# UM ESTUDO COMPUTACIONAL DO PROBLEMA DE AGRUPAMENTO COM SOMA MÍNIMA DE DISTÂNCIAS

Augusto Pizano Vieira Beltrão – UFF E-mail augusto_pvb@hotmail.com
José André de Moura Brito – ENCE/IBGE E-mail jambrito@gmail.com

**Resumo**

Este artigo traz a proposta de um algoritmo que foi aplicado ao problema de agrupamento com soma mínima de distâncias (PASMD). Dada uma base de dados com n objetos e q variáveis, busca-se distribuir os objetos em k grupos, de modo que a soma total de distâncias entre todos os pares de objetos, dentro de cada um dos grupos, seja mínima. Este problema tem uma alta complexidade computacional, o que dificulta a aplicação de métodos de enumeração exaustiva ou implícita. Considerando esta questão, foi desenvolvido um algoritmo heurístico baseado nos algoritmos genéticos de chaves aleatórias viciadas (biased random-key genetic algorithm – BRKGA). De forma a avaliar este algoritmo, foram realizados experimentos computacionais, considerando a sua aplicação em um conjunto de 49 bases dados. Os resultados obtidos indicam que esse algoritmo se constitui como uma boa alternativa à resolução do PASMD.

**Palavras-Chaves:** Análise de Agrupamentos, Soma Mínima e BRKGA.

## 1. Introdução

Com a disponibilidade de máquinas com processadores cada vez mais velozes, e com grande quantidade de memória, tem ocorrido, nos últimos tempos, um grande aumento quanto à capacidade de processamento e de armazenamento de dados. Considerando, então, tantos dados disponíveis e, das mais variadas fontes e finalidades, surge a questão natural de como extrair informações úteis desses dados.

Uma forma de extrair informação relevante, a partir da análise dos dados, consiste na utilização da análise de agrupamentos, que tem um importante papel em diversas áreas do conhecimento. Há vários exemplos de aplicações da análise de agrupamentos, sejam eles: na Medicina (incidência de certos tipos de câncer), na Química (classificação de compostos), no Marketing (segmentação de clientes) etc.

A análise de agrupamentos é uma técnica de Análise Multivariada que agrega vários métodos que têm, como objetivo, a formação de grupos. Mais especificamente, busca-se dividir os n objetos de a uma base de dados em grupos, de maneira que cada grupo seja composto por objetos similares entre si (homogêneos). Tanto a similaridade quanto a dissimilaridade são

calculadas utilizando uma medida de distância escolhida (Euclidiana, Manhattan etc) que é função das q variáveis associadas aos n objetos da base de dados.

A importância do agrupamento de objetos em diversas ciências e o enorme acúmulo de observações em bancos de dados tem motivado, nas últimas décadas, o desenvolvimento de vários novos métodos de agrupamento. Esses métodos estão diretamente associados à forma de definir os agrupamentos, à métrica (distância) e à função objetivo utilizada. Ou seja, de acordo com a função, define-se um problema de agrupamento associado.

Este artigo traz a proposta de um novo algoritmo heurístico para o PASMD, que corresponde a um problema de alta complexidade computacional (BRITO; BRITO, 2008). O algoritmo foi desenvolvido mediante o estudo da metaheurística Algoritmo Genético de Chaves Aleatórias Viciadas. Foi aplicado em um conjunto de 49 bases de dados, de diferentes tamanhos (número de objetos) e dimensões (quantidade de atributos).

O artigo está dividido da seguinte forma: Na seção dois são apresentados os conceitos básicos de análise de agrupamentos. A seção três traz uma descrição do problema que foi abordado neste artigo. A seção quatro traz uma descrição do BRKGA e uma descrição do algoritmo proposto para o PASMD. Na seção cinco são apresentados alguns resultados computacionais referentes à aplicação do algoritmo nas bases de dados, análises e conclusões.

