A Hybrid Heuristic with Hopkins Statistic for the Automatic Clustering Problem

G. Semaan, A. Fadel, J. Brito, and L. Ochi

***Abstract*—Cluster Analysis is a multivariate method to handle real problems associated with several fields. This area combines several methods of unsupervised classification, which can be applied in order to identify groups in a data set. The Clustering Problems are classified as NP-Hard and, in order to obtain such classification, the number of groups k may be fixed, or, in the Automatic approach, the ideal k must be identified upon evaluation of some validation index. In this paper the Silhouette Index was considered and a new proposed Hybrid Heuristic Algorithm (HHA) operates to identify the ideal number of groups. The HHA consider two heuristic algorithms based on metaheuristics: an algorithm based on Iterated Local Search (ILS) that considers a density-based approach and a literature Evolutionary Algorithm (EA). Besides, the HHA have a heuristic algorithm that verify clustering tendency, considering the Hopkins Statistic. Basically, according with the clustering tendency level, the HHA use a specific heuristic (ILS or EA). The computational experiments used three literature data sets with eighty-two instances, and all of them were considered and reported by different researchers. The effectiveness and the efficiency of the proposed heuristic are reflected in substantially lower computational time and in the solutions quality, that are competitive when compared with the best results reported in the literature.**

***Index Terms*—Heuristic, Metaheuristics, Iterated Local Search, Automatic Clustering Problem, Hopkins Statistics, Silhouette Index, density-based.**

I. Introdução

O problema clássico de agrupamento pode ser definido da seguinte maneira: dado um conjunto formado por n objetos $X=\{x_1,x_2,...,x_i,...,x_n\}$, com cada objeto $x_i \in X$ possuindo p atributos (dimensões ou características), ou seja, $x_i=\{x_i^1,x_i^2,...,x_i^p\}$, deve-se construir k grupos C_j (j=1,...,k) a partir de X, de maneira a garantir que os objetos de cada grupo sejam homogêneos segundo alguma medida de similaridade. Considerando estas observações, em um problema de agrupamento, uma solução (ou partição) pode ser representada como $\pi=\{C_1,C_2,...,C_k\}$. Além disso, devem ser respeitadas as restrições concernentes a cada problema particular abordado [1][2]: o conjunto X corresponde à união dos objetos dos grupos, cada objeto pertence a exatamente um grupo e todos os grupos possuem, pelo menos, um objeto. Além dessas restrições, define-se, também, uma função objetivo que permite avaliar a homogeneidade dos grupos formados [3]-[5]. Já o problema de agrupamento automático, que foi o objeto de estudo desse artigo, além das restrições supracitadas, o número de grupos k não é definido a priori.

O objetivo do presente trabalho é propor um novo método híbrido eficiente (quanto ao tempo computacional) e eficaz (quanto à qualidade das soluções) para a resolução de um problema de alta complexidade na área de Otimização, qual seja, o Problema de Agrupamento Automático. O método considera, em sua fase inicial, a utilização da Estatística de Hopkins, sendo essa estatística baseada na aplicação de um Teste de Aleatoriedade Espacial. Tal teste possibilita verificar a tendência à formação de agrupamentos para uma dada instância do problema. Ainda neste sentido, conforme o resultado do referido teste, uma heurística específica será aplicada à instância. Em relação às heurísticas utilizadas, foram consideradas aquelas com os melhores resultados obtidos em experimentos preliminares. O Algoritmo Evolutivo com Busca Local AECBL1[6] se destacou em relação às instâncias que não possuem tendência a formação de agrupamentos. Já a heurística ILS-DBSCAN, baseada na metaheurística Busca Local Iterada (do inglês *Iterated Local Search*), que usa como procedimento de construção de soluções iniciais o clássico algoritmo baseado em densidade DBSCAN (*Density-based spatial clustering of applications with noise*), obteve os melhores resultados para as instâncias com tendência a formação de agrupamentos.

Esse trabalho está dividido em cinco seções, incluindo a introdução. A seção 2 apresenta uma revisão da literatura sobre Problema de Agrupamento Automático e o índice relativo silhueta. A seção 3 apresenta algoritmos heurísticos para o problema abordado, as instâncias consideradas e alguns experimentos computacionais preliminares. A seção 4 apresenta o método proposto, experimentos computacionais e uma análise de tendência para formação de grupos. Por fim, a seção 5 apresenta as conclusões e possíveis desdobramentos.

II. Revisão da Literatura

Conforme [2], talvez um dos problemas de seleção de parâmetros mais conhecido seja o de determinar o número ideal de grupos em um problema de agrupamento. Neste sentido, diversas técnicas não supervisionadas de avaliação de soluções podem ser utilizadas.

G.S. Semaan, Instituto do Noroeste Fluminense de Educação Superior (INFES) da Universidade Federal Fluminense (UFF), Santo Antônio de Pádua, RJ, Brasil (gustavosemaan@id.uff.br).

A.C. Fadel, Instituto Brasileiro de Geografia e Estatística (IBGE), Rio de Janeiro, RJ, Brasil (augusto.fadel@ibge.gov.br).

J.A.M. Brito, Escola Nacional de Ciências Estatísticas (ENCE) do Instituto Brasileiro de Geografia e Estatística (IBGE), Rio de Janeiro, RJ, Brasil (jose.m.brito@ibge.gov.br).

L. S. Ochi, Instituto de Computação (IC) da Universidade Federal Fluminense (UFF), Niterói, RJ, Brasil (satoru@ic.uff.br).

Com base no algoritmo *k-Means*, que utiliza a ideia de protótipos, foi proposto em [7] o algoritmo *X-Means* para a resolução do problema de agrupamento automático. Esse algoritmo tem como parâmetros de entrada a instância a ser processada e um intervalo com a quantidade de grupos [k_{min}, k_{max}]. A partir destes dados, o algoritmo utiliza o índice BIC (*Bayesian Information Criterion*) para identificar e retornar qual o melhor número de grupos. Em [8] é apresentado um algoritmo que também adapta o *k-Means* para resolver um problema de agrupamento automático.

Ainda no que diz respeito ao problema agrupamento automático, vários trabalhos na literatura propõem algoritmos baseados em metaheurísticas que têm por objetivo encontrar um número ideal de grupos e a sua solução (partição) correspondente. Dentre estes, pode-se destacar os seguintes trabalhos: [6] [9]-[15][18].

Existem, também, as heurísticas que utilizam alguns procedimentos de busca local baseados no algoritmo *k-Means*. Em uma primeira etapa, essas heurísticas utilizam algoritmos para construção de grupos, denominados grupos parciais (temporários, componentes conexos) com o objetivo de unir os objetos mais homogêneos. Em seguida, são aplicados procedimentos de busca local e de perturbação sobre esses grupos produzindo soluções de boa qualidade, ou seja, os grupos parciais são unidos e formam grupos finais [6] [10] [11] [14].

Em [18] foi proposto um *Método de Classificação Baseado em Densidade*, que utiliza o conhecido algoritmo DBSCAN [2] para produzir soluções para o problema de agrupamento automático. Esse algoritmo foi adaptado para que os objetos identificados como *outliers* não fossem ignorados.

Os algoritmos de agrupamento baseados em densidade têm como objetivo a determinação de grupos (regiões) de alta densidade de objetos separados por regiões de baixa densidade. Nesse contexto, as soluções do conjunto base, que serão submetidas ao método proposto no presente trabalho, foram obtidas com a aplicação do algoritmo DBSCAN, baseado em densidade, apresentado em [19].

Em [33] é proposta uma metaheurística de inteligência coletiva inspirada no comportamento das formigas para a resolução do PAA. Foi utilizado o índice silhueta para avaliação das soluções obtidas e o algoritmo foi paralelizado.

Algoritmos genéticos que consideram conceitos de densidade e ruído são propostos por [34], com o objetivo de identificar o número ideal de grupos e, assim, definir os melhores pontos para origens/destinos com base em rotas de taxis e dados de GPS. Nos algoritmos propostos foram consideradas variantes do algoritmo k-means para produzir a população inicial (conjunto solução da 1ª geração).

O trabalho [35] aborda o problema de agrupamento automático, considerando a divisão de instâncias em grupos. O algoritmo proposto baseia-se em uma colônia de abelhas artificiais, e técnicas de agrupamento apoiam a exploração dessas abelhas, direcionando seus movimentos para a resolução do Problema de Segmentação de Clientes.

