UM NOVO MÉTODO BASEADO EM GRADE E DENSIDADE COM TRATAMENTO DE RUÍDOS PARA A IDENTIFICAÇÃO DO NÚMERO IDEAL DE GRUPOS

Wálace Cotrim Rodrigues
Centro Universitário Plinio Leite – UNIPLI
walacerodrigues56@gmail.com

Augusto César Fadel[1]
Instituto Brasileiro de Geografia e Estatística – IBGE
augustofadel@gmail.com

Gustavo Silva Semaan
Instituto do Noroeste Fluminense de Educação Superior – INFES/UFF
gsemaan@ic.uff.br

José André de Moura Brito
Escola Nacional de Ciências Estatísticas – ENCE/IBGE
jambrito@gmail.com

RESUMO

A área de *cluster analysis* agrega diversos métodos que têm como objetivo a identificação de grupos dentro de um conjunto de dados. Considerando tal objetivo, o presente trabalho traz a proposta de um novo método para a solução do Problema de Agrupamento Automático (PAA). O método, desenvolvido a partir do estudo de uma técnica baseada em grade e densidade, determina o número ideal de grupos com base na maximização do índice de silhueta. Ele implementa um procedimento para o deslocamento da grade, uma busca local para unir células densas vizinhas e realiza a identificação de objetos do tipo ruído. O método foi aplicado em um conjunto com 63 instâncias da literatura, sendo produzidos resultados de boa qualidade no que concerne aos valores observados para o índice de silhueta.

PALAVRAS CHAVE. Agrupamento Automático, Densidade, Grade, Índice Silhueta.

Otimização Combinatória, Metaheurísticas, Estatística.

ABSTRACT

The cluster analysis field has several methods that aim to identify groups within a dataset. This paper presents a new method for solving the automatic clustering problem (ACP). The method, developed based on studies of density and grid concepts, determines the ideal number of clusters in a given dataset by maximizing the silhouette index. A procedure was implemented for the grid shift, local search was performed to join neighboring dense cells and noise objects were identified. The method was applied to 63 well known datasets where good quality results were obtained, regarding the silhouette index.

KEYWORDS. Automatic Clustering. Density. Grid. Silhouette Index.

Combinatorial optimization, Metaheuristics, Statistics.

[1] O IBGE está isento de qualquer responsabilidade pelas opiniões, informações, dados e conceitos emitidos neste artigo, que são de exclusiva responsabilidade do autor.

1. Introdução

A análise de agrupamentos é uma técnica de análise multivariada [Johnson e Wichern, 2002] que agrega um conjunto de métodos cujo objetivo principal é segmentar uma base de dados, constituída por objetos definidos por certas características (atributos), em grupos homogêneos e o mais separados possível, ou seja, que reúnam objetos semelhantes, no que diz respeito a uma métrica de similaridade. Pensando o problema de agrupamento como um problema de otimização, deve-se maximizar a similaridade (homogeneidade) entre os objetos de um mesmo grupo e minimizar a similaridade entre objetos de grupos distintos [Han e Kamber, 2006] sendo tal similaridade função dos atributos.

Formalmente, 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 *f* atributos (características), ou seja, $x_i=\{x_i^1, x_i^2,..., x_i^f\}$, deve-se construir *k* grupos C_j (j=1,...,k) a partir de X, sendo tais grupos correspondentes a uma solução (ou partição). Ao construir os grupos, ou seja, produzir uma solução π, tal que, $\pi=\{C_1, C_2,..., C_k\}$, deve-se garantir que os objetos de cada grupo sejam homogêneos entre si, segundo alguma medida de similaridade. Além disso, considerando o problema de agrupamento clássico, cada solução produzida, independentemente do método de agrupamento, deve satisfazer três restrições, quais sejam: (1) o conjunto X corresponde à união dos objetos dos grupos; (2) cada objeto pertence a exatamente um grupo; (3) todos os grupos possuem pelo menos um objeto; definidas, respectivamente, segundo as Equações (1), (2) e (3) apresentadas a seguir. Além dessas restrições, em problemas de agrupamento específicos, como os relatados nos trabalhos de [Hruschka e Ebecken, 2001] [Han e Kamber, 2006], com restrições particulares.

$$\bigcup_{j=1}^{k} C_j = X \quad (1) \qquad C_i \cap C_j = \emptyset \quad \begin{matrix} i,j = 1,\dots,k \\ i \neq j \end{matrix} \quad (2) \qquad C_j \neq \emptyset \quad j = 1,\dots,k \quad (3)$$