## 2. Análise de Agrupamentos

Atualmente, existe um grande interesse, por parte de diversos pesquisadores, em analisar bases de dados constituídas por n objetos e q atributos, com o objetivo de aplicar os métodos análise de agrupamentos para identificar grupos, mais especificamente, separar os objetos em grupos homogêneos. Desta forma, há a necessidade de se utilizar e desenvolver métodos que traduzam bases de dados em informação sobre a estrutura "natural" dos dados (HAIR et al., 2005). O advento da computação, e a crescente massa de dados disponibilizada nos últimos anos representam incentivos ao aprimoramento e desenvolvimento de métodos que sejam capazes de identificar grupos a partir das bases de dados para atingir este objetivo. Essas bases podem ser encontradas, por exemplo, em sites de órgãos públicos, universidades etc.

Os métodos de agrupamento operam através da utilização de medidas de distância que permitem a avaliação da dissimilaridade ou similaridade, sendo essas distâncias função das q variáveis associadas aos n objetos. Desta forma, a distância entre dois objetos quaisquer de uma base de dados é calculada a partir dos valores das variáveis dos dois objetos, através de

uma fórmula matemática. Quanto menor a distância entre dois objetos, maior é a similaridade entre eles ou, equivalentemente, menor é a dissimilaridade.

Uma base de dados com n objetos, pode ser representada pelo conjunto $X = \{x_1, x_2, ..., x_n\}$, onde cada elemento $x_i$ (objeto da base) corresponde a um vetor com q atributos (variáveis), ou seja, $x_i = (x_{i1}, x_{i2}, ..., x_{iq})$. Sendo assim, a base de dados X é representada em uma matriz $A_{nxq}$, onde as linhas correspondem aos objetos a serem agrupados e as colunas correspondem aos atributos de cada objeto. Ainda neste sentido, os algoritmos de agrupamentos têm, como entrada, uma matriz de distâncias $D_{nxn}$, sendo o valor de cada entrada $d_{ij}$ da matriz correspondente à distância entre dois objetos $x_i$ e $x_j$. Para fins da avaliação da dissimilaridade, neste trabalho foi utilizada a distância euclidiana.

## 2.1 Métodos de Agrupamento

Testar todas as alocações possíveis dos n objetos em k grupos, a fim de obter a melhor solução possível (grupos mais homogêneos) é uma tarefa que, em geral, não é possível, mesmo com os melhores computadores disponíveis atualmente. Por conta dessa dificuldade, e das inúmeras aplicações da análise de agrupamentos, muitos algoritmos de agrupamento têm sido desenvolvidos. Esses algoritmos, em geral, produzem soluções "razoáveis" (ótimos locais[1]), sem a necessidade de explorar todas as soluções possíveis (JOHNSON et al., 2007), ou seja, aplicar uma enumeração exaustiva, que tem por objetivo produzir a solução ótima global[2], pode não ser viável dependendo do número de objetos. Pois, conforme aumenta-se n, o número de soluções a serem enumeradas para o problema de agrupamento aumenta substancialmente (FADEL, 2013). Mais especificamente, o número de soluções possíveis (a enumerar) para n objetos e k grupos é dada pela fórmula do número de Stirling de segundo tipo (JOHNSON et al., 2007, p.672). A Tabela 1 ilustra, a partir desta fórmula, o aumento de soluções de acordo com o número de objetos em uma determinada base de dados.

Tabela 1- Quantidade de soluções possíveis para o problema de agrupamento de com n e k

| n \ k | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| 20 | $5{,}24*10^{11}$ | $5{,}81*10^{14}$ | $4{,}52*10^{16}$ | $7{,}49*10^{17}$ |
| 30 | $5{,}37*10^{14}$ | $3{,}43*10^{19}$ | $4{,}80*10^{22}$ | $7{,}71*10^{24}$ |
| 40 | $5{,}50*10^{17}$ | $2{,}03*10^{24}$ | $5{,}04*10^{28}$ | $7{,}57*10^{31}$ |
| 50 | $5{,}63*10^{20}$ | $1{,}20*10^{29}$ | $5{,}28*10^{34}$ | $7{,}40*10^{38}$ |
| 60 | $5{,}76*10^{23}$ | $7{,}07*10^{33}$ | $5{,}54*10^{40}$ | $7{,}23*10^{45}$ |
| 70 | $5{,}90*10^{26}$ | $4{,}17*10^{38}$ | $5{,}81*10^{46}$ | $7{,}06*10^{52}$ |

Fonte: Elaboração Própria

[1] Ótimo local: o ótimo local é a melhor solução possível, para um conjunto de soluções vizinhas.
[2]Ótimo global: o ótimo global é a melhor solução, dentre todas as soluções possíveis para determinado problema de otimização.