A busca por organismos simbióticos (SOS-symbiotic organism search) é uma metaheurística recente que simula estratégias de interação simbiótica adotadas pelos organismos para sobreviver e se propagar em um ecossistema.

Um SOS para resolver problema de agrupamento é proposto por [36], e utiliza conjuntos de dados bem conhecidos da literatura. O algoritmo apresentou um alto nível de estabilidade.

A. O Índice Silhueta

Dentre os muitos índices disponíveis associados ao critério relativo de validação, o índice silhueta, proposto por [20], é um dos mais utilizados para validação de agrupamentos. Esse índice possui propriedades desejáveis, quais sejam, combina as ideias de coesão e de separação. Ainda neste sentido, a sua utilização permite avaliar a qualidade das soluções com base na proximidade entre os objetos de determinado grupo e na distância desses objetos ao seu grupo mais próximo.

Esse índice é calculado para cada objeto, sendo possível identificar se o mesmo está alocado ao grupo mais adequado. Os passos a seguir explicam, brevemente, como calculá-lo:

(1) Considerando que d_{ij} corresponde à distância euclidiana entre dois objetos x_i *e* x_j, calcula-se , para cada objeto x_i , a sua distância média $a(x_i)$ (Equação (1)) em relação a todos os demais objetos do mesmo grupo. Nessa equação, $|C_w|$ representa a quantidade de objetos do grupo C_w , ao qual o objeto x_i pertence.

$$a(x_i) = \frac{1}{|C_w| - 1} \sum d_{ij} \quad \forall x_j \neq x_i, \quad x_j \in C_w \tag{1}$$

(2) A Equação (2) apresenta a distância entre o objeto x_i e os objetos do grupo C_t, em que $|C_t|$ é a quantidade de objetos do grupo C_t. Para cada objeto x_i calcula-se a sua distância média em relação a todos os objetos dos demais grupos ($b(x_i)$) (Equação (3)).

$$d(x_i, C_t) = \frac{1}{|C_t|} \sum d_{ij} \quad \forall x_j \in C_t \tag{2}$$

$$b(x_i) = \min d(x_i, C_t) \quad C_t \neq C_w \tag{3}$$

(3) O índice silhueta do objeto x_i ($s(x_i)$) pode ser obtido pela equação 4.

$$s(x_i) = \frac{b(x_i) - a(x_i)}{\max\{b(x_i), a(x_i)\}} \tag{4}$$

$$\max Silhueta(S) = \frac{1}{n} \sum_{i=1}^{n} s(x_i) \tag{5}$$

(4) O cálculo da silhueta de uma solução S (partição π) é a média das silhuetas de cada objeto, conforme definido na Equação (5) acima, sendo n correspondente à quantidade de objetos da solução. Essa função deve ser maximizada. Valores de $s(x_i)$ positivos e mais próximos de 1 indicam que o objeto está bem localizado em seu grupo, enquanto valores negativos indicam que o objeto deveria ser alocado a outro grupo.

III. Algoritmos para o Problema de Agrupamento Automático

Os algoritmos apresentados nesta seção possuem algumas características similares ou em comum no que concerne à estrutura de dados e aos seus procedimentos. Eles incorporam um procedimento para a construção de *Grupos Parciais*, considerando conceitos de densidade, em duas etapas: *Formação* e *União (ou Junção) de Grupos Parciais*. Justifica-se a construção de tais grupos como um procedimento de pré-processamento [6], a fim de reduzir a cardinalidade da instância, ou seja, o número de objetos.

No algoritmo *CLUSTERING* [13], *o* conceito de *Grupos Parciais* diz respeito aos componentes conexos de *Grafos* construídos com base na formação do *Grafo de Vizinhança G*. Os algoritmos *AEC* e *SAPCA* [10] adotam, também, o conceito baseado em *Grafos*, utilizando um *Grafo de Vizinhança Relativa*. No algoritmo *CLUES* [21] o procedimento responsável pela formação dos *Grupos Parciais* é denominado *Encolhimento*, sendo considerados, nesse caso, os *k*-vizinhos mais próximos a cada objeto.

A. Algoritmos Heurísticos

Esta subseção traz uma breve descrição da heurística baseada na *metaheurística Busca Local Iterada* (ILS, do inglês *Iterated Local Search*). Em especial, foram considerados procedimentos utilizados pelo eficiente Algoritmo Evolutivo proposto por [6]. Destaca-se que os algoritmos atuam na maximização do *Índice Silhueta Tradicional*. O objetivo é construir soluções com *Grupos Parciais* de boa qualidade e, ainda, refiná-las, em busca de uma solução *ótima global* ou um *ótimo local* de excelente qualidade. Algumas características dos algoritmos propostos são descritas a seguir:

1) Procedimento de Construção: ambos os procedimentos construtivos utilizados consideram conceitos de agrupamentos baseados em densidade. São eles:

1.1) Uma variante do algoritmo DBSCAN [19] em que todos os objetos devem pertencer a um grupo, inclusive os objetos classificados como *ruídos*.

1.2) FJGP [3]: procedimento para construção de soluções iniciais utilizado pelo algoritmo AECBL1. Em uma primeira etapa esse procedimento de construção forma *m* grupos parciais ($m \leq n$ objetos), denominados grupos iniciais. Um vetor binário *auxB* com *m* posições é utilizado e, de maneira aleatória, um valor zero ou um é atribuído a cada grupo parcial. O *bit* 1 indica que um grupo parcial será considerado do tipo "*pai*" e o *bit* 0 indica que o grupo parcial será do tipo "*filho*". A segunda etapa do procedimento de construção realiza junções entre os grupos parciais. Nesse sentido, cada grupo do tipo "filho" deve se unir ao grupo "pai" mais próximo e, para isso, as distâncias dos centroides dos grupos parciais devem ser consideradas. Após a execução do procedimento, a solução inicial obtida irá possuir *k´* grupos, quantidade de grupos "pais" do vetor *auxB*.

2) Buscas Locais: todas as buscas locais relatadas a seguir atuam na manipulação do vetor *auxB*.

2.1) Inversão Individual: a cada posição *i* do vetor ($auxB_i$), deve-se inverter o seu valor e aplicar o procedimento de *Formação de Soluções*. No presente contexto, inverter o seu valor implica alterar a classificação de um objeto de "*filho*" para "*pai*" e vice-versa, ou seja, a quantidade de grupos da solução formada será *k´ + 1* ou *k´ - 1*, sendo *k´* a quantidade de *Grupos Parciais "Pais"* antes da inversão. É importante destacar, que a cada operação de inversão, a função de avaliação (cálculo do *Índice Silhueta*) é aplicada e uma nova solução é armazenada somente se for melhor que a melhor solução armazenada até o momento.

2.2) Troca entre Pares: Esse procedimento realiza troca entre os valores dos *bits* das posições *i* e *j* apenas se $\text{auxB}_i \neq \text{auxB}_j$. Tratar-se de uma busca intensiva pois, todas as combinações de i e j são avaliadas e, após a realização da troca, a quantidade de grupos da solução permanece inalterada (k + 1 - 1 = k).

2.3) Inversão Individual com Sentido Aleatório: em experimentos preliminares com as *Buscas Locais Troca entre Pares* observou-se que o tempo consumido em relação ao benefício alcançado (maximização da silhueta) não foi satisfatório. A busca local *Inversão Individual* percorre o vetor binário *auxB* de maneira sequencial realizando a inversão dos valores ("*filho*" para "*pai*" e vice-versa). Com o objetivo de percorrer novos caminhos de busca, um novo procedimento foi proposto apenas tornando aleatória a seleção da posição *i* do vetor *auxB* a ser alterada. Nesse sentido, assim como a versão da busca local proposta, todas as posições têm seus valores invertidos, porém em ordem aleatória. Embora seja pequena a alteração, os resultados obtidos em experimentos preliminares indicaram que as heurísticas propostas no presente trabalho devem utilizar apenas a nova busca local *Inversão Individual Sentido Aleatório.*

2.4) Perturbação: inversão aleatória de *bits* do vetor auxB e aplicação do procedimento de formação de soluções.

2.5) Filtro: são produzidas *f* soluções com a utilização dos procedimentos de construção, em que *f* é submetido como parâmetro. Cada uma dessas soluções é refinada pela *Busca Local Inversão Individual Aleatória*. Em seguida, os *Grupos Iniciais* da melhor solução obtida são submetidos aos procedimentos de *Busca Local* e de *Perturbação* para um refinamento mais intensivo.

