An Effective VNS Algorithm for k-Medoids Clustering Problem

José André de Moura Brito, Gustavo Silva Semaan and Augusto Cesar Fadel

***Abstract*—This paper proposes an algorithm based on VNS metaheuristcs for k-medoids clustering, which is a NP-hard optimization problem. The VNS algorithm was applied in fifty data bases (instances) with small, medium, and large sizes, considering the number of clusters between 2 and 7. The obtained results from these experiments show the effectiveness of this approach, comparing it with nine other related clustering algorithms and an optimization formulation. Furthermore, we found that our algorithm obtained the optimal solutions for the vast majority of the cases.**

***Index Terms*—k-medoids clustering, metaheuristcs, VNS, integer programming.**

I. Introdução

A análise de agrupamentos é, atualmente, uma ferramenta indispensável para abordar e resolver uma variada gama de aplicações reais associadas às áreas de Medicina, Estatística, Computação, Engenharia etc [1], [2]. De acordo com [2], a análise de agrupamentos é uma técnica de Estatística Multivariada, que contempla um conjunto de algoritmos que são aplicados para segmentar bases de dados, compostas por n registros (objetos) com f variáveis, em k grupos, ou seja, resolver o problema de agrupamento (PA).

Ao aplicar tais algoritmos tem-se, como premissa, a formação de grupos com alto grau de homogeneidade internamente (coesão) e baixo grau de homogeneidade externa, i.e., entre grupos distintos (separação) [3]. Por sua vez, para definição de tais grupos e, consequentemente, avaliação da solução, utiliza-se uma função objetivo [4]–[7] baseada em alguma métrica como, por exemplo, as distâncias Euclidiana ou de Manhattan.

Todavia, independentemente da métrica e da função objetivo consideradas [4]–[6], [8], [9], a obtenção do ótimo global para problemas de agrupamento (PA) consiste em uma tarefa computacionalmente árdua [10]–[13]. Adicionalmente, o tamanho do conjunto S, correspondente ao espaço de soluções viáveis para o PA, é definido pelo número de Stirling de Segundo Tipo (ST) [8], conforme Eq. (1):

$$ST(n,k) = \frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j}C_k^j j^n \qquad (1)$$

É possível observar a partir da Fig. 1 que, mesmo ao fixar $k = 2$ (dois grupos), à medida que a quantidade de objetos n aumenta, os valores de ST crescem exponencialmente, tornando inviável a resolução do PA clássico mediante a aplicação de um algoritmo de força bruta.

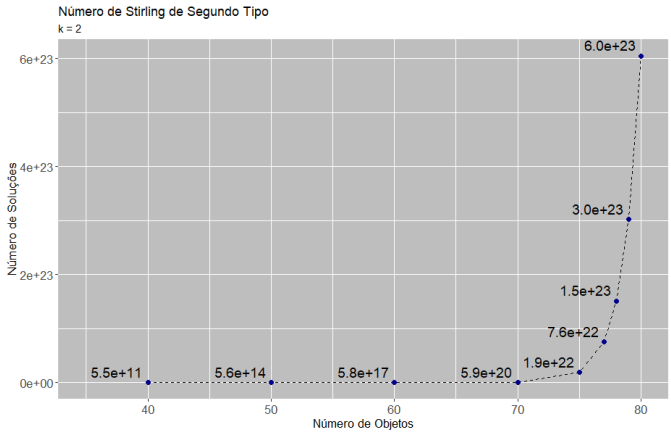

Fig. 1. Número de Stirling de segundo tipo ($k = 2$).

Considerando a complexidade dos PAs e suas variadas aplicações, nas últimas décadas, além do uso de algoritmos de agrupamento de uso mais geral - hierárquicos e não-hierárquicos [2], [8] - foram propostos diversos algoritmos baseados em metaheurísticas [11], [14]–[16] e formulações de programação matemática [4], [17], [18].

Em particular, neste trabalho é proposto um novo algoritmo para o problema de agrupamento com k-medoids [19]. Um medoid é definido, basicamente, como o objeto do grupo para o qual a soma das distâncias em relação aos demais objetos do mesmo grupo é mínima. De acordo com [19] e [20], algoritmos baseados em medoids produzem grupos de maior qualidade no que diz respeito à homogeneidade interna, mais robustos à existência de *outliers* ou ruídos e são de uso mais geral, sendo aplicáveis em bases de dados cujos objetos têm atributos quantitativos e/ou qualitativos.

Embora, nas últimas décadas, tenham sido propostos diversos algoritmos para este particular problema de agrupamento, em geral eles produzem soluções correspondentes a ótimos locais de qualidade razoável.

Com o objetivo de produzir soluções de boa qualidade, este artigo propõe um algoritmo baseado na metaheurística VNS (do inglês *Variable Neighborhood Search*, Busca em Vizinhança Variável) [21]–[23] e que tem os seguintes diferenciais: (i) Eficácia (qualidade) associada à definição de estrutura de vizinhanças e procedimentos de busca local e perturbação que garantem a produção de soluções melhores, muitas vezes ótimos globais; (ii) Eficiência (velocidade) quando comparado

José A. M. Brito is with ENCE/IBGE, Escola Nacional de Ciências Estatísticas Brasil, e-mail:jambrito@gmail.com.

Gustavo Silva Semaan is with INFES/UFF, Instituto do Noroeste Fluminense de Educação Superior, e-mail:gustavosemaan@id.uff.br.

Augusto Fadel is with IBGE, Instituto Brasileiro de Geografia e Estatística, e-mail: augustofadel@gmail.com.

com outras abordagens baseadas em metaheurísticas e (iii) Estabilidade do algoritmo, ratificada pelos experimentos computacionais.

Além da introdução, este artigo traz, na seção II, a descrição do problema de agrupamento com k-medoids. Nas seções III e IV apresenta-se, respectivamente, uma revisão dos trabalhos da literatura associados a este problema e uma descrição da metaheurística VNS e do algoritmo proposto. Na seção V são apresentadas as 50 bases de dados utilizadas nos experimentos computacionais e a calibração dos parâmetros de entrada do algoritmo VNS. Na seção VI são apresentados os resultados obtidos nos experimentos e suas análises. Por fim, a seção VII traz as conclusões e os trabalhos futuros.