Uma alternativa à enumeração consiste na aplicação de métodos de agrupamento classificados em hierárquicos e não hierárquicos (HAIR et al., 2005, p.398).

### 2.1.1 Métodos Hierárquicos

São caracterizados, em sua execução, pela formação de estruturas hierárquicas formadas por sucessivas fusões ou divisões de grupos. Os métodos hierárquicos são divididos em dois tipos, a saber: Hierárquicos aglomerativos - começam com número de grupos igual ao de objetos e, a cada passo, os objetos mais próximos (de menor distância) são agrupados. Então, grupos mais similares também são agrupados até que finalmente todos os objetos formem um único grupo. Hierárquicos divisivos - começam com todos os objetos em apenas um grupo, e, no primeiro passo, divide-se esse grupo em dois, segundo a dissimilaridade de seus objetos, e os grupos permanecem sendo divididos de acordo com as distâncias entre eles, calculadas passo a passo, até que cada objeto seja um grupo.

### 2.1.2 Métodos Não Hierárquicos

Estes métodos têm como peculiaridade a definição, a priori, do número de grupos. Uma vantagem dos métodos não hierárquicos é que eles podem ser aplicados a bases de dados maiores (mais objetos), pois não há a necessidade de se armazenar a priori, em uma estrutura de dados, a informação de várias soluções para diferentes quantidades de agrupamentos como nos métodos hierárquicos. Dentre os métodos não hierárquicos destacamos o k-means (k-médias) (HAIR et al., 2005) e k-medoides (KAUFMAN; ROUSEEUW, 2009).

## 3. Problema de Agrupamento com Soma Mínima de Distâncias

O tipo de distância considerada e a função objetivo escolhida (avalia a homogeneidade dos grupos), determinam diferentes estruturas de agrupamento. Seja o conjunto X constituído por n objetos $X = \{x_1, x_2, ..., x_i, ..., x_n\}$, com q variáveis, tal que $x_i = (x_{i1}, x_{i2}, ..., x_{iq})$ e que a distância $d_{ij}$ entre dois objetos $x_i$ e $x_j$ quaisquer seja definida, por exemplo, como a distância euclidiana e suponha que o número de grupos k seja fixado a priori. No problema de agrupamento com soma mínima de distâncias (PASMD), busca-se alocar os n objetos em k grupos, denotados por $C_1, C_2, ..., C_k$, de forma que a soma total das distâncias $d_{ij}$ entre todos os objetos, tomados dois a dois, dentro de cada um dos grupos, seja mínima. Ou seja, busca-se minimizar a seguinte função objetivo:

$$\min f = \sum_{g=1}^{k} \sum_{\forall x_i, x_j \in C_g} d_{ij}$$

Ressalta-se, que resolver o PASMD não é uma tarefa trivial, e que realizar um processo de busca exaustiva, levando em conta todas as possíveis soluções para o problema pode não ser viável, visto que a quantidade de soluções possíveis para o problema é limitada, também, pelo número de *Stirling* de segundo tipo. Ainda neste sentido, dentre as propostas mais recentes encontradas na literatura, para a resolução desse problema, temos o trabalho de Brito e Brito (2008) com duas metaheurísticas, um algoritmo genético e o VNS (Variable Neighborhood Search). Serpa D. R. (2009) agrupa dados biológicos utilizando a função objetivo do problema de soma mínima e utilizando as metaheurísticas VNS e GRASP. Nascimento et al. (2009) resolve o problema proposto através de um algoritmo baseado na metaheurística GRASP.