2.6) União de Grupos Parciais: recálculo de centroides com redução do custo computacional de $O(n.p)$ para $O(p)$, em que *n* é a quantidade de objetos e *p* e a quantidade de atributos.

3) Critérios de Parada: o algoritmo é executado até a obtenção de uma solução com *índice Silhueta* maior ou igual ao valor submetido como parâmetro. Existe, ainda, um tempo máximo de execução, também submetido como parâmetro (opcional).

Com base nas características relatadas, são apresentados dois algoritmos heurísticos baseados na metaheurística ILS. Esses algoritmos diferenciam-se, apenas, quanto aos procedimentos de construção. São eles: o ILS-FJGP e o ILS-DBSCAN.

B. Estatística de Hopkins

O *Teste de Aleatoriedade Espacial* possibilita verificar se um dado conjunto *X* (instância) com *n* objetos $X = \{x_1, x_2, ..., x_n\}$, em um espaço *p*-dimensional, $x_i = \{ x_i^1, x_i^2, ..., x_i^p \}$ [22] [1] tem uma tendência à formação de agrupamentos. Neste sentido, um teste de hipóteses pode ser conduzido, adotando como hipótese nula a hipótese de homogeneidade, ou seja, de

que os objetos são distribuídos de maneira uniforme no espaço e que, portanto, não há estrutura de grupos. A hipótese de não homogeneidade é caracterizada como hipótese alternativa. Caso a estatística de teste indique a rejeição da hipótese nula, conclui-se que não há evidência de que os objetos da instância sejam distribuídos homogeneamente, sugerindo, portanto, a presença de grupos.

No que diz respeito à estatística de teste, uma opção conveniente é a Estatística de Hopkins (EH), que utiliza um critério interno em que nenhuma informação a priori é necessária para a realização das análises. Em [1] a regra de decisão adotada não rejeita a hipótese nula quando o valor obtido para EH for igual ou menor do que 0,5. Um estudo realizado por [23] apontou que tal teste apresenta maior probabilidade de rejeitar a hipótese nula, quando a hipótese alternativa é verdadeira, do que diversos testes bem conhecidos de aleatoriedade espacial.

Além do conjunto *X*, dois outros conjuntos são considerados no cálculo da EH, sejam eles: ***X**** corresponde a uma amostra do conjunto *X* ($X^* \subset X$) com *m* objetos que são selecionados de maneira aleatória; ***A*** possui *m* objetos construídos artificialmente, segundo uma distribuição uniforme, no espaço de cada uma das *p*-dimensões.

Após a definição dos conjuntos de objetos utilizados, devem ser estabelecidas as distâncias utilizadas: w_j é a distância entre um objeto $x^* \in X^*$ até o objeto de $X - \{x^*\}$ mais próximo e u_j é a distância entre um objeto $a \in A$ até o objeto mais próximo em X.

Na Equação (6) busca-se a maximização de *H*, cujo valor pertence ao intervalo [0,1]. Conforme [22], há três classes em que a instância pode ser classificada, considerando a distribuição dos seus objetos:

Classe em que os objetos são regularmente espaçados: instância sem tendência a formação de agrupamentos. Em resultados da literatura, para essa classe, o valor de *H* variou no intervalo (0,0.3].

Classe com objetos distribuídos de maneira aleatória no espaço: indica que o conjunto de objetos não têm uma estrutura propícia para o agrupamento (*H* próximo a 0.5).

Classe com objetos com uma tendência a formação de agrupamentos: existem grupos bem definidos. Em resultados da literatura, para essa classe, o valor de *H* variou no intervalo [0.7,1).

$$\max H = \frac{\sum_{j=1}^{m} u_j}{\sum_{j=1}^{m} u_j + \sum_{j=1}^{m} w_j} \qquad (6)$$

O Algoritmo 1 apresenta a heurística para verificar a tendência *para formação de agrupamentos.*

C. Experimentos Computacionais Preliminares

As implementações dos algoritmos propostos foram feitas em Linguagem C++. É uma prática comum, para a identificação da quantidade ideal de grupos em problemas de agrupamento automático, utilizar k=2,...,k_{max} , sendo $k_{max} = n^{1/2}$ [24][25]. Em [1], entretanto, um método simples para a estimativa do número ideal de grupos consiste em utilizar valores inteiros de k próximos a $\sqrt{n/2}$, na expectativa que cada grupo possua cerca de $\sqrt{2n}$ objetos. Com o objetivo de contemplar ambos os intervalos apresentados na literatura, neste trabalho foi considerado $k=2,...,n^{1/2}$.

Algoritmo 1: Algoritmo da Estatística de Hopkins

Parâmetros: numero_iteracoes, conjunto X e m

1 i=0;

2 **Enquanto** (i < numero_iteracoes) **Faça**

3 Selecionar m objetos a partir de X, definindo subconjunto X^*

4 Gerar artificialmente um subconjunto A com m objetos

5 Calcular w_i=min d(x_i,x_j), tal que $x_i \in X^*$ e $\forall x_j \in X$ ($x_i \neq x_j$)

6 Calcular u_i=min d(a_i,x_j), tal que $a_i \in A$ e $\forall x_j \in X$ ($x_i \neq x_j$)

7 $H_i = \sum_{j=1}^{m} u_j / \left(\sum_{j=1}^{m} u_j + \sum_{j=1}^{m} w_j \right)$

8 i=i+1

9 **Fim-Enquanto**

10 Retornar a mediana de H

Para a realização dos experimentos foram utilizadas *82* instâncias da literatura que estão distribuídas em três conjuntos (*DS - Datasets*). Essas instâncias possuem quantidades de objetos entre *30* e *2000*, a quantidade de dimensões (atributos) entre *2* e *13* e diferentes características relacionadas, por exemplo, com a coesão, à separação, formatos e às densidades dos grupos. Todas as instâncias utilizadas no trabalho estão disponíveis em http://labic.ic.uff.br/Instance.

O primeiro conjunto (DS1) contempla nove instâncias conhecidas da literatura com a quantidade de objetos entre 75 e 1484 e dimensões (quantidade de atributos) entre 2 e 13 [26] -[30].

O segundo conjunto (DS2) contempla 51 instâncias que foram construídas por [6]. Essas instâncias possuem quantidades de objetos entre 100 e 2000, sendo todas com duas dimensões. Nesse conjunto os nomes das instâncias foram definidos de acordo com a quantidade de objetos, de grupos, e se os grupos são bem definidos, coesos e separados (denominados *"comportadas"* e "*não comportadas*" em [6]). A Fig. 1(a) apresenta a instância *100p7c* (nome com final "*c*") considerada "comportada" com 100 objetos e 7 grupos, enquanto a Fig. 1(b) indica uma instância "*não comportada*" com *112* objetos (*100p7c* com final " *c1*").

Por fim, o terceiro conjunto (DS3) contempla *22* instâncias que foram construídas e utilizadas por [9]. Essas instâncias possuem a quantidade de objetos entre *30* e *2000* e duas dimensões.

D. Experimentos com Heurísticas ILS-FJGP e ILS-DBSCAN

O primeiro experimento consistiu em executar o algoritmo que obteve os melhores resultados em relação ao tempo de processamento e maximização do índice silhueta, dentre os apresentados na revisão da literatura. Trata-se do algoritmo

evolutivo [32] com busca local (AECBL1) apresentado em [6] ou [15].

O AECBL1 foi executado 5 vezes, cada uma com 50 iterações, sendo armazenados os seguintes parâmetros de saída: ***k***: número de grupos identificado como *ideal*; **Silhueta**: valor da maior silhueta obtida; **Tempo Total**: o tempo total (em segundos) para a execução das 50 iterações entre as execuções; **Tempo Iteração**: menor tempo de execução (em segundos) em que o algoritmo alcançou o seu melhor resultado (maior silhueta). É importante ressaltar que a melhor solução pode não ter sido obtida em todas as execuções.

O próximo experimento consistiu em executar os Algoritmos baseado em Busca Local Iterada [32] ILS-FJGP e ILS-DBSCAN utilizando os resultados obtidos no experimento anterior, realizado com o AECBL1. Os critérios de parada utilizados foram: alcançar o maior valor de silhueta obtido pelo AECBL1 ou processar até atingir o Tempo Total utilizado pelo AECBL1. A proposta do ILS-FJGP consiste em realizar mais construções de soluções iniciais, utilizando o mesmo procedimento construtor do AECBL1, a *Formação e Junção de Grupos Parciais*.