No que diz respeito ao PA, o número de maneiras em que os *n* objetos podem ser agrupados (soluções possíveis) em *k* grupos é dado pelo número de Stirling (NS) de segundo tipo [Johnson e Wichern, 2002] (Equação 4). No problema de agrupamento automático (PAA), a identificação da quantidade de grupos (*k*) faz parte do problema e, portanto, o número de soluções possíveis corresponde ao somatório da Equação 5 para o número de grupos variando no intervalo $[1,k_{max}]$, sendo k_{max} o número máximo de grupos.

$$NS(n,k) = \frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j}\binom{k}{j}j^n \quad (4) \qquad NS(n) = \sum_{j=1}^{k_{max}} NS(n,j) \quad (5)$$

Para ilustrar a ordem de grandeza da quantidade de soluções possíveis, no caso de 20 (n=20) objetos e 3 grupos (*k*=3), o número de soluções possíveis aumenta é igual a NS(20, 3)=580.606.446. Nesses dois problemas (PA e PAA) a quantidade de soluções possíveis cresce exponencialmente com o aumento da quantidade de objetos. Essa característica torna proibitiva a obtenção do ótimo global mediante a aplicação de procedimentos de enumeração exaustiva ou de métodos de enumeração implícita que são aplicados à resolução de formulações de programação matemática associadas a alguns problemas de agrupamento. Em função da complexidade desse problema e da sua correspondência com uma grande variedade de aplicações reais, as últimas décadas foram marcadas pela pesquisa e desenvolvimento de diversos algoritmos de agrupamento. Tais algoritmos têm sido aplicados em diversos domínios, quais sejam: inteligência artificial, reconhecimento de padrões, marketing, economia, ecologia, estatística, pesquisas médicas, ciências políticas etc. Porém, nenhum desses algoritmos é apropriado para todos os tipos de dados (por exemplo, variáveis numéricas ou nominais), formatos de grupos (por exemplo, formatos alongado, hiperesférico ou elipsoidal) e aplicações [Tan et al., 2009].

O restante deste trabalho está dividido em quatro seções, incluindo a Introdução. A Seção 2 apresenta uma revisão da literatura com uma breve descrição dos algoritmos que tratam

o PAA. Ainda nessa seção é apresentado o índice silhueta para avaliação e validação das soluções. Já a Seção 3 apresenta o método baseado em densidade e grade proposto neste trabalho. A Seção 4 traz os resultados computacionais obtidos, enquanto a Seção 5 apresenta as conclusões do trabalho e sugere possíveis trabalhos futuros.

2. Revisão da Literatura

O presente trabalho considera como proposta para solução do PAA a aplicação de índices de validação a soluções produzidas a partir de múltiplas execuções de um algoritmo de agrupamento. O método adotado é baseado nos conceitos de densidade e grade, introduzidos por [Rakesh et al., 1999], no qual a variedade do espaço de soluções se dá através da variação do tamanho da célula e do deslocamento da grade. Uma vez obtidas as soluções, o índice silhueta, proposto por [Rousseeuw, 1987], é aplicado para avaliá-las. É escolhida como solução final aquela que obtém o maior do índice. Método similar foi apresentado por [Semaan et al., 2015], todavia, sem considerar os procedimentos de deslocamento de grade, que traz grande contribuição no que diz respeito à variedade do espaço de soluções, e identificação de ruído, que favorece a obtenção da solução ótima em instâncias com objetos dispersos ou grupos parcialmente sobrepostos, especialmente no que diz respeito a identificação correta do número de grupos. Essa seção apresenta uma breve revisão da literatura acerca dos conceitos adotados.

No que diz respeito aos algoritmos de agrupamento, os métodos baseados em densidade classificam como grupos as regiões em que há o maior número de elementos (objetos) no espaço de dados que são separados pelas regiões de baixa densidade, ou seja, com menor concentração de objetos [Han e Kamber, 2006]. Uma das principais características desses métodos é a habilidade em identificar grupos de formatos arbitrários. É exemplo de algoritmo baseado em densidade o algoritmo *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN) [Ester et al., 1996]. Sua aplicação permite a identificação de grupos de objetos em regiões de alta densidade, grupos de formatos arbitrários em dados multidimensionais e objetos do tipo ruído.

