	9029,6	2766,83	0,128645	0,268611	16883	104925
	Geo_10	51785,5	51621,4	12905,35	2993,73	0,082564	0,121827	20037	106521

https://proceedings.science/p/85427?lang=pt-br

Já as tabelas 4 e 5 apresentam os resultados dos experimentos para as instâncias onde algoritmos não conseguiram obter a solução ótima dentro do tempo e utilização de memória pré-definidos. De forma semelhante as tabelas anteriores, a tabela 4 apresenta os resultados para as instâncias onde os grupos possuem tamanhos diferentes (60 instâncias) e a tabela 5 os resultados dos experimentos para instâncias onde os grupos possuem tamanhos iguais (60 instâncias). Nestas tabelas, estão presentes a coluna de identificação da instância solucionada (*RanReal*, *RanInt* ou *Geo*), assim como o valor da solução encontrada até o momento do término da execução. Para cada algoritmo (*Lin*, *Quad*) é apresentado: o tempo de execução (em segundos), o GAP que é informado pelo *solver*, e o número de nós gerados na árvore de solução.

Os resultados apresentados nas tabelas 2 e 3 mostram que em 84, de 120 instâncias, o tempo de execução do algoritmo *Lin* foi inferior ou igual ao tempo de execução do algoritmo *Quad*. Quanto ao consumo de memória, o número de nós criados pelo algoritmo *Quad*, exceto para 40 instâncias simples onde o número de nós criados é 0 em ambos algoritmos, em 73, das 80 instâncias restantes, é maior que a quantidade de nós gerado pelo algoritmo *Lin*.

Analisando os resultados apresentados na Tabela 4 e 5, nota-se que o GAP gerado pelo algoritmo *Quad* é até 120 vezes mais alto que o GAP gerado pelo algoritmo *Lin*, para as instâncias do tipo *RanReal* e *RanInt*. Já para as instâncias do tipo *Geo* não há uma grande diferença entre o GAP obtido pelos dois algoritmos, sendo ambos valores considerados altos, indicando que ambos algoritmos encontram dificuldades para solucionar esta classe de instâncias nas condições de gasto tempo e utilização memória pré-definidos. Novamente, em relação ao consumo de memória, quando maior o valor de n maior a discrepância entre a criação de nós do algoritmo *Quad* em comparação ao algoritmo *Lin* na maioria das instâncias.

5. Conclusões e trabalhos futuros

Apresentamos neste trabalho um modelo de programação linear inteira para o PDMG jutamente com uma análise do seu desempenho em comparação com modelo quadrático proposto por [Fan et al. 2011]. A análise foi realizada considerando o tempo de execução e a utilização de memória gasto por cada modelo para obtenção de soluções ótimas. Para tal, foi utilizada uma amostra de 240 instâncias pertencentes ao MDGPLIB. Embora os experimentos considerem um tempo de execução máximo e uma utilização de memória limitados, 3600 segundos e 26000 MB, respectivamente, para cada execução, é possível afirmar que apesar da extensão desse limite promover um possível aumento no número de instâncias solucionadas, a utilização de memória e tempo de execução para solucionar essas novas instâncias também cresceriam de forma proporcional em ambos algoritmos. Sendo assim, podemos concluir que o modelo linear inteiro proposto neste trabalho pode ser considerado superior ao modelo quadrático existente na literatura, principalmente no requisito de utilização de memória.

Quanto ao uso de algoritmos exatos para resolução do PDMG, embora o modelo de programação linear inteira tenha sido superior em comparação ao modelo quadrático proposto por [Fan et al. 2011], para instâncias com valores de n maiores que 30, o modelo proposto ainda encontra dificuldades para encontrar a solução ótima em um curto intervalo de tempo. Espera-se em trabalhos futuros explorar de melhor forma o mo-

https://proceedings.science/p/85427?lang=pt-br

delo programação linear inteiro proposto, efetuando buscas de desigualdades válidas que possam fortalecê-lo, assim como estudos para criação de algoritmos de planos de corte utilizando tais desigualdades. Outra possível proposta seria a criação de modelo híbridos, uma vez que a literatura expõe um extenso número de trabalhos focados em abordagens heurísticas para o PDMG.

Referências

[Glover et al. 1998] **Glover, Fred; Ching-Chung Kuo; Krishna S. Dhir** *Heuristic algorithms for the maximum diversity problem*. Journal of Information and Optimization Sciences, 19 penalty0 109–132, 1998.

[Rodriguez et al. 2013] **Rodriguez, F. J., Lozano, M., García–Martínez, C., González–Barrera, J. D.** *An artificial bee colony algorithm for the maximally diverse grouping problem*. Information Sciences, 230 183–196, 2013.

[Urošević 2014] **Urošević, D** *Variable neighborhood search for maximum diverse grouping problem*. Yugosl. J. Oper. Res., 20 penalty0 21–33, 2014.

[Palubeckis et al. 2011] **Palubeckis, G.; Karciauskas, A.R.E.** *Comparative performance of three metaheuristic approaches for the maximally diverse grouping problem, Information Technology and Control*. Journal of the Operational Research Society, 40 penalty0 277–285, 2011.

[Gallego et al. 2013] **Gallego, M.; Laguna, M.; Martí, R.; Duarte, A.** *Tabu search with strategic oscillation for the maximally diverse grouping problem*. Journal of Operational Research Society, 64 724–734, 2013.

[Fan et al. 2011] **Fan, Z. P., Chen, Y., Ma, J. and Zeng, S.** *A hybrid genetic algorithmic approach to the maximally diverse grouping problem*. Journal of the Operational Research Society , 62 penalty0 92–99, 2011.

[Feo T. and Khellaf 1990] **Feo, T. and Khellaf, M.** *A class of bounded approximation algorithms for graph partitioning*. Networks, 20:181–195, 1990.

[Feo et al. 1992] **Feo, T., Goldschmidt, O. and Khellaf, M.** *One-half approximation algorithms for the k-partition problem*. Operations Research, 40(1):S170–S1735, 1992.

[O'Brien and Mingers 1995] **Mingers, J. and O'Brien, F. A.** *Creating students groups with similar characteristics: a heuristic approach*. Omega, 23(1):313–321, 1995.

[O'Brien and Mingers 1995] **O'Brien, F. A. and Mingers, J.** *The equitable partitioning problem: a heuristic algorithm applied to the allocation of university student accommodation*. Warwick Business School, 187, 1995.

[Kral, J. 1965] **Kral, J.** *To the problem of segmentation of a program*. Information Processing Machines, 2(1):116–127, 1965.

[Chen 1986] **Chen, C. C.** *Placement and partitioning methods for integrated circuit layout*. Ph.D. Dissertation, EECS Department, University of California, Berkeley, 1986.

[Hettich and Pazzani 2006] **Hettich S. and Pazzani, M. J.** *Mining for element reviewers: Lessons learned at the national science foundation*. Proceedings of the KDD'06, ACM: New York, NY, 862-871.

https://proceedings.science/p/85427?lang=pt-br

[Bhadury et al. 2000] **Bhadury, J., Mighty E. J. and Damar, H.** *Maximizing workforce diversity in project teams: A network flow approach*. Omega, 28 143-153, 2000.

[Kuo et al. 1993] **Kuo, C.-C.; Glover, F.; Dhir, K.** *Analyzing and modeling the maximum diversity problem by zero-one programming*. Decision Science, 24 1171-1185, 1993.

https://proceedings.science/p/85427?lang=pt-br