Com base nos resultados do ILS-FJGP, em apenas 6% das instâncias a diferença em relação à melhor silhueta do AECBL1 foi inferior a 0,01. Os resultados obtidos indicam que as soluções iniciais obtidas pelo procedimento FJGP necessitam de mais aprimoramento. Nesse sentido, as gerações do Algoritmo Evolutivo AECBL1 utilizam-se de buscas locais e de um procedimento de reconexão por caminhos, além dos operadores genéticos (mutações, cruzamentos e seleções de indivíduos). As iterações desse algoritmo atuam em um refinamento mais intensivo nas soluções iniciais.

Em experimentos anteriores realizados com o MRDBSCAN [17], observou-se que para as instâncias denominadas *comportadas* o método produziu resultados de boa qualidade, em que a diferença média entre a melhor silhueta e a silhueta obtida foi de 0,1 e a mediana das diferenças foi 0. Além disso, em todos os experimentos relacionados a essas instâncias, quando a quantidade de grupos não foi a mesma do melhor resultado existente, a diferença máxima foi de apenas duas unidades em relação ao número de grupos.

O ILS-DBSCAN, assim como o método sistemático MRDBSCAN, utiliza como procedimento de construção um algoritmo baseado no DBSCAN. A heurística ILS-DBSCAN, entretanto, não utiliza a técnica *DistK* [2] para calibrar seus parâmetros. Nesse sentido, o parâmetro raioDBSCAN é obtido com a multiplicação de d_{Media} (média das menores distâncias entre cada objeto *i* e outro objeto j, $i \neq j$) com a variável *z* (valor fracionário aleatório no intervalo [1.5, 4.5]). Já o parâmetro *qtdeObjetos* corresponde a um valor inteiro selecionado aleatoriamente do conjunto {2,3,4,5}.

Os valores dos parâmetros *raioDBSCAN* e *qtdeObjetos* foram calibrados em experimentos preliminares, e a motivação para utilizá-los, considerando fatores aleatórios, foi a obtenção de configurações diversificadas, diferentes das utilizadas nas 28 versões do MRDBSCAN devido ao uso da técnica *DistK*.

Conforme foi apresentado nos resultados do ILS-FJGP, em apenas 6% das instâncias a diferença em relação à melhor silhueta do AECBL1 foi inferior a 0,01. Quanto aos comparativos dos resultados dos algoritmos AECBL1 e ILS-DBSCAN, o percentual aumenta para 58%. Esse resultado sugere uma análise com o objetivo de identificar características comuns às instâncias em que a heurística proposta alcançou resultados de boa qualidade com reduzido custo computacional. Para isso, a tabela I apresenta percentuais em que os algoritmos alcançaram as melhores soluções. É possível observar que o algoritmo AECBL1 destacou-se produzindo o melhor resultado em mais de 96% das instâncias utilizadas, enquanto o ILS-DBSCAN alcançou o melhor resultado para cerca de 50%. Porém, ao considerar apenas as instâncias *comportadas*, o ILS-DBSCAN alcançou os melhores resultados em todos os casos, enquanto o AECBL1 obteve em cerca de 89% das instâncias.

TABELA I
ALCANCE À MELHOR SOLUÇÃO OBTIDA POR ALGORITMO

Instância	AECBL1	ILS-FJGP	ILS-DBSCAN
Todas	96,3%	4,9%	50,6%
DS2	96,0%	4,0%	46,0%
DS2 – Comportadas	88,9%	11,1%	100,0%

Com o objetivo de analisar também os custos computacionais (tempos de processamento) do algoritmo ILS-DBSCAN para as instâncias comportadas, a tabela II apresenta: o ***Gap total***, razão entre o tempo total de processamento do AECBL1 e do ILS-DBSCAN e o ***Gap Iteração***, razão entre o tempo em que foi obtida a melhor solução do AECBL1 e do ILS-DBSCAN.

Em relação ao ***Gap Iteração***, o ILS-DBSCAN consumiu, em média, menos de 0,9% do tempo consumido pelo AECBL1. No pior caso, o tempo computacional do algoritmo foi de apenas cerca de 3%. Quando considerado o ***Gap Total***, a diferença é ainda mais significativa. Em termos de tempo de processamento total, o consumo máximo do ILS-DBSCAN foi aproximadamente 0,6% do tempo total utilizado pelo AECBL1.

TABELA II
TEMPO DE EXECUÇÃO PARA INSTÂNCIAS COMPORTADAS DO ILS-DBSCAN EM RELAÇÃO AO AECBL1

Estatísticas	Gap	
	Total	Iteração
Menor	0,00%	0,02%
Média	0,11%	0,87%
Mediana	0,01%	0,08%
Maior	0,57%	3,08%

Conforme apresentado nas tabelas I e II, os resultados associados ao algoritmo ILS-DBSCAN foram, em geral, melhores ou iguais aos resultados do melhor algoritmo da literatura para as instâncias *comportadas*. Além disso, no pior caso, o algoritmo precisou apenas de cerca de 3% do tempo consumido pelo AECBL1.

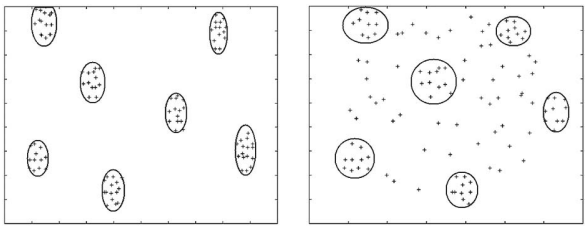
Com o objetivo de verificar a eficiência do ILS-DBSCAN**,** foi realizado um experimento baseado na *Análise de Probabilidade Empírica* proposta em [31]. Nesse experimento

foram utilizadas 4 instâncias consideradas *comportadas* com diferentes quantidades de objetos (entre 100 e 2000). Para cada instância o algoritmo foi executado 100 vezes. O critério de parada utilizado foi o alcance do alvo difícil, ou seja, o maior valor de silhueta obtida no experimento com o algoritmo AECBL1 (com apenas duas casas decimais).

E. Heurística para Estatística de Hopkins e Experimentos

Como foi apresentado anteriormente, a EH utiliza uma amostra do conjunto de objetos da instância ($X^* \subset X$) e um conjunto de objetos artificiais A, cujo os atributos possuem valores aleatórios no espaço de cada uma das p-dimensões. Uma vez que fatores aleatórios foram considerados (tanto em X^* quanto em A), tornou-se necessário o desenvolvimento de um algoritmo que realizasse diferentes iterações com o objetivo de obter estatísticas para os valores de H. A cada iteração o algoritmo seleciona objetos para a formação do conjunto X^* e constrói objetos artificiais para o conjunto A.

A Fig. 1(a) e a Fig. 1(b) apresenta duas instâncias e as medianas dos valores de H para a instância 100p7c, em que é possível identificar 7 grupos bem definidos (coesos e bem separados), com 100 objetos e para a instância *100p7c1* classificada como "*não comportada*" que possui 112 objetos. Nesse exemplo o algoritmo foi executado com 1000 iterações.



(a) instância 100p7c (H=0,831) (b) instância 100p7c1 (H=0,516)

Fig. 1. Instâncias "comportada" (100p7c) e "não comportada" (100p7c1).

No experimento da presente subseção foram consideradas 51 instâncias propostas e utilizadas por [6], que possuem entre 100 e 2000 objetos. Além disso, 63% das instâncias utilizadas possuem grupos bem definidos, denominadas "*comportadas*", e as demais 37% são as instâncias denominadas "*não comportadas*", conforme a classificação indicada por [6]. É importante ressaltar que a classificação proposta pelo autor foi realizada em uma análise visual, durante a construção das instâncias. Dessa forma, a hipótese é que uma instância classificada como *comportada* também seja classificada como instância com *tendência à formação de agrupamentos*.