II. Problema de Agrupamento com k-Medoids

Dado um conjunto X formado por n objetos com f variáveis, tal que $X = \{x_1, ..., x_i, ..., x_n\}$, deve-se selecionar, desse conjunto, k objetos que definam os medoids utilizados para formação de k grupos definidos por $G_1, ..., G_k$ e que, por sua vez, satisfaçam as seguintes restrições:

$$|G_r| \geq 1, r = 1, ..., k \tag{2}$$

$$\bigcup_{r=1}^{k} G_r = X \tag{3}$$

$$G_r \cap G_l = \emptyset, r \neq l, 1 \leq r, l \leq k \tag{4}$$

Adicionalmente, define-se $M = \{m_1, ..., m_k\}$ $(M \subset X)$ como o conjunto dos medoids, sendo cada elemento de M correspondente ao índice do objeto de X selecionado como o medoid do respectivo grupo. Os elementos de M são determinados de forma que seja mínima a soma das distâncias (Euclideana, Manhattan etc) dos $(n - k)$ objetos restantes de X ao seu medoid mais próximo, o que equivale a minimizar a função objetivo da Eq. 5:

$$fobj = \sum_{r=1}^{k} \sum_{\forall x_i \in G_r} d_{m_r i} \tag{5}$$

Para exemplificar o problema, considere um conjunto X com $n = 7$ objetos e a definição de dois medoids $(k = 2)$. Na matriz de distâncias $D = [d_{ij}]_{7\times7}$ apresentada na Fig. 2, cada entrada d_{ij} contém o valor da distância entre dois objetos x_i, x_j quaisquer de X. Neste exemplo, são apresentadas duas soluções viáveis (conjuntos de medoids) denotadas, respectivamente, por $M_1 = \{2, 4\}$ e $M_2 = \{1, 5\}$. Considerando essas soluções e a função objetivo definida na Eq. 5, ao efetuar-se a alocação dos 5 objetos restantes de X ao grupo cujo medoid associado esteja mais próximo, são obtidos os seguintes valores para função definida em Eq. (5): $fobj_1 = 30$ e $fobj_2 = 25$. Neste caso, a melhor solução está associada a M_2, sendo $G_1 = \{1, 2\}$ e $G_2 = \{5, 3, 4, 6, 7\}$.

De acordo com [24], assim como outros PA's, o problema dos k-medoids é NP-difícil. Tal característica direciona, naturalmente, para a adoção de abordagens baseadas em algoritmos heurísticos que, mesmo sem garantir a obtenção de soluções ótimas globais, podem produzir soluções que são ótimos locais

	1	2	3	4	5	6	7
1	0	7	22	18	13	14	10
2	7	0	29	25	20	21	17
3	22	29	0	4	9	8	12
4	18	25	4	0	5	4	8
5	13	20	9	5	0	1	3
6	14	21	8	4	1	0	4
7	10	17	12	8	3	4	0

Fig. 2. Matriz de Distâncias.

de boa qualidade, demandando baixo tempo (custo) computacional [25]. Ainda nesse sentido, para bases de dados com número de objetos da ordem de até centenas, pode-se utilizar a formulação apresentada a seguir - proposta inicialmente em [26], cuja resolução se dá mediante aplicação de métodos exatos como *Branch and Cut* ou *Branch and Bound* [27]. Todavia, por conta do seu número quadrático de variáveis (n^2+n) e restrições (n^2+n+1), a resolução de tal formulação pode demandar um tempo computacional expressivo, por vezes inaceitável.

Nesta formulação, y_i é uma variável 0-1 que assume valor 1 se o $i-$ésimo objeto de X $(i = 1, ..., n)$ é definido como medoid e zero em caso contrário; x_{ij} também é uma variável 0-1 que assume valor 1 se o $j-$ésimo objeto é alocado ao grupo definido pelo medoid i. A função objetivo em Eq. (6) minimiza a soma das distâncias dos objetos dos grupos aos seus respectivos medoids. A restrição Eq. (7) garante que cada objeto j deve ser associado a um único medoid. A restrição Eq. (8) garante que um objeto j somente pode ser associado a um objeto i se este objeto for definido como medoid. A restrição da Eq. (9) garante que o número de medoids é igual a k e na Eq. (10) temos as restrições de integralidade.

$$Minimize \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \tag{6}$$

$$\sum_{i=1}^{n} x_{ij} = 1, j = 1, \ldots, n \tag{7}$$

$$x_{ij} \leq y_i, \ i = 1, \ldots, n, j = 1, \ldots, n \tag{8}$$

$$\sum_{i=1}^{n} y_i = k \tag{9}$$

$$y_i, x_{ij} \in \{0, 1\}, i, j = 1, \ldots, n \tag{10}$$

III. Trabalhos Relacionados

Um algoritmo bem conhecido (década de 90) e muito utilizado para este problema até os dias atuais, denominado PAM [19], determina os k-medoids e os respectivos grupos, mediante a aplicação de dois procedimentos, sejam eles: *Build* correspondente a uma solução gulosa e *Swap* correspondente a uma busca local. Uma variante desse algoritmo, denominada CLARA (Clustering Large Applications) [19], consiste, basicamente, na combinação de um procedimento de amostragem aleatória simples e do algoritmo PAM.

Além desses dois algoritmos, nas últimas décadas, foram desenvolvidas propostas mais sofisticadas, em que algumas

eram mais focadas na eficiência (velocidade) e outras na eficácia (qualidade das soluções). Em [20] é proposta uma versão modificada do algoritmo CLARA, denominada CLARANS, que tem por base conceitos de Teoria dos Grafos e utiliza uma técnica de computação mais intensiva para determinação dos medoids. Em [28] os medoids são definidos em função da maximização do índice de silhueta [3]. Já em [29], com objetivo de reduzir o tempo computacional demandado pelo algorito PAM e produzir soluções de melhor qualidade, é proposto um algoritmo denominado CLATIN, que introduz uma modificação no procedimento de *swap*. Em [30] o algoritmo PAM é revisitado, sendo também propostas melhorias no procedimento de swap. Em [31] é apresentado um algoritmo rápido e que utiliza algumas ideias do algoritmo k-means para definir os medoids iniciais, e em [32] é apresentada uma heurística lagrangiana para o problema dos k-medoids.

No algoritmo proposto em [33], em cada iteração os medoids são atualizados para o objeto mais dissimilar em relação aos demais objetos do grupo e, uma vez atingido o número máximo de iterações, cada objeto é alocado ao grupo cujo medoid é mais similar. Segundo os autores, tal estratégia possibilita escapar de ótimos locais de baixa qualidade.

Com base em trabalhos mais recentes, em [34] é proposto um algoritmo que utiliza uma medida de variância para determinar os medoids, com foco na eficiência. Em [35] é feita a proposta de um algoritmo paralelizado, denotado PAMAE, que pode ser aplicado em bases de dado de grande porte e atingir uma boa acurácia e eficiência. Em [36] são propostas versões mais rápidas dos algoritmos PAM, CLARA e CLARANS, a partir do aperfeiçoamento do procedimento de *swap* utilizado no algoritmo PAM. Em [37] também é proposto um algoritmo paralelizado para o problema dos k-medoids, mas considerando o número de grupos variável. O trabalho de [38] traz a proposta de um algoritmo que seleciona os medoids iniciais utilizando uma função de variância e conceitos de densidade. Em [39] é desenvolvido um algoritmo que produz bons limites primais e duais para o problema dos k-medoids e que tem por base o método do subgradiente.