O algoritmo CLIQUE, proposto por [Rakesh et al., 1999], aborda o PA através de um método híbrido baseado em densidade e grade. A base de dados é particionada em subespaços (grade de células), de maneira que cada objeto passa a ser representado pela célula que ocupa na grade e o conceito de densidade se dá em termos do número de objetos em cada célula e da maneira com a qual as células densas se distribuem na grade. [Han e Kamber, 2006] apontam como a principal vantagem dessa abordagem o fato de que o número de operações do algoritmo não é função do número de objetos na instância, mas do número de células da grade, parâmetro controlado. Tal característica melhora o desempenho, em termos de tempo de processamento, dos algoritmos baseados nessa heurística.

Muitos algoritmos de agrupamento necessitam do número de grupos como parâmetro de entrada e, segundo [Tan et al., 2009], talvez um dos problemas de seleção de parâmetros mais conhecido seja o de determinar o número ideal de grupos em um PA. Duas estratégias são bem conhecidas para tal problema: (i) realizar múltiplas execuções de algoritmos, variando os parâmetros de entrada, e comparar as soluções produzidas através de índices de validação relativos; (ii) aplicar técnicas de avaliação de tendência de agrupamento. Na primeira, é assumido como número ideal de grupos aquele indicado pela solução de melhor desempenho, segundo os índices de validação adotados. Essa estratégia, em geral, apresenta custo computacional mais elevado, entretanto, oferece uma solução final para o problema de agrupamento. A segunda abordagem consiste, basicamente, em aplicar técnicas capazes de identificar o número de grupos sem executar um algoritmo de agrupamento, tais como as técnicas de geração de imagens de dissimilaridade reordenada (do inglês, *reordered dissimilarity images*, RDI). Dentre as mais presentes na literatura estão aquelas baseadas no algoritmo *Visual Assessment Tendency* (VAT), proposto por [Bezdeck et al., 2002]. Embora o custo computacional da abordagem (ii) tenda a ser inferior ao da abordagem (i), após a obtenção do número de grupos, em geral ainda é necessário executar um algoritmo de agrupamento para obter a solução final. A estratégia (i) é adotada em [Naldi, 2011], sendo propostas duas abordagens sistemáticas que atuam na maximização do índice silhueta (índice de validação) e que consistem em múltiplas execuções do algoritmo

k-means. São elas: o *MRk-means* (do inglês *Multiple Runs of k-means*) e o *OMRk-means* (do inglês *Ordered Multiple Runs of k-means*). Um método sistemático (MS) baseado em densidade é proposto em [Semaan et al., 2012a]. O MRDBSCAN (do inglês, *Multiple Runs of DBSCAN*) realiza múltiplas execuções do algoritmo DBSCAN [Ester et al., 1996] considerando uma estratégia de calibração dos parâmetros de raio de abrangência e densidade. Em [Semaan et al., 2012b] é proposto um método sistemático hierárquico para a determinação do número ideal de grupos.

Vários trabalhos da literatura propõem meta-heurísticas para o PAA cujo objetivo é encontrar um número ideal de grupos e sua solução correspondente [Tseng e Yang, 2001] [Hruschka e Ebecken, 2003] [Soares e Ochi, 2004] [Hruschka et al., 2004a] [Cruz, 2010].

Embora o foco do presente trabalho seja a resolução do PAA por meio de múltiplas execuções do método baseado em grade e densidade, é muito importante abordar na revisão o clássico método *k-means* para o PA. Ainda que o primeiro trabalho relativo a esse método seja de 1955, esse método ainda inspira várias novas abordagens na área de análise de agrupamentos [Jain, 2010]. Especificamente sobre o PAA podem-se citar heurísticas derivadas do *k-means*, como: *k'-means* [Zalik, 2008] e o *bisecting k-means* [Steinbach et al., 2000]. O *k-means* ainda é muito utilizado no projeto de meta-heurísticas, mais especificamente, como parte de procedimentos de construção, refinamento (busca local) ou mesmo de perturbação. Trabalhos que propõem heurísticas sofisticadas para resolver PAA têm sido publicados, como algoritmos genéticos ou evolutivos [Liu et al., 2012] e *Greedy Randomized Adaptive Search Procedure* (GRASP) [Nascimento et al., 2010].