Em experimentos preliminares foram utilizados conjuntos de amostras (X^*) com tamanhos 1%, 3%, 5%, 10% e 15% em relação à quantidade de objetos da instância (n). Além disso, foram consideradas as seguintes quantidades de iterações: 10, 100, 500 e 1000. Após a análise dos resultados obtidos nos experimentos preliminares, foi selecionada uma configuração em que o tamanho da amostra é de 1% e o algoritmo realiza 10 iterações. Essa escolha é decorrente do fato de compatibilidade entre os resultados concernentes ao valor de H e, além disso, do reduzido custo computacional necessário. A tabela III apresenta as estatísticas dos resultados obtidos com a configuração selecionada considerando, separadamente, apenas as instâncias *comportadas* e as instâncias *não comportadas*.

Com base na tabela III, a EH identificou a *Tendência de Agrupamentos* em todas as instâncias consideradas *comportadas*, em que a média e a mediana foram 0,9. Em relação ao conjunto de instâncias *não comportadas,* a média e a mediana foram inferiores a 0,7. É importante ressaltar que os valores extremos (menor e maior), embora sejam apresentados na tabela, não são considerados na análise. A justificativa para isto é que, eventualmente, configurações dos objetos do conjunto de amostra e dos objetos artificiais podem resultar em falso positivo, ou seja, indicar tendência à formação onde não existe. Por exemplo, em um dos resultados para as instâncias não comportadas o resultado de H foi 0,95. É importante ressaltar que uma instância classificada como *não comportada* não necessariamente corresponde a uma instância *sem tendência à formação de agrupamentos*.

TABELA III
RESULTADOS DA ESTATÍSTICA DE HOPKINS COM 1% DE AMOSTRA E 10 ITERAÇÕES

Estatística de Hopkins (H)			
Medidas	**Não Comportadas**	**Comportadas**	**Tempo (s)**
Maior	0,95	0,96	0,03
Menor	0,45	0,77	0,00
Média	0,66	0,90	0,01
Mediana	0,64	0,90	0,00

A tabela IV sumariza os tempos de processamento considerando 1000 execuções para amostras de 3%, 5%, 10% e 15% e ainda, para amostras de 1% e número de execuções igual a 100, 500 e 1000. Destaca-se que a configuração associada à amostra de 1% e 100 execuções foi suficiente para identificar instâncias com tendência à formação de agrupamentos, conforme apresenta a tabela IV.

Com o objetivo de analisar a distribuição dos resultados obtidos com a heurística para a EH, a Fig. 2 apresenta gráficos de caixa (*Boxplots*) para os experimentos com amostra de 1% e com respectivamente 100 execuções. São apresentados seis gráficos, em que foram discriminadas as instâncias por classificação (*comportadas* e *não comportadas*). Os gráficos foram construídos considerando os valores do primeiro Quartil (Quartil 1), mediana e terceiro Quartil (Quartil 3) entre os valores de H obtidos para cada instância nas 100 amostras. O Boxplot Quartil 1, por exemplo, relaciona apenas os valores do primeiro quartil nas 100 amostras de 1%, ou seja, apresenta a distribuição dos valores do primeiro quartil de H obtidos em cada uma das 100 amostras, para cada uma das instâncias envolvidas no experimento (32 instâncias *comportadas* e 19 instâncias *não comportadas*).

TABELA IV
ESTATÍSTICAS GERAIS DE TEMPOS DE EXECUÇÃO (SEGUNDOS) DOS EXPERIMENTOS COM A HEURÍSTICA PARA ESTATÍSTICA DE HOPKINS

	1000 execuções				**Amostra de 1%**		
Medidas	**Tamanho da amostra**				**Quantidade de amostras**		
	3%	**5%**	**10%**	**15%**	**100**	**500**	**1000**
Maior	4,30	9,41	13,74	28,25	0,13	0,66	1,37
Menor	0,01	0,02	0,04	0,05	0,00	0,00	0,00
Média	0,82	1,35	2,52	4,06	0,02	0,12	0,24
Mediana	0,26	0,43	0,88	1,28	0,01	0,04	0,08

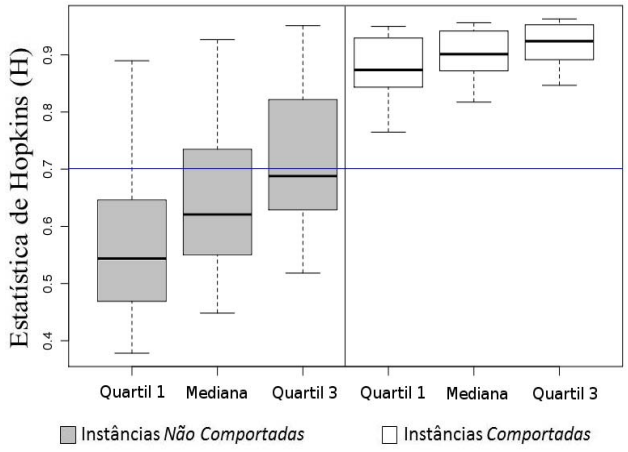


Fig. 2. Gráficos *boxplot* para o experimento da EH para 1% de amostra e 100 amostras.

Ainda com base na Fig. 2 é possível observar todos os valores de *H* relacionados no gráfico para as instâncias *comportadas* foram superiores a 0,75. Os resultados apresentados na Fig. 2, bem como os resultados relatados na tabela IV, indicam que as instâncias *comportadas* consideradas possuem tendência à formação de agrupamentos.

IV. Método Proposto

A seção anterior apresentou os resultados da heurística baseada na *Estatística de Hopkins*. A partir dos resultados reportados nessa seção é possível observar que, de forma eficiente, o algoritmo foi capaz de identificar instâncias em que existe *Tendência à formação de Agrupamentos*. Foi verificado, também, que todas as instâncias que não possuem tendência à formação de agrupamentos foram classificadas como *não comportadas* por [6]. Entretanto, seis instâncias classificadas como *não comportadas* possuem tendência à formação de agrupamentos. Esses resultados foram obtidos, inclusive, considerando o menor tamanho de amostra utilizado nos experimentos (1% e 10 execuções da heurística).

Na seção III foram apresentados resultados obtidos com as heurísticas baseadas em ILS propostas, sejam elas: ILS-FJGP e ILS-DBSCAN. A heurística ILS-DBSCAN destacou-se, produzindo resultados melhores ou equivalentes aos obtidos com o algoritmo da literatura AECBL1 e com tempos de processamento substancialmente inferiores.

Com base nos resultados obtidos pela *Heurística para a Estatística de Hopkins*, ILS-DBSCAN e com o algoritmo da literatura AECBL1, o presente trabalho propõe um método que adota uma heurística que utiliza a *Estatística de Hopkins* para direcionar qual é a heurística mais adequada à resolução da instância submetida.

Experimentos computacionais foram conduzidos a fim de analisar a tendência à formação de grupos e apresentar os resultados do método proposto. No que diz respeito à tendência de agrupamento, os valores da estatística de Hopkins, obtidos previamente à obtenção de grupos, foram comparados com os valores do índice silhueta, obtidos após a aplicação do algoritmo de agrupamento. Para o método proposto, os experimentos apresentam seu desempenho computacional em relação ao algoritmo da literatura AECBL1.

A. Análise de Tendência à Formação de Grupos

Com base nos experimentos relatados da *Heurística para EH*, um caminho natural é analisar os valores de silhuetas e de *H* (Hopkins) também em relação ao significado das soluções obtidas. Nesse sentido, a tabela V traz os resultados obtidos para quatro instâncias. Com base nos resultados apresentados nessa tabela, para as instâncias *comportadas*, é possível observar que os valores de *H* foram altos, superiores *a 0,84*. Esses resultados indicam a existência de uma tendência à formação de agrupamentos, ou seja, que os objetos não estão *distribuídos uniformemente* e nem estão *dispersos* no espaço. Os resultados indicam, também, que a realização do agrupamento de dados, independente da técnica e/ou algoritmo utilizados, pode resultar em soluções que possuem algum significado real, e não apenas grupos artificiais que foram obtidos com a maximização ou minimização de uma função objetivo.

TABELA V
Estatísticas Dos Valores De H* Obtidos Com 100 Execuções De 1%.

Tipo Instância	Instância	AECBL1		
		k	Silhueta	H*
Comportadas	100p7c	7	0,834	0,843
	1300p17c	17	0,823	0,949
Não comportadas	100p7c1	7	0,491	0,571
	1000p27c1	28	0,523	0,546

Ainda no que concerne às instâncias *comportadas*, as silhuetas obtidas foram altas, superiores a *0,82*. Além disso, foram identificadas as mesmas quantidades de grupos em três situações, sejam elas: (i) Na construção das instâncias, conforme, inclusive, a nomenclatura utilizada [6]; (ii) Com a utilização das heurísticas que alcançaram o valor da maior silhueta para cada instância; (iii) Com a utilização do algoritmo DBSCAN em experimentos preliminares.