Os algoritmos de agrupamento produzem, em geral, soluções que correspondem a ótimos locais. Por sua vez, esses ótimos podem ser considerados de boa qualidade (satisfatórios) quando têm proximidade com o ótimo global. Um algoritmo de agrupamento pode produzir ótimos locais distantes da solução ótima. Fato esse que pode ser problemático, pois as estruturas formadas na solução final podem não corresponder a estrutura real dos dados. Considerando essa questão, as metaheurísticas têm sido aplicadas a diversos problemas de agrupamento. Metaheurísticas são heurísticas de uso geral desenvolvidas para uso em diversos problemas de otimização.

## 4. Algoritmo Genético de Chaves Aleatórias Viciadas

Assim como no algoritmo genético clássico, os termos cromossomo, população, geração e os operadores genéticos também estão presentes no BRKGA (GONÇALVES; RESENDE, 2011). Cada cromossomo corresponde a um vetor u de n posições com valores reais gerados segundo uma distribuição uniforme [0,1]. Assim sendo, a população, em cada geração, é dada por um conjunto de p vetores u. Os operadores de seleção, cruzamento e mutação são aplicados nestes vetores. Também há um procedimento específico no BRKGA denotado por decodificador. O decodificador é aplicado em cada um dos p vetores u, produzindo p vetores solução s, que correspondem às soluções viáveis para o problema de otimização em questão, sendo o decodificador particular de cada problema. Obtidos os vetores s, calcula-se o valor da função objetivo do problema em questão, para cada um dos vetores.

1) Seleção – Calcula-se a função objetivo para cada um dos p vetores s e, com estes valores, ordena-se os cromossomos u do melhor para o pior considerando os valores da função objetivo. Neste caso, os melhores cromossomos são aqueles que apresentam o menor valor da função objetivo. Os $P_e$*100% melhores cromossomos são classificados como elite, compondo

o conjunto elite (E) que é copiado para a próxima geração e o resto dos cromossomos são classificados como não-elite. O objetivo deste operador é garantir que os vetores u que produzem as melhores soluções s, encontradas até o momento, não sejam perdidas. O tamanho de E é dado por:

$$TE = p*P_e$$

2) Mutação – Gera-se um número de novos vetores u que formam o conjunto mutação M. O objetivo deste operador é evitar que as soluções fiquem presas a mínimos locais não satisfatórios e dar diversidade ao algoritmo, pois adiciona à população soluções completamente novas, geradas estocasticamente. O tamanho do conjunto mutante é dado por:

$$TM = p*P_m$$

3) Cruzamento – Este operador cria novos cromossomos (vetores u) para a nova população a partir de cromossomos já existentes na população atual, combinando vetores de chaves aleatórias da população atual repetidas vezes para formar um conjunto da nova população. Um dos pais é um cromossomo escolhido aleatoriamente do conjunto elite, enquanto o outro é um outro cromossomo escolhido aleatoriamente do conjunto não-elite. Um mesmo pai pode ter mais de um filho por geração, pois os cromossomos são escolhidos com reposição. Então, os dois cromossomos (pais) são combinados da seguinte maneira:

i) É criado um cromossomo filho igual ao cromossomo não-elite.

ii) É gerado um vetor de dimensão 1 x n, suas entradas são números gerados estocasticamente no intervalo [0,1], sendo esse vetor chamado de vetor RNG.

iii) Para cada entrada i do vetor RNG verifica-se se este valor é menor que $Prob_e$ que corresponde à probabilidade de cruzamento**.** Caso seja, então a i-ésima entrada do cromossomo filho é substituída pela i-ésima entrada (característica) do cromossomo elite. Deste modo, quanto maior o valor de $Prob_e$, maiores as chances de um filho possuir muitas características do cromossomo (pai) elite.

Figura 1 - Exemplo de um cruzamento entre dois vetores pais para gerar um vetor filho (o tamanho do vetor de chaves aleatórias é 6 (n=6) e $Prob_e$ = 0,8.)

| | | | | | | |
|---|---|---|---|---|---|---|
| RNG | 0,28 | 0,32 | 0,83 | 0,24 | 0,97 | 0,28 |
| Pai Conjunto Elite | **0,73** | **0,38** | 0,96 | **0,09** | 0,15 | **0,95** |
| Pai Conjunto Não-Elite | 0,98 | 0,39 | **0,08** | 0,04 | **0,70** | 0,52 |
| Filho | **0,73** | **0,38** | 0,08 | **0,09** | 0,70 | **0,95** |