De modo adicional, são apresentados trabalhos baseados na aplicação de metaheurísticas, como o Algoritmo Genético proposto por [40] para bases de dados de grande porte. Em [41] é apresentado um algoritmo genético híbrido - denominado HKA, que combina a aplicação de um operador de cruzamento com um procedimento de busca local baseado no algoritmo k-means. Em um trabalho relacionado [42] é proposta uma variante do algoritmo genético apresentado em 2004 por [41] e que resolve problema dos k-medoids sem considerar o valor de k fixo, utilizando, para isso, a combinação de um operador de cruzamento com o índice de *Davies-Bouldin*. No trabalho de [43] é proposto um algoritmo baseado na metaheurística BRKGA (*Biased Random Key Genetic Algorithm*) e em [44] um algoritmo híbrido baseado no CRO (*Chemical Reaction Optimization*).

IV. Metaheurística VNS e Algoritmo Proposto

A Busca em Vizinhança Variável [21], [22], é uma metaheurística que possibilita uma exploração eficiente do espaço de soluções viáveis S de problemas de otimização combinatória, mediante a troca sistemática de estruturas de vizinhança. O VNS se diferencia de outras metaheurísticas baseadas em métodos de busca local por não considerar uma trajetória, mas por explorar vizinhanças gradativamente mais distantes da solução atual, focando a busca local em uma região em torno de uma nova solução somente se há melhoria no valor da função objetivo. Adicionalmente, a busca local implementada no VNS deve realizar uma sequência de modificações em uma vizinhança de uma solução, procurando melhorar o valor da função objetivo até que um ótimo local seja encontrado. Contudo, em muitos casos, o ótimo local está distante do ótimo global, sendo necessário analisar o valor da função objetivo em outras vizinhanças, a fim de encontrar soluções melhores. Por fim, a definição da estrutura de vizinhanças está intrinsecamente associada ao problema de otimização em questão, tendo impacto direto na qualidade das soluções produzidas na busca local. A descrição detalhada dessa metaheurística e de suas variantes pode ser encontrada em [21]–[23].

A Fig. 3 traz o pseudocódigo do VNS básico (problema de minimização). Na linha 3, constrói-se uma solução inicial de forma aleatória ou mediante aplicação de alguma heurística de construção. Em cada iteração do VNS seleciona-se (linha 8), mediante aplicação de um procedimento de perturbação, uma solução qualquer s' contida na vizinhança $N_v(s)$ associada à solução atual s. Em seguida, aplica-se uma busca local sobre a solução s' (linha 9). Caso o valor de função objetivo da solução ótima local s'' ($f(s'')$) seja inferior ao valor de $f(s)$, é realizada a atualização da solução atual s com s'' e retorna-se à primeira vizinhança $N_1(s)$. Caso contrário, incrementa-se a ordem de estrutura de vizinhança (linha 15), gera-se um novo vizinho em relação à solução s e aplica-se novamente a busca local. O critério de parada comumente adotado (linha 4) nesta metaheurística é o número de iterações.

```
1  Defina as estruturas de vizinhança N_v , v=1,...,v_max
2  niter←0
3  s←Construa_solucão
4  Enquanto (niter<MAXITER) Faça
5     niter←niter+1
6     v←1
7     Enquanto (v<v_max) Faça
8           s' ← Perturbação (s,N_v)
9           s"←Busca_Local (s',N_1)
10          Se (f(s") < f(s)) Então
11             s ← s"
12             v←1
13             niter←0
14          Senão
15             v←v+1
16          Fim-se
17       Fim-Enquanto
18 Fim-Enquanto
19 Retorne s
```

Fig. 3. Algoritmo VNS.

Apresenta-se a descrição da estrutura de vizinhança e dos procedimentos de construção, perturbação e busca local propostos para o problema dos k-medoids.

A. Algoritmo VNS

No algoritmo VNS proposto para o problema de agrupamento com k-medoids, denominado VNSKMED, cada solução é representada por um vetor s com k posições correspondentes aos medoids, ou seja, $s = (m_1, m_2, ..., m_k)$ (conjunto dos medoids M definido na seção II). No procedimento de construção são gerados g vetores s - objetos associados aos medoids que são selecionados aleatoriamente de X, sendo definido como solução inicial s_o aquela que produz o menor valor para a função objetivo da Eq. (5).

A estrutura de vizinhanças $N_v(s)$ é definida da seguinte maneira: dada uma solução s, toma-se, para cada um dos seus medoids, os p objetos $x_i \in X$ mais próximos (de acordo com a distância euclidiana). Os índices desses objetos são armazenados em uma matriz $M_{k\times p}$. Além disso, uma solução $s' \in N_v(s)$ difere de s por, exatamente, v medoids.

A fim de ilustrar essa estrutura de vizinhanças, suponha $n = 30$, $k = 3$, $s = (2, 7, 8)$, $p = 3$ e a matriz M^1 com os índices dos três objetos mais próximos (em ordem crescente) de cada um dos medoids de s.

$$M^1 = \begin{bmatrix} 1 & 11 & 14 \\ 16 & 5 & 10 \\ 9 & 20 & 19 \end{bmatrix}$$

Considerando estas informações, uma possível solução na vizinhança $N_1(s)$, produzida a partir de uma perturbação em s pode ser $s' = (\underline{\mathbf{11}}, 7, 8)$. De igual forma, no caso de $N_2(s)$, outra solução possível é: $s' = (\underline{\mathbf{1}}, 7, \underline{\mathbf{20}})$.

Na busca local, considerando a mesma definição de estrutura de vizinhança, constrói-se uma matriz $M^2_{k\times l}$ com os l elementos (objetos) mais próximos de cada um dos k medoids de s'. Em seguida, toma-se os C_k^{k-1} subconjuntos de medoids de s' e combina-se cada um desses subconjuntos com cada um dos elementos de M^2, produzindo-se, no total, q soluções s_j ($q = k^2 l$). Por fim, em um passo posterior, calcula-se o valor da função objetivo da Eq. (5) para cada solução s_j e define-se $w = argmin_{j=1,...,q} f(s_j)$ e $s'' = s_w$, ou seja, é considerada a estratégia *best improvement*. Caso o valor de $f(s'') < f(s)$, sendo s a melhor solução obtida até o momento, define-se $s = s''$ e repete-se o procedimento de busca local, até que $f(s'') \geq f(s)$.

Utilizando, como exemplo, a solução $s' = (11, 7, 8)$, $l = 2$ e a matriz M^2, ao aplicar a busca local, são produzidas as dezoito soluções s_j apresentadas na Fig. 4. A definição dos valores de g e l será objeto de discussão na seção V, onde é descrito um experimento de calibração de parâmetros.

$$M^2 = \begin{bmatrix} 4 & 27 \\ 16 & 5 \\ 9 & 20 \end{bmatrix}$$

V. Bases de Dados e Calibração de Parâmetros

De forma a avaliar a performance do VNSKMED frente a um conjunto de nove algoritmos da literatura e à formulação descrita na seção II, foram utilizadas 50 bases de dados diversificadas quanto ao número de objetos (n) e variáveis (f). Nessas bases, n varia entre 49 e 5.000 e f varia entre 1 e 1.213, conforme apresentado na Tabela I.