Em [Ma e Chow, 2004], o método híbrido de densidade e grade é aplicado em conjunto com um procedimento de deslocamento de grade. Duas divisões das dimensões do espaço de dados são consideradas: a primeira é obtida pela construção da grade através da divisão dos atributos em partes iguais; a segunda é obtida através do deslocamento da grade formada, equivalente à metade do tamanho da célula. No trabalho de [Oliveira, 2007] é proposto um novo algoritmo evolutivo para a tarefa de análise de agrupamento chamado EDACluster. O método de densidade e grade é aplicado na composição dos grupos. Após a identificação dos centros, as células adjacentes são sucessivamente adicionadas ao grupo desde que sua densidade mantenha-se acima do um limite definido. Em [Cruz e Ochi, 2015a], uma heurística *multi start* baseada no GRASP, incluindo dois operadores de busca local e um procedimento de reconexão por caminhos, é aplicada para resolver o PAA. [Cruz e Ochi, 2015b] abordam o problema através de um algoritmo evolutivo com busca local, associado a uma formulação de programação inteira.

Finalmente, no que diz respeito aos índices de validação, os índices relativos têm como finalidade avaliar a qualidade relativa das soluções produzidas por diferentes métodos de agrupamento. Esses índices não têm a propriedade de monotonicidade, ou seja, não são impactados pelo aumento ou pela redução do número de grupos da solução. Desta forma, podem ser utilizados para a avaliação de diversas soluções, provenientes de diversos algoritmos [Naldi, 2011]. Em particular, o índice silhueta é um índice de validação relativo proposto por [Rousseeuw, 1987]. Ele possibilita a avaliação da qualidade das soluções com base na proximidade entre os objetos de determinado grupo (similaridade *intra-cluster*) e na distância dos objetos desse grupo aos objetos do grupo mais próximo (similaridade *inter-cluster*). Esse índice combina as ideias de coesão e de separação, sendo calculado para cada objeto x_i de X e seu valor pertence ao intervalo [-1, 1]. Esse índice é calculado para cada objeto e, com base em seu valor, é possível identificar se o objeto está alocado ao grupo mais adequado. Nesse sentido, um objeto que possui silhueta positiva e próxima a 1 está bem localizado em seu grupo, enquanto valores negativos indicam que o objeto está mais próximo de outro(s) grupo(s) vizinho(s).

Após o cálculo da silhueta de cada objeto, pode-se obter a silhueta de uma solução π por meio da utilização da Equação 11. Trata-se da média das silhuetas de todos os n objetos do conjunto X. É importante destacar que a silhueta de um objeto de um grupo *singleton* (grupo com apenas um objeto) é zero. Os passos a seguir explicam o cálculo do índice silhueta:

i. Neste trabalho, d_{ij} (Equação 6) corresponde à distância euclidiana entre os objetos x_i e x_j e q representa os atributos, onde q=1, ..., f. Para cada objeto x_i calcula-se a sua distância média $a(x_i)$ (Equação 7) em relação aos demais objetos do mesmo grupo (similaridade *intra-cluster*).

$$d_{ij} = \sqrt{\sum_{q=1}^{f} \left(x_i^q - x_j^q\right)^2} \quad (6)$$

$$a(x_i) = \frac{1}{|C_w| - 1} \sum d_{ij} \quad \forall x_j \neq x_i \quad x_j \in C_w \quad (7)$$

ii. A Equação 8 apresenta a distância entre o objeto x_i e os objetos de outro 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 9). As Equações 8 e 9 possibilitam o cálculo da similaridade *inter-cluster*.

$$d(x_i, C_t) = \frac{1}{|C_t|} \sum d_{ij} \quad \forall x_j \in C_t \quad (8)$$

$$b(x_i) = \min d(x_i, C_t) \quad C_t \neq C_w \quad C_t \in C \quad t = 1, \dots, k \quad (9)$$

iii. O coeficiente silhueta de cada objeto x_i ($s(x_i)$) é obtido mediante a aplicação da Equação 10. Destaca-se que, nessa equação, o termo do numerador corresponde à diferença entre as similaridades *inter-clusters* ($b(x_i)$, Equação 9) e *intra-cluster* ($a(x_i)$, Equação 7).

$$s(x_i) = \frac{b(x_i) - a(x_i)}{\max\{b(x_i), a(x_i)\}} \quad (10)$$

iv. O cálculo da silhueta de uma solução π={C_1,...,C_k} é a média das silhuetas dos objetos (Equação 11) e n é a quantidade de objetos da solução. Essa função deve ser maximizada.