Fonte: Elaboração Própria

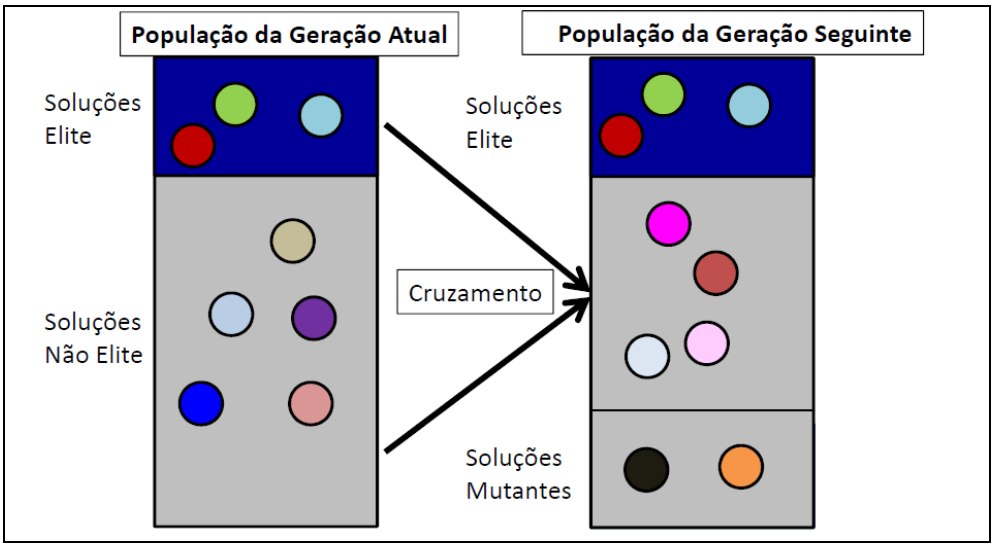
iv) Depois destes três passos, o cromossomo filho é colocado no conjunto cruzamento que fará parte da próxima geração. Desta forma, tanto os cromossomos elite como os não-elite

têm probabilidade de passar suas características para as próximas gerações. Então, outros dois cromossomos (pais) são escolhidos de igual modo (e com reposição) de maneira sucessiva até que se tenha gerado uma quantidade suficiente de cromossomos filho para a geração seguinte. O tamanho do conjunto cruzamento é dado por:

$$POP_{cruza} = p - TE - TM$$

A Figura 2 ilustra a transição da geração atual para a próxima, através da aplicação dos operadores genéticos.

Figura 2 - Transição da geração atual para a nova geração.




Fonte: Figura baseada em figura apresentada em Gonçalves e Resende (2011).

Os três operadores genéticos são aplicados em todas as m gerações, em particular, da primeira para a segunda.

### 4.1 Decodificador Proposto para o PASMD

O decodificador implementado para o problema de soma mínima recebe um vetor u de chaves aleatórias de tamanho n (número de objetos) e retorna um vetor solução s de igual tamanho e o valor numérico calculado a partir da função objetivo abaixo:

$$\min f = \sum_{g=1}^{k} \sum_{\forall x_i, x_j \in C_g} d_{ij}$$

Cada entrada $s_i$ do vetor s corresponde a um inteiro positivo entre 1 e k. Esse vetor fornece a alocação dos objetos aos k grupos. A Figura 3 ilustra um agrupamento com n = 10 e k = 3. A primeira linha representa o número de cada objeto, a segunda linha o vetor u e a terceira linha representa o número do grupo que cada objeto está alocado. Por exemplo, o grupo 1 é

formado pelos objetos (7,9), o grupo 2 pelos objetos (2, 3, 4, 5) e o grupo 3 é formado pelos objetos (1,6,8,10).