4	**7**	**8**
27	**7**	**8**
16	**7**	**8**
5	**7**	**8**
9	**7**	**8**
20	**7**	**8**
11	4	**8**
11	27	**8**
11	16	**8**
11	5	**8**
11	9	**8**
11	20	**8**
11	**7**	4
11	**7**	27
11	**7**	16
11	**7**	5
11	**7**	9
11	**7**	20

Fig. 4. Soluções s_j produzidas na Busca Local.

Ainda nesse sentido, tendo por objetivo a comparabilidade e reprodutibilidade dos experimentos, a função em R que implementa o algoritmo VNSKMED e todas as bases estão disponíveis em https://github.com/jambrito/VNSKMED.

TABELA I
INFORMAÇÕES SOBRE AS BASES DE DADOS.

Base	n	f	Base	n	f
2-FACE*	200	2	IONOSPHERE	351	34
200DATA	200	2	IRIS	150	4
400P3C	400	2	MARONNA	200	2
A1	3.000	2	MORESHAPES	489	2
AGGREGATION	788	2	NEW-THYROID	215	5
BANKNOTE	1.372	5	NORMAL300	300	2
BREASTB	49	1.213	NUMBERS2	540	2
BROKEN-RING*	800	2	OUTLIERS	131	2
BUPA*	345	6	PARKINSONS	195	23
CHART	600	60	PIB.MINAS	853	1
COMPOUND	399	2	PIB100	100	1
CONCRETEDATA	1.030	9	PRIMA.INDIANS	569	8
DBLCA	141	661	RUSPINI	75	2
DBLCB	180	661	SONAR	208	60
DOWJONES	750	4	SPHERICAL4D3C	400	3
ECOLI	336	7	SPRDESP	645	2
FACE	296	2	SYNTHETICCONTROL	600	51
FORESTFIRES*	517	7	TRIPADVISOR	980	10
GAMMA400	500	3	UNIFORM400	400	2
GAUSS9*	900	2	UNIFORM700*	700	2
GLASS	214	9	VOWEL2*	528	2
HABERMAN	306	3	WAVEFORM21	5.000	21
HAYES-ROTH	132	6	WDBC	569	30
INDIAN	583	9	WINE	178	13
INDOCHINACOMBAT	72	4	YEAST	1.484	7

Conforme apresentado anteriormente, um fator que contribui para boa performance de qualquer metaheurística é a determinação dos valores associados ao seu conjunto de parâmetros. No caso do algoritmo VNSKMED, tais valores foram definidos a partir de um experimento preliminar de calibração. Mais especificamente, foram selecionadas sete bases (assinaladas com asterisco na Tabela I), dentre as 50 bases de dados, sendo o VNSKMED executado 5 vezes para cada base, considerando $k \in \{3, 4, 5\}$ e 192 combinações dos parâmetros v_{max}, $MAXITER$, p e l - perfazendo 20.160 execuções (n^o bases x valores de k x combinações dos parâmetros x 5) do algoritmo. A descrição dos parâmetros e valores considerados neste experimento são apresentados na Tabela II.

TABELA II
PARÂMETROS CALIBRADOS NO ALGORITMO VNSKMED.

Parâmetro	Descrição	Valores
v_{max}	Máximo de Vizinhanças	$\{2, 3, 4, 5\}$
$MAXITER$	Máximo de Iterações sem Melhoria	$\{5, 10, 15, 20\}$
p	N^o Vizinhos dos medoids na Perturbação	$\{5, 10, 15\}$
l	N^o Vizinhos dos medoids na Busca Local	$\{10, 30, 50, 100\}$

Considerando cada combinação destes quatro parâmetros, bases de dados e valores de k, foi calculada a média dos valores de função objetivo (Eq. (5)) obtidos nas 5 execuções. Em seguida, tomando-se como combinação vencedora aquela correspondente ao maior número de soluções com menores valores da média (em todas as bases e valores de k), chegou-se à seguinte quádrupla: $v_{max} = 3$ (correspondente aos melhores valores, em particular, para $k = \{4, 5\}$), $MAXITER = 10$, $p = 10$ e $l = 30$. O parâmetro g, correspondente ao número de soluções geradas mediante a aplicação do procedimento de construção, foi fixado a priori como 20.

VI. EXPERIMENTOS COMPUTACIONAIS

Nesta seção são apresentados resultados relativos à aplicação do algoritmo VNSKMED, da formulação descrita na seção II e de nove algoritmos da literatura, a saber: PAM e CLARA [19], FASTPAM, FASTCLARA e FASTCLARANS [36], HKA [41], PARK [31], RANK [33] e BRKGA [43].

Os algoritmos VNSKMED e HKA foram implementados utilizando a linguagem de programação R [www.r-project.org] [45] e o código do algoritmo BRKGA (em R) foi cedido pelos autores. Os demais algoritmos estão disponíveis em funções implementadas em dois pacotes do R, conforme Tabela III. A formulação foi implementada utilizando o solver de otimização Gurobi (versão 9.1), disponível no pacote gurobi no R. Todos os experimentos relacionados à aplicação dos nove algoritmos e da formulação foram realizados em um computador com 16GB de memória RAM e dotado de processador AMD FX-6300 com 3.50 GHz.

TABELA III
PACOTES ASSOCIADOS AOS ALGORITMOS.

Algoritmo	Pacote
PAM, CLARA	cluster
FASTPAM, FASTCLARA, FASTCLARANS	fastkmedoids

A. Experimento I

O VNSKMED e os nove algoritmos foram aplicados nas 50 bases de dados da Tabela I, considerando $k \in \{2, ..., 7\}$ (300 soluções por algoritmo). Além disso, utilizando o mesmo intervalo de variação de k e o tempo máximo de execução de 3 horas para o solver Gurobi, a formulação foi aplicada em 49 bases de dados, excetuando-se a base WAVEFORM21 onde não foi possível aplicar o solver, por conta de um erro de memória. A partir do uso do Gurobi, foram obtidos os ótimos globais para 48 bases para todos os valores de k. O único caso onde não foi produzido o ótimo global, considerando o tempo de 3 horas, diz respeito à base A1.

Os parâmetros do VNSKMED foram definidos no experimento de calibração reportado na seção V. Para os algoritmos HKA e BRKGA foram adotados os valores de parâmetro apresentados, respectivamente, em [41] e [43]. Por fim, em relação aos demais algoritmos da literatura, foram considerados os valores padrões das funções disponíveis nos pacotes listados na Tabela III.