$$\max Silhueta(\pi) = \frac{1}{n} \sum_{i=1}^{n} s(x_i) \quad (11)$$

3. Método Sistemático Baseado em Densidade e Grade

O método proposto neste trabalho foi desenvolvido com base no da técnica definida como densidade e grade e nos conceitos do algoritmo CLIQUE [Rakesh et al., 1999]. Tal método permite identificar grupos, conforme a associação e localização de cada objeto de uma instância na grade, formando soluções para o problema de agrupamento automático (PAA). Um Algoritmo de Grade similar foi desenvolvido por [Ma e Chow, 2004] utilizando esse mesmo conceito, associado a um procedimento de deslocamento de grade. O presente trabalho utiliza uma nova abordagem desse algoritmo, proposta por [Semaan et al., 2015], e acrescenta dois novos procedimentos: identificação de ruídos e deslocamento de grade, com objetivo de formar soluções inéditas e encontrar a quantidade ideal de grupos. Para avaliar as soluções obtidas foi utilizado o índice silhueta tradicional, dentre todas as iterações a solução que obteve o melhor resultado é apresentado como a melhor solução. Dois parâmetros devem ser fornecidos pelo usuário: o número máximo de iterações desejadas e a proporção de objetos do tipo ruído a desconsiderar no processo de agrupamento, sendo, esse, um vetor. Uma vez que não é possível conhecer a proporção de ruídos em uma instância real, os elementos de tal vetor devem ser arbitrados dentro de um intervalo considerado razoável, a partir de uma análise exploratória dos dados. É recomendável que o vetor sempre contenha o zero, a fim de permitir a obtenção de soluções contendo todos os objetos da instância. Finalmente, em cada iteração, um parâmetro de perturbação, *fator*, é gerado, ampliando o espaço de soluções. Os passos para execução do algoritmo proposto são descritos a seguir.

1. No primeiro passo seleciona-se a instância associada a um conjunto de objetos definido por X={x_1,x_2,x_3,...,x_i,...,x_n}, sendo cada x_i um objeto com f atributos, x_i={x_i^1,x_i^2,...,x_i^f}. A quantidade de iterações do algoritmo é fornecida como parâmetro de entrada. Em cada iteração, os passos a seguir são executados.

O algoritmo é executado para cada par de atributos, logo o método considera a associação de todos os atributos, por exemplo, para uma instância de 3 atributos, são definidas grades com as combinações c={(x_i^1,x_i^2), (x_i^1,x_i^3), (x_i^2,x_i^3)}. No exemplo esse passo será executado apenas uma vez a cada iteração, pois a instância só possui dois atributos.

2. O próximo passo consiste em definir o valor de m_{Cell}, conforme descrito na Equação 12, sendo min_{gap} correspondente à menor diferença entre o maior e menor valor de cada atributo *f* da instância X e *fator é* o parâmetro de perturbação utilizado para tornar aleatório o valor de m_{Cell} em múltiplas execuções. O parâmetro *fator* é um número pseudoaleatório, gerado conforme descrito na equação 13, e calibrado para assumir a valores no intervalo [0,5;2,0]. Experimentos preliminares indicaram que o algoritmo produz melhores resultados com o valor do parâmetro de perturbação compreendido nesse intervalo, uma vez que esse parâmetro influencia no tamanho das células da grade. Isso se deve ao fato de que células de tamanho muito grande podem, por exemplo, unir grupos que deveriam estar separados. Em contrapartida, células muito pequenas aumentam o tempo de processamento (aplicação da busca local) necessário para obter uma solução equivalente. Por fim, ainda com base na equação 12, *n* é a quantidade de objetos da instância. Na instância exemplo o *gap* do atributo *x* é dado por *Gap*(*x*)=8–1=7, o *gap* do atributo *y* é dado por *Gap*(*y*)=8–1=7, o que implica min_{gap}(x,y)=7. Assim, dado *n*=11 e, supondo parâmetro *fator* ≈ 0,95, a fim de facilitar a demonstração do algoritmo, o valor final de m_{Cell} é 2.

$$m_{cell} = \frac{min_{gap}}{\sqrt{n}} * fator \qquad (12) \qquad\qquad fator = \frac{500 + Rand(1501)}{1000} \qquad (13)$$