Figura 3-Exemplo de agrupamento de 10 objetos em 3 grupos.

| objeto | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| u | 0,89 | 0,51 | 0,45 | 0,64 | 0,62 | 0,75 | 0,13 | 0,77 | 0,06 | 0,75 |
| s | 3 | 2 | 2 | 2 | 2 | 3 | 1 | 3 | 1 | 3 |

Fonte: Elaboração Própria

O vetor solução para PASMD é obtido a partir do vetor de chaves aleatórias em dois passos: (1) Multiplica-se cada entrada do vetor u pelo número k de agrupamentos produzindo um vetor z e (2) Aplica-se uma função em cada entrada de z. Esta função recebe um número real Y (entre 0 e k) e retorna o menor número inteiro maior que Y em cada posição de s. Em seguida, aplica-se a função objetivo em s, calculando as distâncias entre objetos, tomados dois a dois dentro de cada um dos grupos.

## 5. Resultados Computacionais

O algoritmo BRKGA, que incorpora os operadores descritos na seção anterior, além do decodificador específico, foi aplicado em um conjunto de 49 bases de dados, de diferentes fontes, com o número de objetos variando de 16 a 900 e o número de variáveis de 2 a 60. Todas as bases de dados foram padronizadas (*z-score*) a fim de evitar que variáveis de magnitudes diferentes tivessem impacto muito maior no agrupamento.

O BRKGA proposto neste artigo recebe como parâmetros de entrada: o tamanho n, o número k de grupos, percentual $P_e$ da população considerado elite (menor que 50%), percentual $P_m$ de mutantes na população, probabilidade $Prob_e$ de uma característica de um cromossomo elite estar presente em seu filho (maior que 50%) e o número m de gerações. Em consonância com Gonçalves e Resende (2011), o valor escolhido para o tamanho da população foi 100. O algoritmo foi programado em linguagem R e os experimentos computacionais foram realizados em um computador com processador Intel(R) Core(TM) i5-6400 CPU @ 2.70 GHz e dotado de 16 GB de memória. Para definir os valores dos parâmetros $P_e$, $P_m$ e $Prob_e$, foram escolhidas 5 bases de dados de diferentes tamanhos e quantidades de variáveis (dentre as 49). Nessas bases, o algoritmo BRKGA denotado, doravante, BRKGASOMA, foi aplicado considerando diferentes combinações desses parâmetros, para verificar qual combinação produziria os melhores resultados para a função objetivo. Esse experimento foi realizado considerando k = 4.

O BRKGA evolui uma população de vetores de chaves aleatórias ao longo de m gerações, ou até que atinja uma quantidade de gerações sem que o valor da função objetivo da melhor solução encontrada diminua, ou seja, existem dois critérios de parada. O número máximo de gerações foi 1000 e o número máximo de gerações sem melhoria (redução) no valor da função objetivo foi 200.

A combinação de parâmetros escolhida como a melhor foi 20% $P_e$, 20% para $P_m$ e 80% $Prob_e$. Esta combinação foi considerada nos experimentos realizados com o algoritmo em todas as 49 bases. O BRKGASOMA foi aplicado nas 49 bases de dados, considerando k variando de 2 a 5, ou seja, foram feitas 196 execuções do algoritmo. Para cada uma destas execuções, foi registrado: o valor da função objetivo associado à melhor solução encontrada, o tempo de processamento (em segundos), o número de gerações necessárias até encontrar a melhor solução, o total de gerações até o fim do processamento e o vetor s resposta associado à melhor solução encontrada.

A tabela 2 a seguir traz algumas estatísticas descritivas quanto ao número de gerações, além do tempo médio que algoritmo consumiu para cada uma das bases de dados, considerando o número de grupos entre 2 e 5. É possível observar que para um número de grupos acima de 2 foi necessário, em média, um número de gerações superior a 500. Ou seja, melhores soluções são obtidas às custas de um maior número de gerações do algoritmo.