A partir deste experimento obteve-se o valor de função objetivo (Eq. (5)) e o tempo de processamento. Os valores da função objetivo foram utilizados para calcular, por número de grupos: (i) percentual de ótimos globais produzidos pelos algoritmos, tendo por base o total de ótimos globais produzidos pela formulação no tempo máximo de 3 horas; (ii) percentual de melhores soluções (vencedoras): melhor solução produzida considerando os nove algoritmos e a formulação, não necessariamente correspondente a um ótimo global e (iii) as estatísticas resumo calculadas a partir dos gaps relativos (Eq. (11)) obtidos a partir da diferença entre a solução vencedora (s_{best}) e a solução produzida por cada um dos algoritmos e a formulação (s_{algf}), para as 50 bases de dados (no caso da base WAVEFORM1 foi considerada, como a solução vencedora, aquela associada a, pelo menos, um dos nove algoritmos) e $k \in \{2, ..., 7\}$.

$$gap = 100.(s_{algf} - s_{best})/s_{best} \quad (11)$$

Analisando a Tabela IV verifica-se a eficácia do VNSKMED frente aos demais algoritmos, tendo em vista seu o percentual de ótimos globais produzidos - função do total de ótimos globais - Nglobal (em negrito) produzidos pela formulação. Em particular, para $k = 2$ e $k = 3$, o VNSKMED produziu o ótimo em 100% dos casos e o percentual mais baixo desse algoritmo, da ordem de 92%, ocorreu para $k = 6$. Adicionalmente, os algoritmos BRKGA, PAM, PAMF e HKA foram os que apresentaram os percentuais de ótimos globais mais próximos do VNSKMED. O cenário mais favorável a esses quatro algoritmos ocorreu para $k = 2$, quando as diferenças foram, em pontos percentuais, de, respectivamente, 4.2% e 10.4% (HKA e BRKGA) e 20.8% (PAM e PAMF). Os algoritmos RANK, CLARANSF e CLARA produziram os menores percentuais.

TABELA IV
PERCENTUAIS DE ÓTIMOS GLOBAIS POR ALGORITMO E k.

Algoritmo	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$
BRKGA	89.6	87.5	68.8	50.0	33.3	37.5
CLARA	8.3	2.1	4.2	2.1	4.2	2.1
CLARAF	22.9	18.8	6.2	6.2	0.0	0.0
CLARANSF	6.2	4.2	6.2	2.1	0.0	0.0
HKA	95.8	77.1	58.3	39.6	29.2	20.8
PAM	79.2	72.9	70.8	58.3	47.9	52.1
PAMF	79.2	70.8	70.8	54.2	47.9	52.1
PARK	33.3	14.6	14.6	6.2	2.1	0.0
RANK	2.1	0.0	0.0	0.0	0.0	0.0
VNS	100.0	100.0	97.9	97.9	91.7	93.8
Nglobal	**48**/50	**48**/50	**48**/50	**48**/50	**48**/50	**48**/50

Por fim, observa-se que a performance dos nove algoritmos da literatura é fortemente impactada à medida que k aumenta, sendo VNSKMEDS o único algoritmo estável.

A Tabela V traz os gaps médios (Me) e medianos (Md) entre as soluções, onde as células em negrito e as células em itálico

e sublinhado correspondem, respectivamente, a valores de gap com média (Me) e mediana (Md) nos seguintes intervalos $[0, 0.1\%]$ e $(0.1\%, 0.5\%]$. O VNSKMED produziu, em geral, valores de gaps de 0%. Em particular, analisando-se o gap médio, o pior resultado do VNSKMED foi de 0.4% para $k = 6$ e $k = 7$. Além disso, os algoritmos BRKGA, PAM e PAMF são os que apresentaram gaps mais próximos aos gaps do VNSKMED (inferiores a 1.0%), seguidos pelo algoritmo HKA. Por fim, o algoritmo RANK foi o que apresentou os piores resultados, independentemente do número de grupos, o que pode ser constatado pelos seus gaps percentuais que variaram entre 24.1% e 47.7% no caso médio e entre 14.6% e 28.0% no caso mediano.

TABELA V
GAPS PERCENTUAIS POR ALGORITMO E k.

	k=2		k=3		k=4	
Probs	Md	Me	Md	Me	Md	Me
BRKGA	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.1**
CLARA	1.4	1.7	2.2	3.0	4.0	4.1
CLARAF	*0.2*	*0.5*	*0.4*	1.0	0.8	1.5
CLARANSF	1.2	1.4	1.5	2.3	1.9	3.0
HKA	**0.0**	**0.0**	**0.0**	**0.1**	**0.0**	*0.3*
PAM	**0.0**	*0.3*	**0.0**	*0.2*	**0.0**	*0.3*
PAMF	**0.0**	0.3	**0.0**	0.3	**0.0**	*0.3*
PARK	*0.4*	2.5	1.1	3.5	1.8	5.9
RANK	14.6	24.1	19.8	32.2	25.5	34.9
VNSKMED	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**

	k=5		k=6		k=7	
Probs	Md	Me	Md	Me	Md	Me
BRKGA	**0.0**	*0.3*	*0.2*	*0.3*	*0.2*	*0.4*
CLARA	4.8	5.3	5.8	7.2	6.3	6.8
CLARAF	1.1	1.5	1.4	2.0	1.8	2.5
CLARANSF	2.5	3.9	3.2	4.5	2.8	5.0
HKA	**0.1**	*0.5*	*0.2*	1.3	0.6	1.5
PAM	**0.0**	*0.3*	**0.0**	0.4	**0.0**	*0.5*
PAMF	**0.0**	*0.3*	**0.0**	*0.4*	**0.0**	*0.5*
PARK	4.7	7.6	4.1	9.0	6.2	13.6
RANK	26.9	37.0	28.0	47.7	26.3	42.9
VNSKMED	**0.0**	**0.1**	**0.0**	*0.4*	**0.0**	*0.4*

A Fig. 5 traz, por valor de k e em relação aos melhores algoritmos, os percentuais de melhores soluções produzidas. Mais uma vez, o algoritmo VNSKMED teve performance superior aos demais algoritmos, com percentuais entre 92% e 100%, sendo seguido pelos algoritmos BRKGA, PAM e HKA.

Além da análise da eficácia do algoritmo VNSKMED, apresenta-se, na Tabela VI, uma análise de sua eficiência, tendo por base os tempos de processamento (em segundos) desse algoritmo e dos algoritmos BRKGA e HKA, além da formulação. Tal análise ficou restrita aos três algoritmos, uma vez que, por serem baseados em metaheurísticas, demandam uma computação intensiva, o que implica, consequentemente, mais tempo de processamento na busca por soluções de boa qualidade. Os demais algoritmos considerados, são rápidos, com tempo de execução da ordem de poucos segundos, porém, em geral, produzem soluções de qualidade inferior àquelas alcançadas pelo VNSKMED, conforme pode ser observado nas Tabelas IV e V.