Embora o DBSCAN seja capaz de identificar e ignorar objetos *outliers*, no problema abordado no presente trabalho todos os objetos precisam estar associados a um grupo. Dessa forma**,** cada objeto *ruído*deve formar um grupo *singleton* ou deve ser alocado a um grupo de maneira a otimizar a função objetivo ou mesmo em um critério guloso (por exemplo, ao grupo com o objeto mais próximo). É importante ressaltar que a silhueta de um objeto em um grupo *singleton* é *0*. Além disso, adicionar um objeto a um grupo, mesmo que seja o grupo com um objeto mais próximo, também pode reduzir o valor da silhueta da solução.

No que concerne aos resultados apresentados para as instâncias *não comportadas* na tabela V, observa-se que os valores de *H* foram próximos a *0,5*, e indicam que os objetos estão dispersos no espaço [1] [22]. Nesses casos, embora algoritmos e/ ou técnicas de agrupamento de dados produzam soluções, os grupos podem não possuir um significado. No que se refere ao índice Silhueta, mesmo com valores médios razoáveis (próximos a 0,5), existem muitos objetos que

possuem silhuetas baixas por não estarem alocados em grupos bem definidos, coesos e separados.

Nas soluções produzidas pelo DBSCAN tradicional para as instâncias *não comportadas* a quantidade de grupos obtida foi diferente da quantidade relatada no processo de construção da instância e também da quantidade identificada como ideal pelas heurísticas que alcançaram o valor da maior silhueta. Com base na *Estatística de Hopkins*, que utiliza conceitos de *Testes Estatísticos de Aleatoriedade Espacial*, a aplicação de algoritmos e/ ou técnicas de agrupamento de dados em instâncias que não apresentam tendência à formação de agrupamentos (nesse trabalho a maioria das instâncias *não comportadas*), pode resultar em soluções em que os grupos não possuem significados *"reais"* [1] [22].

B. Resultados Obtidos com o Método Proposto

A seção III apresentou os resultados da heurística baseada na *Estatística de Hopkins*. A partir dos resultados reportados nessa seção é possível observar que, de forma eficiente, o algoritmo foi capaz de identificar instâncias em que existe Tendência à formação de Agrupamentos. Foi verificado, também, que todas as instâncias que não possuem tendência à formação de agrupamentos foram classificadas como *não comportadas* por [6]. Entretanto, seis instâncias classificadas como *não comportadas* possuem tendência à formação de agrupamentos.

Ainda na seção III foram apresentados resultados obtidos com a heurística ILS-DBSCAN, que destacou-se obtendo resultados melhores ou equivalentes aos obtidos com o algoritmo da literatura AECBL1 e com tempos de processamento consideravelmente inferiores.

Com base nos resultados obtidos pela *Heurística para a Estatística de Hopkins*, ILS-DBSCAN e com o algoritmo da Literatura AECBL1, o presente trabalho propõe uma abordagem heurística híbrida (HH). Trata-se de uma heurística que utiliza a *Estatística de Hopkins* para identificar qual é a heurística mais adequada à resolução da instância submetida.

As tabelas VI, VII e VIII trazem resultados obtidos com a utilização da HH para, respectivamente, os conjuntos de dados DS1, DS3 e DS2. Essas tabelas apresentam, para cada instância, a silhueta, o tempo de processamento, o *gap* em relação ao algoritmo da literatura AECBL1 (tempo consumido em relação ao tempo de AECBL1) e a Sobrecarga (tempo excedido em relação ao tempo de AECBL1).

TABELA VI
EXPERIMENTO HEURÍSTICA HÍBRIDA: ALVO OU TEMPO MÁXIMO DE EXECUÇÃO – DS1

Instância	Silhueta	Tempo	Gap	Sobrecarga
ruspini	0,738	0,17	0,78%	0,00%
gauss9	0,482	228,89	100,00%	0,00%
maronna	0,575	178,70	100,00%	0,00%
vowel2	0,451	1821,63	100,00%	0,00%
yeast	0,628	372,61	100,00%	0,00%
200DATA	0,823	54,21	100,03%	0,03%
iris	0,687	17,92	100,08%	0,08%
spherical_4d3c	0,689	43,20	100,03%	0,03%
wine	0,660	35,77	100,04%	0,04%

Em especial, a tabela VIII (dividida em 3 partes) traz os resultados discriminando as instâncias classificadas como *Comportadas* e *Não Comportadas***,** bem como a identificação de Tendência à Formação de Agrupamentos.

TABELA VII
EXPERIMENTO HEURÍSTICA HÍBRIDA: ALVO OU TEMPO MÁXIMO DE EXECUÇÃO – DS3

Instância	Silhueta	Tempo	Gap	Sobrecarga
100p6c	0,736	0,17	0,04%	0,00%
181p	0,737	0,07	0,04%	0,00%
2000p11c	0,713	0,55	0,03%	0,00%
2face	0,667	0,03	0,06%	0,00%
300p4c	0,750	0,04	0,04%	0,00%
30p	0,787	6,57	39,34%	0,00%
350p5c	0,759	0,05	0,12%	0,00%
3dens	0,762	0,03	0,09%	0,00%
450p4c	0,766	0,06	0,07%	0,00%
500p3c	0,825	0,06	0,13%	0,00%
600p3c	0,751	0,08	0,11%	0,00%
900p5c	0,716	0,14	0,01%	0,00%
convdensity	0,854	0,82	3,30%	0,00%
convexo	0,668	0,03	0,11%	0,00%
moreshapes	0,732	1,01	0,89%	0,00%
numbers2	0,600	0,27	0,01%	0,00%
outliers	0,758	0,40	1,44%	0,00%
face	0,527	472,44	100,00%	0,00%
numbers	0,581	521,07	100,00%	0,00%
outliers_ags	0,748	12,48	100,12%	0,12%
97p	0,711	24,91	100,06%	0,06%
157p	0,666	52,83	100,03%	0,03%

(1) **Instâncias *Comportadas e com Tendência***: todas as 18 instâncias *comportadas*. Mesmo com a adição do tempo de processamento da *Heurística para a Estatística de Hopkins*, a economia mínima foi de cerca de 90%, na instância 1000p14c, onde a HH consumiu 10,36% do tempo gasto por AECBL1, e a maior economia foi de 99,7%.

(2) **Instâncias *não Comportadas e sem Tendência***: todas as instâncias que não possuem tendência foram classificadas por [6] como *não comportadas*. Entre as 26 instâncias nessa situação, a maior sobrecarga foi de apenas 0,08%. Além disso, em 12 instâncias a sobrecarga foi de 0,00%, para 18 instâncias a sobrecarga foi de até 0,01% e para 21 instâncias a sobrecarga foi de até 0,02%.

(3) **Instâncias *não Comportadas e com Tendência***: Seis instâncias estão nessa situação. Para duas instâncias houve sobrecarga no tempo de processamento (cerca de 34% para a instância *200p3c1* e cerca de 81% para a 300p2c1). Em relação às demais quatro instâncias, a economia máxima ocorreu para a instância *300p3c1* (cerca de 98%), a menor economia ocorreu para a instância *300p10c1*(cerca de 42%) e a economia média foi superior a 80%.

As tabelas IX e X apresentam estatísticas relacionadas à *Economia* e *Sobrecarga* resultantes da utilização da HH para todas as instâncias consideradas no presente trabalho. Com base nessas tabelas é possível observar que as economias média e mediana entre as instâncias que possuem tendência à formação de agrupamentos foram superiores a 96% e 99%, respectivamente. Além disso, em relação às instâncias sem tendência, as sobrecargas média e mediana foram de apenas 2,84% e 0,01%, respectivamente. É importante ressaltar que a média foi elevada devido aos casos de instâncias não comportadas que possuem tendência (*200p3c1* e *300p2c1*).

Somente para essas duas instâncias a sobrecarga foi superior a 0,08%.