Tabela 2 - Estatísticas descritivas da distribuição dos valores das Gerações e tempo médio.

| **k \ Estatísticas** | **Min** | **Q1** | **Q2** | **Média** | **Q3** | **Máximo** | **CV** | **Tempo Médio (seg.)** |
|---|---|---|---|---|---|---|---|---|
| 2 | 5 | 137 | 278 | 445,89 | 803 | 998 | 80% | 2122,62 |
| 3 | 15 | 328 | 846 | 668,86 | 987 | 1000 | 52% | 1515,22 |
| 4 | 14 | 746 | 968 | 782,30 | 995 | 1000 | 41% | 1241,00 |
| 5 | 27 | 889 | 980 | 837,84 | 998 | 1000 | 33% | 971,87 |

Além desse experimento, de forma a avaliar a qualidade das soluções produzidas pelo BRKGASOMA, em função do tempo de processamento, foi realizado um segundo experimento, utilizando sete bases dados. Este experimento consistiu em aplicar a formulação de programação inteira apresentada nos trabalhos de Brito e Brito (2008) e Nascimento et al (2009) para resolver o PASMD. Essa formulação foi implementada no software LINGO (versão 14.0).

Conforme comentado anteriormente, a aplicação de formulações para problemas de agrupamento só é factível, por conta do tempo computacional, para problemas de pequeno porte (n). Assim sendo, foi estipulado um tempo máximo de processamento para a execução dessa formulação, mais especificamente, 3600 segundos (1 hora). Ao final desse tempo, foi registrado o ótimo local ou global. O Quadro 1 a seguir traz os resultados do BRKGA e da formulação para as sete bases.

Quadro 1 - Resultados do BRKGA e da Formulação

| | | k=2 | | | | | k=3 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Base** | **n** | **BRKGASOMA** | **Tempo** | **Formulação** | **Tempo** | **gap** | **BRKGASOMA** | **Tempo** | **Formulação** | **Tempo** | **gap** |
| IDH_AM | 22 | **10,6** | 3 | 10,6 | 5 | 0,0% | **5,3** | 3 | 5,3 | 132 | 0,0% |
| IDH_ES | 78 | **78,8** | 24 | 89,6 | 3600* | -12,0% | **44,4** | 27 | 68,9 | 3600* | -35,7% |
| IDH_RJ | 92 | **106,7** | 28 | 144,0 | 3600* | -25,9% | **56,4** | 31 | 74,1 | 3600* | -23,9% |
| Iris | 150 | **9.063,5** | 136 | 10.006,5 | 3600* | -9,4% | **4.436,6** | 115 | 8.561,9 | 3600* | -48,2% |
| Wine Data Set | 178 | **28.567,9** | 164 | 34.185,7 | 3600* | -16,4% | **16.507,7** | 156 | 30.391,3 | 3600* | -45,7% |
| maronna | 200 | **12.229,5** | 190 | 17.814,7 | 3600* | -31,4% | **6.690,4** | 354 | 16.208,7 | 3600* | -58,7% |
| face | 296 | **29.181,4** | 1.115 | 59.060,3 | 3600* | -50,6% | **13.709,7** | 767 | 34.021,2 | 3600* | -59,7% |

| | | k=4 | | | | | k=5 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Base** | **n** | **BRKGASOMA** | **Tempo** | **Formulação** | **T** | **gap** | **BRKGASOMA** | **Tempo** | **Formulação** | **Tempo** | **gap** |
| IDH_AM | 22 | 3,6 | 4 | **3,5** | 3600* | 1,2% | 2,5 | 4 | **2,4** | 3600* | 1,2% |
| IDH_RO | 78 | **30,8** | 20 | 47,7 | 3600* | -35,5% | **21,6** | 43 | 31,2 | 3600* | -30,8% |
| IDH_RJ | 92 | **37,5** | 42 | 60,9 | 3600* | -38,3% | **28,0** | 33 | 50,1 | 3600* | -44,2% |
| Iris | 150 | **3.112,4** | 98 | 4.724,5 | 3600* | -34,1% | **2.466,0** | 137 | 4.544,3 | 3600* | -45,7% |
| Wine Data Set | 178 | **11.999,3** | 196 | 17.845,8 | 3600* | -32,8% | **9.396,8** | 165 | 13.192,7 | 3600* | -28,8% |
| maronna | 200 | **3.359,9** | 275 | 10.632,9 | 3600* | -68,4% | **2.625,7** | 233 | 4.840,7 | 3600* | -45,8% |
| face | 296 | **9.877,9** | 594 | 21.028,7 | 3600* | -53,0% | **7.052,5** | 489 | 17.568,8 | 3600* | -59,9% |