A partir da Tabela VI pode-se obervar que, entre os três algoritmos baseados em metaheurísticas, o VNSKMED foi, na maioria dos casos, mais eficiente tanto em termos da média quanto da mediana. Quando comparado à Formulação,

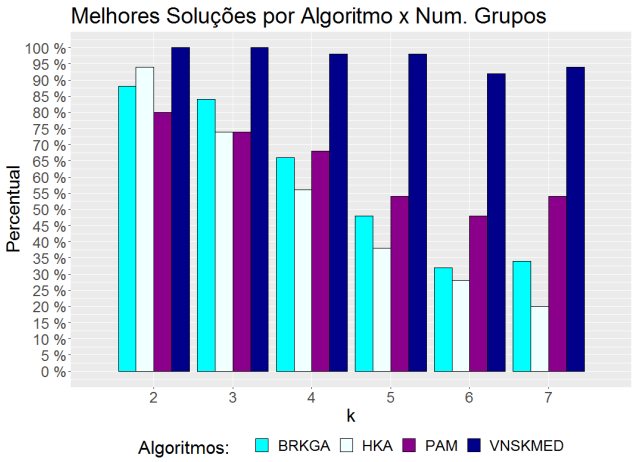

Fig. 5. Percentuais Melhores Soluções

observa-se que, para todos os valores de k, o VNSKMED apresentou, em média, tempos inferiores.

TABELA VI
ESTATÍSTICAS DOS TEMPOS DE PROCESSAMENTO.

		Média		
k	VNSKMED	HKA	BRKGA	Formulação
2	8.3	207.7	47.2	281.4
3	16.8	223.5	67.4	298.4
4	31.3	244.1	74.4	323.3
5	50.4	269.5	85.9	318.4
6	80.6	277.1	89.7	307.6
7	111.8	286.6	94.1	310.0

		Mediana		
k	VNSKMED	HKA	BRKGA	Formulação
2	4.2	154.5	37.3	24.9
3	9.5	173.5	39.5	22.7
4	20.8	186.2	52.8	16.1
5	37.3	217.0	63.7	18.3
6	67.5	225.1	59.1	18.9
7	78.8	240.8	71.4	19.0

B. Experimento II

Para analisar a estabilidade do VNSKMED, algoritmo com a maior eficácia, foi realizado um segundo experimento com um subconjunto de 10 bases de dados apresentadas na Tabela VII e $k \in \{3, 4, 5, 6\}$ - totalizando 40 cenários. O algoritmo VNSKMED foi aplicado 10 vezes nessas bases (para cada k), sendo armazenados, em cada execução, o valor de função objetivo associado à melhor solução produzida.

Na Tabela VII, s_{best} corresponde ao valor da função objetivo obtido no Experimento I e para as demais colunas são apresentadas as estatísticas associadas aos valores de função objetivo obtidos a partir das 10 execuções do VNSKMED, quais sejam: mínimo (Min), média (Me), mediana (Md) e máximo (Max), além do coeficiente de variação (cv) em valores percentuais.

Analisando-se essa Tabela, verifica-se que, nos 40 cenários avaliados, o valor mínimo (Min) obtido para a função objetivo foi igual ao valor da solução produzida pelo VNSKMED no Experimento I (s_{best}). Além disso, em 39 dos 40 cenários o

coeficiente de variação foi igual a zero e com valor inferior a 0.03% em único um caso. Isso significa que, na maioria das execuções, o algoritmo produziu a mesma solução e que esta corresponde à melhor solução, uma vez que a solução média, em todos dos casos, foi igual à solução mínima.

TABELA VII
ESTATÍSTICAS DA FUNÇÃO OBJETIVO.

			$k=3$			
Bases	s_{best}	Min	Me	cv	Md	Max
BANKNOTE	1.2418	1.2418	1.2418	0.0000	1.2418	1.2418
BROKEN-RING	0.8039	0.8039	0.8039	0.0000	0.8039	0.8039
BUPA	1.7615	1.7615	1.7615	0.0000	1.7615	1.7615
CONCRETEDATA	2.4723	2.4723	2.4723	0.0000	2.4723	2.4723
FORESTFIRES	1.8502	1.8502	1.8502	0.0000	1.8502	1.8502
HABERMAN	1.1274	1.1274	1.1274	0.0000	1.1274	1.1274
IONOSPHERE	4.2167	4.2167	4.2167	0.0000	4.2167	4.2167
NEW-THYROID	1.2145	1.2145	1.2145	0.0219	1.2145	1.2153
NUMBERS2	0.7798	0.7798	0.7798	0.0000	0.7798	0.7798
WDBC	3.9913	3.9913	3.9913	0.0000	3.9913	3.9913
			$k=4$			
Bases	s_{best}	Min	Me	cv	Md	Max
BANKNOTE	1.1118	1.1118	1.1118	0.0000	1.1118	1.1118
BROKEN-RING	0.6089	0.6089	0.6089	0.0000	0.6089	0.6089
BUPA	1.6599	1.6599	1.6599	0.0000	1.6599	1.6599
CONCRETEDATA	2.2994	2.2994	2.2994	0.0000	2.2994	2.2994
FORESTFIRES	1.7314	1.7314	1.7314	0.0000	1.7314	1.7314
HABERMAN	0.9849	0.9849	0.9849	0.0000	0.9849	0.9849
IONOSPHERE	3.9613	3.9613	3.9613	0.0000	3.9613	3.9613
NEW-THYROID	1.0769	1.0769	1.0769	0.0000	1.0769	1.0769
NUMBERS2	0.6016	0.6016	0.6016	0.0000	0.6016	0.6016
WDBC	3.8405	3.8405	3.8405	0.0000	3.8405	3.8405
			$k=5$			
Bases	s_{best}	Min	Me	cv	Md	Max
BANKNOTE	1.0103	1.0103	1.0103	0.0000	1.0103	1.0103
BROKEN-RING	0.5150	0.5150	0.5150	0.0000	0.5150	0.5150
BUPA	1.5718	1.5718	1.5718	0.0000	1.5718	1.5718
CONCRETEDATA	2.1755	2.1755	2.1755	0.0000	2.1755	2.1755
FORESTFIRES	1.6585	1.6585	1.6585	0.0000	1.6585	1.6585
HABERMAN	0.8903	0.8903	0.8903	0.0000	0.8903	0.8903
IONOSPHERE	3.8079	3.8079	3.8079	0.0000	3.8079	3.8079
NEW-THYROID	0.9841	0.9841	0.9841	0.0000	0.9841	0.9841
NUMBERS2	0.5253	0.5253	0.5253	0.0000	0.5253	0.5253
WDBC	3.7149	3.7149	3.7149	0.0000	3.7149	3.7149
			$k=6$			
Bases	s_{best}	Min	Me	cv	Md	Max
BANKNOTE	0.9380	0.9380	0.9380	0.0000	0.9380	0.9380
BROKEN-RING	0.4683	0.4683	0.4683	0.0000	0.4683	0.4683
BUPA	1.5149	1.5149	1.5149	0.0000	1.5149	1.5149
CONCRETEDATA	2.0612	2.0612	2.0612	0.0000	2.0612	2.0612
FORESTFIRES	1.5903	1.5903	1.5903	0.0000	1.5903	1.5903
HABERMAN	0.8311	0.8311	0.8311	0.0000	0.8311	0.8311
IONOSPHERE	3.6632	3.6632	3.6632	0.0000	3.6632	3.6632
NEW-THYROID	0.9255	0.9255	0.9255	0.0000	0.9255	0.9255
NUMBERS2	0.4484	0.4484	0.4484	0.0000	0.4484	0.4484
WDBC	3.6003	3.6003	3.6003	0.0000	3.6003	3.6003