TABELA VIII
EXPERIMENTO HEURÍSTICA HÍBRIDA: ALVO OU TEMPO MÁXIMO DE EXECUÇÃO – DS2

DS2 :: COMPORTADAS (COM TENDÊNCIA*)				
Instância	**Silhueta**	**Tempo**	**Gap**	**Sobrecarga**
100p10c	0,834	0,05	0,30%	0,00%
100p3c	0,786	0,03	0,15%	0,00%
100p7c	0,834	0,03	0,20%	0,00%
200p4c	0,773	0,03	0,09%	0,00%
300p3c	0,766	0,04	0,09%	0,00%
400p3c	0,799	0,05	0,10%	0,00%
500p3c	0,825	0,06	0,10%	0,00%
600p15c	0,781	0,19	0,07%	0,00%
700p4c	0,797	0,09	0,10%	0,00%
800p23c	0,787	0,45	0,20%	0,00%
900p12c	0,841	13,21	3,08%	0,00%
900p5c	0,716	0,14	0,11%	0,00%
1000p14c	0,831	25,66	10,36%	0,00%
1000p6c	0,736	0,16	0,09%	0,00%
1300p17c	0,823	2,37	1,02%	0,00%
1800p22c	0,802	0,52	0,03%	0,00%
1900p24c	0,799	0,78	0,04%	0,00%
2000p11c	0,713	0,55	0,04%	0,00%
2000p26c	0,799	0,64	0,03%	0,00%
DS2 :: NÃO COMPORTADAS (COM TENDÊNCIA*)				
Instância	**Silhueta**	**Tempo**	**Gap**	**Sobrecarga**
200p4c1	0,745	6,48	12,09%	0,00%
300p10c1	0,609	722,72	57,70%	0,00%
300p3c1	0,676	3,96	1,78%	0,00%
300p6c1	0,662	19,62	5,64%	0,00%
200p3c1	0,681	93,59	134,23%	34,23%
300p2c1	0,776	189,56	181,60%	81,60%
DS2 :: NÃO COMPORTADAS (SEM TENDÊNCIA*)				
Instância	**Silhueta**	**Tempo**	**Gap**	**Sobrecarga**
100p2c1	0,743	17,94	100,08%	0,08%
100p3c1	0,579	58,67	100,03%	0,03%
100p5c1	0,700	113,19	100,01%	0,01%
100p7c1	0,491	107,65	100,01%	0,01%
100p8c1	0,528	39,64	100,04%	0,04%
200p12c1	0,577	145,31	100,04%	0,01%
200p2c1	0,764	40,32	100,04%	0,04%
200p7c1	0,579	95,50	100,02%	0,02%
200p8c1	0,576	461,33	100,00%	0,00%
300p13c1	0,566	209,08	100,00%	0,00%
300p4c1	0,607	38,93	100,04%	0,04%
400p17c1	0,513	95,20	100,02%	0,02%
400p4c1	0,602	85,96	100,02%	0,02%
500p19c1	0,483	332,54	100,00%	0,00%
500p4c1	0,661	575,67	100,00%	0,00%
500p6c1	0,630	290,67	100,00%	0,00%
600p3c1	0,721	115,74	100,01%	0,01%
700p15c1	0,680	901,19	100,00%	0,00%
800p10c1	0,468	136,97	100,01%	0,01%
800p18c1	0,692	1490,98	100,00%	0,00%
800p4c1	0,702	136,28	100,01%	0,01%
1000p27c1	0,523	1398,77	100,00%	0,00%
1000p5c1	0,639	232,18	100,00%	0,00%
1100p6c1	0,671	467,13	100,00%	0,00%
1500p6c1	0,644	320,24	100,00%	0,00%
2000p9c1	0,624	393,66	100,00%	0,00%

TABELA IX
HEURÍSTICA HÍBRIDA EM RELAÇÃO AO TEMPO DE PROCESSAMENTO DO AECBL1

Economia em tempo de processamento				
Menor	**Mediana**	**Média**	**Maior**	**Nº instâncias**
42,30%	99,89%	96,50%	99,99%	48,78%

TABELA X
HEURÍSTICA HÍBRIDA EM RELAÇÃO AO TEMPO DE PROCESSAMENTO DO AECBL1

Sobrecarga em tempo de processamento				
Menor	**Mediana**	**Média**	**Maior**	**Nº instâncias**
0,00%	0,01%	2,84%	81,60%	51,22%

V. CONCLUSÕES E TRABALHOS FUTUROS

Com o objetivo de resolver o Problema de Agrupamento Automático, o presente trabalho propôs uma abordagem *Híbrida* que utiliza a *Estatística de Hopkins* para realizar o *Teste de Tendência de Agrupamento.*

Com base nos resultados do ILS-FJGP, em apenas 6% das instâncias a diferença em relação à melhor silhueta do AECBL1 foi inferior a 0,01. Os resultados indicam que as soluções iniciais obtidas pelo procedimento FJGP necessitam de mais refinamento. Nesse sentido, as gerações do Algoritmo Evolutivo da literatura (AECBL1) utilizam buscas locais e operadores genéticos (mutações, cruzamentos e seleções de indivíduos) em um conjunto de soluções (uma população).

Em experimentos preliminares realizados com o algoritmo DBSCAN, observou-se que para as instâncias *comportadas* foram obtidos resultados de alta qualidade, em que o *gap* da silhueta era de no máximo 0,1 em relação à melhor silhueta do AECBL1. Além disso, quando o número de grupos não foi o mesmo em relação ao melhor resultado apresentado na literatura, a diferença máxima foi de apenas 2 unidades. Com base nos resultados relatados, um algoritmo baseado em ILS foi proposto, e utilizou-se de um algoritmo baseado no DBSCAN para a construção de soluções iniciais (o ILS-DBSCAN).

Conforme foi apresentado nos resultados do ILS-DBSCAN, em cerca de 58% das instâncias a diferença em relação à melhor silhueta do AECBL1 foi inferior a 0,01. O AECBL1 destacou-se produzindo o melhor resultado em mais de 96% das instâncias utilizadas, enquanto o ILS-DBSCAN alcançou o melhor resultado para cerca de 50%. Porém, ao considerar apenas as instâncias comportadas, o ILS-DBSCAN alcançou os melhores resultados para todas as instâncias, enquanto o AECBL1 obteve os melhores resultados em cerca de 89% das instâncias. Além disso, para as instâncias *comportadas*, o ILS-DBSCAN consumiu em média apenas 0,9% do tempo de processamento do AECBL1 e cerca de apenas 3% do tempo no pior caso.

Com base nos resultados apresentados para os algoritmos ILS-DBSCAN e AECBL1, uma abordagem híbrida foi proposta. Nessa heurística, quando uma instância era identificada com *Tendência à formação de Agrupamentos* (Heurística para EH), o ILS-DBSCAN era selecionado para a resolução do problema. Caso contrário, o AECBL1 deveria ser executado. O objetivo foi a obtenção dos melhores resultados com um menor custo computacional, sendo importante destacar que, no pior caso, a sobrecarga ocasionada pela heurística para EH foi baixa, uma vez que o tamanho de amostra de 1% foi suficiente para a obtenção de resultados de alta qualidade.

Não obstante, de forma a reforçar ainda mais esta análise, em trabalhos futuros serão efetuadas novas análises com mais instâncias da literatura. Seguem algumas propostas para

trabalhos futuros: Utilização de técnicas de processamento paralelo e distribuído; aprofundar estudos sobre tendência para formação de agrupamentos e sobre o índice Silhueta.

AGRADECIMENTOS

Os autores agradecem a CAPES, CNPq, FAPERJ e PROPPi/UFF (Pró-Reitoria de Pesquisa, Pós-graduação e Inovação da UFF) pelo apoio ao desenvolvimento deste trabalho.

REFERÊNCIAS

[1] Han, J., e Kamber, M., *Cluster Analysis*. In: Morgan Kaufmann. Publishers (eds.), *Data Mining: Concepts and Techniques*, 3 ed., chapter 8, New York, USA, Academic Press, 2012.

[2] Kumar, V., Steinbach, M.,Tan, P. N. Introdução ao Data Mining - Mineração De Dados. Ciência Moderna, 2009.

[3] Alexeis Joel Ochoa Reyes, A.J.O., Arturo Orellana Garcia, A.O., Mui, Y.L. System for Processing and Analysis of Information Using Clustering Technique. IEEE Latin America Transactions. VOL. 12, NO. 2, 2015.