3600* Melhor solução viável encontrada no tempo de 1 hora

Os valores da função objetivo em negrito correspondem à melhor solução encontrada e a coluna *gap* indica, em termos relativos, a diferença entre o valor da função objetivo do algoritmo BRKGASOMA e da formulação, sendo

$$gap=100.(FOBJ_{BRKGASOMA}-FOBJ_{formulação})/\ FOBJ_{formulação}$$

É possível observar que, em 23 dos 28 resultados (82%), ou seja, na maioria, o algoritmo BRKGASOMA produziu uma solução de qualidade superior à formulação em um tempo bem menor. A média dos gaps em que o BRKGASOMA apresentou resultados melhores que a formulação é de -38%, mostrando que os resultados do BRKGASOMA são de qualidade consideravelmente superior aos resultados da formulação. Em todos os casos, o BRKGASOMA apresentou os resultados em tempo menor que a formulação, sendo que, a formulação atingiu o limite de 1 hora em 26 dos 28 casos (92%). Nos dois únicos casos em

que a formulação não atingiu o limite de 1 hora, a base de dados continha apenas 22 objetos, mostrando que a formulação se torna uma alternativa pouco viável conforme n aumenta.

Também, à medida que o número k de grupos aumenta, o tempo de processamento do BRKGASOMA diminui. Já em relação à formulação, o tempo de processamento aumenta substancialmente à medida que o tamanho da base (n) aumenta. No caso do BRKGASOMA, o tempo de execução é bem impactado pelo cálculo da função objetivo. Mais especificamente, quando o número de grupos é muito pequeno, a quantidade de objetos em cada grupo é maior e, assim sendo, o número de combinações a serem geradas e utilizadas no cálculo da função torna-se demasiadamente grande, aumentando o tempo de processamento. O aumento do tempo de processamento da formulação está diretamente associado ao número de variáveis, que é da ordem de k.n2+k.n (Brito e Brito, 2008) . Considerando, por exemplo, n = 100 e k = 2, temos 20.200 variáveis no problema. Já no caso de n = 500, são 751.500 variáveis.

Os tempos médio e mediano de processamento para os 28 resultados apresentados do BRKGASOMA foram, respectivamente, de 195 segundos (3 minutos e 15 segundos) e 136 segundos (2 minutos e 16 segundos).

Os resultados apresentados na seção anterior indicam que o algoritmo proposto neste trabalho pode ser uma boa alternativa à solução do problema de agrupamento apresentado neste trabalho. Além disso, um novo estudo pode ser conduzido com diferentes bases de dados de tamanhos maiores a fim de se verificar o incremento no tempo de processamento de acordo com o tamanho das bases utilizadas.

Em um trabalho futuro pode-se analisar a eficácia e a eficiência do BRKGASOMA, considerando a utilização de outros decodificadores.

## REFERÊNCIAS


BRITO, J. A. M.; BRITO, L. R.. Algoritmos Vns e Genéticos Aplicados ao Problema de Agrupamento com Soma Mínima de Distâncias. Anais do XL Simpósio Brasileiro de Pesquisa Operacional, 2008.

FADEL, AUGUSTO. Um Estudo da Aplicação de Técnicas de Combinação de Agrupamentos. Rio de Janeiro: Monografia. Escola Nacional de Ciências Estatísticas, 2013.

HAIR J. F.JR.; ANDERSON R. E.; TATHAM R. L.; BLACK W. C. Análise Multivariada de Dados Bookman, 5ª edição, 2005.

JOHNSON, R. A., and DEAN W. WICHERN. Applied multivariate statistical analysis. Essex: Pearson Education Limited, 2007.

KAUFMAN, LEONARD, and PETER J. ROUSSEEUW. Finding groups in data: an introduction to cluster analysis. Vol. 344. John Wiley & Sons, 2009.

NASCIMENTO M. C. V., et al. Investigation of a new GRASP-based clustering algorithm applied to biological data. Computers and Operations Research, 2009, doi: 10.1016/j.cor.2009.02.014

SERPA D. R.. Clusterização de dados biológicos através da metaheurística VNS. Dissertação. Instituto Nacional de Pesquisas Espaciais – INPE, 2009.