VII. CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi apresentado um algoritmo baseado na metaheurística VNS, que foi aplicado à resolução do problema de agrupamento com k-medoids. Para avaliar a performance do algoritmo foram realizados experimentos com 50 bases de dados variadas, comparando o VNSKMED com o clássico algoritmo PAM, suas variantes e outros algoritmos da literatura. Adicionalmente, foi utilizada uma formulação de programação inteira para resolver este problema, que permitiu avaliar o percentual de ótimos globais produzidos pelo algoritmo.

O VNSKMED, os algoritmos da literatura e a formulação foram aplicados nestas bases, de forma a produzir soluções com número de grupos variando entre 2 e 7. A partir das soluções produzidas, foram avaliados: percentuais de ótimos globais, percentuais de melhores soluções e os gaps relativos. Os resultados obtidos neste experimento mostraram a superioridade do VNSKMED frente aos demais algoritmos, inclusive em relação ao algoritmo PAM, considerado um dos mais eficazes da literatura para este problema, além dos algoritmos HKA e BRKGA, também baseados em metaheurísticas.

Em relação aos ótimos globais, o VNSKMED produziu, em geral, percentuais superiores a 91%. Em particular, foi obtido o percentual 100% para $k=2$ e $k=3$, sendo da ordem de 92% o menor percentual observado ($k=6$). Quanto às soluções de melhor qualidade, foram observados (Fig. 5) percentuais entre 100 e 92% e, concomitantemente, baixos valores para os gaps, tomando por base as soluções deste algoritmo e a melhor solução produzida.

No segundo experimento, onde o VNS foi aplicado 10 vezes em um subconjunto de 10 bases de dados, foi possível observar a estabilidade do algoritmo quanto à qualidade das soluções produzidas. Tal afirmação é corroborada pelos resultados da Tabela VII, em particular, a partir da avaliação conjunta da média e dos valores do coeficiente de variação dos valores obtidos para função objetivo.

Assim, a partir dos resultados apresentados neste artigo, onde foram considerados experimentos com um substancial conjunto de bases de dados, foi possível constatar a eficácia e a eficiência do VNSKMED frente a outros algoritmos da literatura, o que indica que esse algoritmo constitui uma boa alternativa para resolução do problema dos k-medoids.

Como trabalhos futuros, pretende-se desenvolver novos procedimentos de busca local e novas definições de vizinhança, além de um procedimento baseado em Reconexão por Caminhos [23] (*Path Relinking*), para produzir soluções de boa qualidade, demandando menor tempo de processamento. Outra possibilidade é resolver o problema dos k-medoids sem definir, a priori, o número de grupos, caracterizando o problema de agrupamento automático. Para atingir tal objetivo, pode-se fazer uma adaptação do VNSKMED, mediante a utilização da silhueta média [11] em combinação com a função objetivo, de forma a definir o número ideal de grupos.

REFERENCES

[1] D. Gunopulos, *Clustering Overview and Applications*. LIU L., ÖZSU M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, M, 2009.

[2] J. Hair, W. Black, and R. Babin, B.J.and Anderson, *Multivariate Data Analysis*. Cengage, 8th edition, 2018.

[3] J. Han, M. Kamber, and J. Pei, *Data Mining – Concepts and Techniques*. Elsevier, 3rd edition, 2012.

[4] P. Hansen and B. Jaumard, "Cluster analysis and mathematical programming," *Mathematical Programming*, vol. 79, pp. 191–215, 1997.

[5] J. Fiorucci, F. Toledo, and M. Nascimento, "Heuristics for minimizing the maximum within-clusters distance," *Pesquisa Operacional*, vol. 32, no. 3, pp. 497–522, 2012.

[6] J. A. M. Brito, A. C. Fadel, G. S. Semaan, and L. R. Brito, "Algoritmo genético de chaves aleatória viciadas aplicado ao problema de agrupamento com soma mínima," in *Anais do XLIX Simpósio Brasileiro de Pesquisa Operacional*, vol. 11, pp. 2325–2336, Sobrapo, 2017.

[7] J. A. M. Brito, A. C. Fadel, G. S. Semaan, and F. M. T. Montenegro, "Heuristics applied to minimization of the maximum-diameter clustering problem," *IEEE Latin America Transactions*, vol. 19, no. 4, pp. 652–659, 2021.

[8] R. Johnson and D. Wichern, *Applied Multivariate Statistical Analysis (Classic Version)*. Pearson, 6th edition, 2018.