[4] C. D. Guerrero, D. Salcedo and, H. Lamos "A Clustering Approach to Reduce the Available Band width Estimation Error" IEEE LATIN AMERICA TRANSACTIONS, VOL. 11, NO. 3, 2013.

[5] J. C. Riquelme, R. Ruiz, D. Rodríguez and, J. S. Aguilar-Ruiz "Finding Defective Software Modules by Means of Data Mining Techniques" IEEE LATIN AMERICA TRANSACTIONS, VOL. 7, NO. 3, 2009.

[6] Cruz, M. D. O Problema de Clusterização Automática. Tese de Doutorado, COPPE\UFRJ, Rio de Janeiro, 2010.

[7] Pelleg, D. & A. Moore. *X-means: extending k-means with efficient estimation of the number of clusters*. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 727–734, 2000.

[8] An Efficient *K'-Means Clustering Algorithm*, Pattern Recognition Letters 29, 2008.

[9] Soares, A. S. R. F. Metaheurísticas para o Problema de Clusterização Automática, Dissertação de Mestrado, UFF - Niterói, 2004.

[10] Hruschka, E. R., R. J. G. B. Campello, & L. N. de Castro. *Improving the efficiency of a clustering genetic algorithm. In Advances in Artificial Intelligence* - IBERAMIA 2004: 9th Ibero-American Conference on AI, Puebla, Mexico, November 22-25. Proceedings, Volume 3315, pp. 861–870. Springer-Verlag GmbH, Lecture Notes in Computer Science, 2004.

[11] Hruschka, E. R., R. J. G. B. Campello, & L. N. de Castro. *Evolving clusters in gene-expression data. Information Sciences* 176 (13), 1898–1927, 2006.

[12] Alves, V., R. Campello, and E. Hruschka. *Towards a fast-evolutionary algorithm for clustering. In IEEE Congress on Evolutionary Computation*, 2006, Vancouver, Canada, pp. 1776–1783, 2006.

[13] Tseng, L. & . Yang, S.B.. *A genetic approach to the automatic clustering problem*. Pattern Recognition 34, 415–424, 2001.

[14] Naldi, M. C. & A. C. P. L. F. Carvalho (2007). *Clustering using genetic algorithm combining validation criteria*. In Proceedings of the 15th European Symposium on Artificial Neural Networks, ESANN 2007, Volume 1, pp. 139–144.

[15] Cruz, M. D. e Ochi, L. A multi-start heuristic based on GRASP for an automatic clustering Problem. Pesquisa Operacional para Desenvolvimento - PODes, Vol 7(2) , pp. 130-146, 2015.

[16] Semaan, G. S., Cruz, M.D., Brito, J. A. M., and Ochi, L. S. "*Proposta de um método de classificação baseado em densidade para a determinação do número ideal de grupos em problemas de clusterização*", Learning & Nonlinear Models vol. 10 número 4, 2012.

[17] Xiong, H., Li, Z. (2014). Clustering Validation Measures. In C. Aggarwal & C. Reddy (Eds.), Data Clustering: Algorithms and Applications (pp.571-605). CRC Press, 2014.

[18] Naldi, C. N. *Técnicas de Combinação para Agrupamento Centralizado e Distribuído de Dados*. Tese de Doutorado, USP - São Carlos, 2011.

[19] Ester, M., H.-P. Kriegel, J. Sander, & X. Xu. *A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining* (KDD'96), pp. 226–231, 1996.

[20] Rousseeuw, P. J. *Silhouettes: a graphical aid to the interpretation and validation of cluster analysis*. Journal of Computational and Applied Mathematics 20, 53–65, 1987.

[21] Wang et al., Wang, X., Qiu, W., Zamar, R. H. CLUES: A non-parametric clustering method based on local shrinking. Computational Statistics & Data Analysis 52, 2007.

[22] Banerjee, A., Dave, R.. Validating clusters using the hopkins statistic. IEEE International Conference on Date of Conference, 2004.

[23] Assunção, R., Reis, I. Testing spatial randomness: a comparison between T^2 methods and modifications of the Angle test. Brazilian Journal of Probability and Statistics, 14(1), 71-86, 2000.

[24] Campello, R. J. G. B., E. R. Hruschka, & V. S. Alves. *On the efficiency of evolutionary fuzzy clustering. Journal of Heuristics* 15 (1), 43–75, 2009.

[25] Pakhira, M., S. Bandyopadhyay, & U. Maulik. *A study of some fuzzy cluster validity indices, genetic clustering and application to pixel classification*. Fuzzy Sets Systems 155 (2), 191–214, 2005.

[26] Fisher, R. The use of multiple measurements in taxonomic problems. Annual Eugenics 7, 1936.

[27] Ruspini, E. H. Numerical methods for fuzzy clustering. Information 8Science, 1970.

[28] Maronna, R. and Jacovkis, P. M. Multivariate clustering procedures with variable metrics. Biometrics 30, 1974.

[29] Hastie, t., Tibshirani, R., and Friedman, J. The elements of statistical learning. Data Mining, Inference, and prediction, 2001.

[30] Wang et al., Wang, X., Qiu, W., Zamar, R. H. CLUES: A non-parametric clustering method based on local shrinking. Computational Statistics & Data Analysis 52, 2007.

[31] Aiex, R. M., Resende, M. G. C., and Ribeiro, C. C. (2007). Ttt plots: a perl program to create time-to-target plots. Optimization Letters, 1:355-366, 2007.

[32] Gendreau, M., Potvin, J.Y. Handbook of Metaheuristics, Springer, 2010.

[33] Pacheco, T. M.; Brugiolo, L.; Str Oele, V.; Soares, S. S. R. F. An Ant Colony Optimization for Automatic Data Clustering Problem. In: 2018 IEEE Congress on Evolutionary Computation (CEC), 2018, Rio de Janeiro. 2018 IEEE Congress on Evolutionary Computation (CEC), 2018.p.1-8.

[34] Zhou, X.; Miao, F.; Ma, H. Genetic Algorithm with an Improved Initial Population Technique for Automatic Clustering of Low-Dimensional Data. Information 2018, 9, 101.

[35] Kuo, R.J. & Zulvia, F.E. Knowledge and Information Systems (2018) Volume 57, Issue 2, pp 331–357.

[36] Zhou, Y., Wu, H., Luo, Q., Abdel-Baset, M. Automatic data clustering using nature-inspired symbiotic organism search algorithm. Knowledge-Based Systems. Volume 163, 2019, Pages 546-557.

Gustavo Silva Semaan Professor Adjunto da Universidade Federal Fluminense no Instituto do Noroeste Fluminense de Educação Superior (INFES - UFF). Doutor em Computação (Algoritmos e Otimização) e Mestre em Computação Otimização Combinatória e Inteligência Artificial) pelo Instituto de Computação da UFF. Bacharel em Sistemas de Informação pela Faculdade Metodista Granbery. http://lattes.cnpq.br/4519888592231795.

José André de Moura Brito tem bacharelado em Matemática pela Universidade Federal do Rio de Janeiro (1997), Mestrado em Engenharia de Sistemas e Computação (Otimização) pela Universidade Federal do Rio de Janeiro (1999), Doutorado em Engenharia de Sistemas e Computação (Otimização) pela Universidade Federal do Rio de Janeiro (2004) e Pós-Doutorado em Otimização na UFF (2008). http://lattes.cnpq.br/9036541085964477.

Augusto Cesar Fadel é Bacharel em Estatística pela Escola Nacional de Ciências Estatísticas (ENCE) e mestrando em Ciência da Computação na Universidade Federal Fluminense (UFF), onde desenvolve pesquisa na área de algoritmos e otimização. Atua como estatístico na Fundação Instituto Brasileiro de Geografia e Estatística (IBGE), onde desenvolve atividades relacionadas a controle estatístico de sigilo, uso de big data em estatísticas oficiais. http://lattes.cnpq.br/4292194013404546.

Luiz Satoru Ochi possui graduação em Matemática pela Universidade Estadual Paulista Júlio de Mesquita Filho (UNESP), Mestrado em Matemática Aplicada pela Universidade Estadual de Campinas (IMECC-UNICAMP) e Doutorado em Engenharia de Sistemas e Computação pela Universidade Federal do Rio de Janeiro (COPPE-SISTEMAS/UFRJ). Atualmente é Pesquisador nível 1C, do CNPq, comitê de Ciência da Computação. http://lattes.cnpq.br/9171815778534257.