[9] S. Zhu, L. Xu, and E. Goodman, "Evolutionary multi-objective automatic clustering enhanced with quality metrics and ensemble strategy," *Knowledge-Based Systems*, vol. 188(5), pp. 1–21, 2020.
[10] G. Semaan, A. Fadel, J. Brito, and L. Ochi, "A hybrid efficient heuristic with hopkins statistic for the automatic clustering problem," *IEEE Latin America Transactions*, vol. 17(1), pp. 7–17, 2019.
[11] G. Semaan, *Algoritmos para o Problema de Agrupamento Automático*. PhD thesis, Federal Fluminense University, 2013.
[12] A. J. O. Reyes, A. O. Garcia, and Y. L. Mue, "System for processing and analysis of information using clustering technique," *IEEE Latin America Transactions*, vol. 12(2), pp. 364–371, 2014.
[13] J. C. R. Thomas, M. S. Penãs, M. M. Cofre, and N. D. Carralero, "Performance analysis of clustering internal validation indexes with asymmetric clusters," *IEEE Latin America Transactions*, vol. 17(5), pp. 807 – 814, 2019.
[14] M. Nascimento, F. Toledo, and A. de Carvalho, "Investigation of a new grasp - based clustering algorithm applied to biological data," *Computers & Operations Research*, vol. 37, pp. 1381–1388, 2010.
[15] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithm," *Annals of Data Science*, vol. 2(2), pp. 165–193, 2015.
[16] R. Oliveira, A. Chaves, and L. Lorena, "A comparison of two hybrid methods for constrained clustering problems," *Applied Soft Computing*, vol. 54, pp. 256–266, 2017.
[17] M. Rao, "Cluster analysis and mathematical programming," *Journal of American Statistical Association*, vol. 1971, pp. 622–626, 1971.
[18] N. Negreiros, M.J.and Maculan, P. Batista, J. Rodrigues, and A. Palhano, "Capacitated clustering problems applied to the layout of it-teams in software factories," *Annals of Operations Research - doi.org/10.1007/s10479-020-03785-4*.
[19] L. Kaufman and P. Rousseeuw, *Finding Groups in Data – An Introduction to Cluster Analysis*. Wiley-Interscience, 1st edition, 1990.
[20] J. Han and R. Ng, "Clarans: A method for clustering objects for spatial datamining," *IEEE Transactions Knowledge of Data Enginnering*, vol. 14(5), pp. 1003–1016, 2002.
[21] N. Mladenović and P. Hansen, "Variable neighborhood search," *Computers & Operations Research*, vol. 24, no. 11, pp. 1097–1100, 1997.
[22] P. Hansen, N. Mladenović, and J. Perez, "Variable neighbourhood search: methods and applications," *Annals of Operations Research*, vol. 175, pp. 367–407, 2010.
[23] M. Gendreau and J.-Y. Potvin, *Handbook of Metaheuristics*. Springer, 3rd edition, 2019.
[24] N. Megiddo and K. J. Supowit, "On the complexity of some common geometric location problems," *SIAM Journal on Computing*, vol. 13(1), pp. 182–196, 1984.
[25] G. S. Semaan, J. A. M. Brito, I. M. Coelho, E. F. Silva, A. C. Fadel, L. S. Ochi, and N. Maculan, "A brief history of heuristics: from bounded rationality to intractability," *IEEE Latin America Transactions*, vol. 18, no. 11, pp. 1975–1986, 2020.
[26] H. Vinod, "Integer programming and theory of grouping," *Journal of American Statistical Association*, vol. 64, pp. 506–517, 1969.
[27] L. Wolsey, *Integer Programming*. Wiley, 2nd edition, 2020.
[28] M. J. van der Laan, K. S. Pollard, and J. Bryan, "A new partitioning around medoids algorithm," *Journal of Statistical Computation and Simulation*, vol. 73(8), pp. 575–584, 2003.
[29] Q. Zhang and I. Couloigner, "A new and efficient k-medoid algorithm for spatial clustering," *Lecture Notes in Computer Science*, vol. 342, pp. 181–189, 2005.
[30] S. C. Chu, J. F. Roddick, and J. S. Pan, "Improved search strategies and extensions to k-medoids-based clustering algorithms," *International Journal of Business Intelligence and Data Mining*, vol. 3(2), pp. 212–231, 2009.
[31] H. S. Park and C. H. Jun, "A simple and fast algorithm for k-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336–3341, 2009.
[32] M. Nascimento, F. Toledo, and A. de Carvalho, "Hybrid heuristc for the k-medoids clustering problem," *A Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference*, vol. 1, pp. 417–424, 2012.
[33] S. Zadegan, M. Mirzaie, and F. Sadoughi, "Ranked k-medoids: A fast and accurate rank-based partitioning algorithm for clustering large datasets," *Elsevier*, vol. 39, pp. 133–143, 2013.
[34] D. Yu, G. Liu, M. Guo, and X. Liu, "An improved k-medoids algorithm based on step increasing and optimizing," *Expert Systems with Applications*, vol. 92, pp. 464–47, 2018.
[35] H. Song and W. Lee, J.G.and Han, "Pamae: Parallel k-medoids clustering with high accuracy and efficiency," *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 1087–1096, 2017.
[36] E. Schubert and P. J. Rousseeuw, "Faster k-medoids clustering: Improving the pam, clara, and clarans algorithms," *Similarity Search and Applications - Lecture Notes in Computer Science*, vol. 11807, p. 171–187, 2019.
[37] A. Ushakov and I. Vasilyev, "A parallel heuristic for a k-medoids clustering problem with unfixed number of clusters," *MIPRO - International Convention on Information and Communication Technology, Electronics and Microelectronics,Proceedings*, pp. 1116–1120, 2019.
[38] Y. W. Wang, A. Jian, Y. Liang, and H. Wang, "Optimization of k-medoids algorithm for initial clustering center," in *IOP Conf. Series: Journal of Physics:*, vol. 1487, pp. 1–7, IOP Publishing, 2020.
[39] A. Ushakov and I. Vasilyev, "Near-optimal large-scale k-medoids clustering," *Information Sciences*, vol. 545, pp. 344–362, 2021.
[40] C. Lucasius, A. Dane, and G. Kateman, "On k-medoid clustering of large data sets with the aid of a genetic algorithm: background, feasibility and comparison," *Analytica Chimica Acta*, vol. 282, pp. 647–669, 1993.
[41] W. Sheng and X. Liu, "A hybrid algorithm of k-medoid clustering of large data sets," *Proceedings of the Congress on Evolutionary Computation – IEE*, pp. 77–82, 2004.
[42] W. Sheng and X. Liu, "A genetic k-medoids clustering algorithm," *Journal of Heuristics*, vol. 12, pp. 447–46, 2006.
[43] J. A. M. Brito, G. S. Semaan, and L. R. Brito, "Resolução do problema dos k-medoids via algoritmo genético de chaves aleatórias viciadas," *Revista Pesquisa Naval*, vol. 27, pp. 126–142, 2015.
[44] A. Hudaib, M. Khanafseh, and O. Surakhi, "An improved version of k-medoid algorithm using cro," *Modern Applied Science*, vol. 12, no. 2, pp. 116–12, 2018.
[45] R Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020.

José André de Moura Brito Tem bacharelado em Matemática pela Universidade Federal do Rio de Janeiro (1997), Mestrado (1999) e Doutorado (2004) em Engenharia de Sistemas e Computação (Otimização) pela COPPE/UFRJ e Pós-Doutorado em Otimização na Universidade Federal Fluminense (2008). Atualmente é professor da Escola Nacional de Ciências Estatísticas (ENCE/IBGE), onde leciona disciplinas na graduação.Tem experiência nas áreas de Otimização, Estatística e Computação.

Gustavo Silva Semaan é professor da Universidade Federal Fluminense (UFF) no Instituto do Noroeste Fluminense de Educação Superior (INFES) desde 2014. Pós-doutorado realizado no Laboratório de Inteligência Computacional (LabIC), no Instituto de Computação (IC) da UFF. Doutor em Computação (Algoritmos e Otimização) e Mestre em Computação (Otimização e Inteligência Artificial) pelo IC-UFF. Bacharel em Sistemas de Informação pela Faculdade Metodista Granbery.

Augusto César Fadel é bacharel em Estatística pela Escola Nacional de Ciências Estatísticas (ENCE) e tem mestrado em Ciência da Computação na UFF. Atua como estatístico no Instituto Brasileiro de Geografia e Estatística (IBGE), onde desenvolve atividades relacionadas a controle estatístico de sigilo e uso de big data na produção de estatísticas oficiais.