3. Em seguida é necessário definir a grade. Para cada par de atributos, as dimensões da região da grade são definidas pelo *gap* (diferença) entre o maior e o menor valor de cada um dos atributos selecionados no passo 2, formando uma grade bidimensional. A grade é então dividida em células utilizando a variável m_{Cell}. Com o objetivo de que todas as células tenham o mesmo tamanho. Caso a divisão da dimensão por m_{Cell} não tenha valor inteiro, o tamanho da dimensão em questão é arredondado para cima. No exemplo, a dimensão *x* tem *gap* = 7 e, para m_{Cell} = 2, seria necessário construir 3,5 células. Sendo assim, um espaço extra é acrescentado à grade, possibilitando a formação de 4 células. Com este ajuste, a grade passa a ter dimensão 8x8 e, após divisão, um total de 16 células (Figura 1). Esse procedimento é repetido para os outros atributos e, naturalmente, o espaço extra não contém objetos.
4. Após a definição da grade, os objetos são associados às células, ou seja, é determinado a qual célula cada objeto pertence. Para isso**,** o algoritmo leva em consideração os atributos dos objetos como coordenadas para sua localização na grade (Figura 2). Cada objeto deve pertencer a uma e somente uma célula e, caso o objeto esteja na fronteira entre duas células, é necessário arbitrar a qual célula o objeto deve pertencer. No presente estudo, objetos nessa situação foram associados à célula imediatamente inferior.

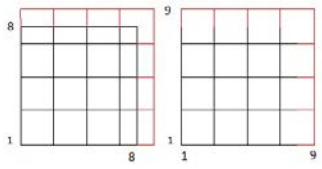

Figura 1 - Dimensão e divisão da grade.

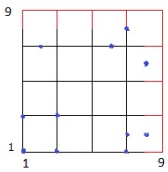

Figura 2 - Associação dos objetos.

5. Após todos os objetos estarem alocados às células, totaliza-se a quantidade de objetos em cada célula, a fim de gerar um vetor *vGrau* (utilizado na identificação de ruídos). Para isso**,** as quantidades de objetos das células são atribuídas a cada posição de um vetor temporário dinâmico. As quantidades distintas desse vetor são armazenadas no vetor *vGrau*, que é ordenado de maneira crescentemente. A Figura 3 traz o resultado obtido na instância exemplo.

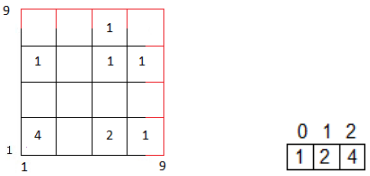

Figura 3 - Quantidade de objetos por célula e o vetor *vGrau* correspondente.

6. O presente passo é responsável pela identificação dos objetos ruído[2]. Nesse sentido o vetor *vGrau* é utilizado e, com base no parâmetro *l* submetido ao algoritmo, as *l* primeiras posições do vetor *vGrau* (com os menores valores) são consideradas. A cada iteração uma dessas posições irá indicar a densidade mínima que deve ser considerada, e as células com densidade inferior ao valor indicado serão desconsideradas.
7. Um ponto importante nessa etapa é o conceito de objetos limítrofes, também utilizado no DBSCAN: se uma célula é desconsiderada inicialmente, segundo o critério do número mínimo de objetos, mas é vizinha de uma célula considerada densa, ela deve ser considerada (Figura 4 (b)). Esse procedimento pode ter grande impacto na solução. Por exemplo, uma célula desconsiderada (não densa) for vizinha a duas células densas, ela é considerada e pode unir as células vizinhas. No exemplo, a quantidade de valores mínimos distintos considerados em *vGrau* é 3 (comprimento do vetor).
8. Assim, os próximos passos consideram *vGrau* = {1,2,4} como quantidades mínimas para a célula (Figura 4).

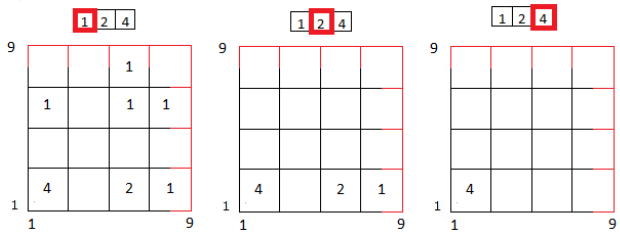

Figura 4 - Iteração Elimina Ruídos.

9. Uma vez desconsiderados os ruídos, cada célula que possui ao menos um objeto passa a formar um grupo. A busca local é então aplicada verificando as ligações entre as células vizinhas com maior quantidade de objetos e realizando as uniões necessárias. São consideradas vizinhas todas as células adjacentes, conforme Figura 6.

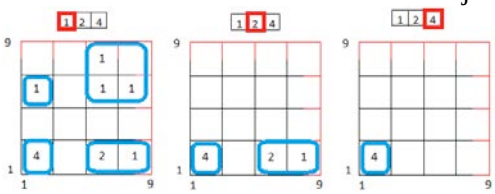

Figura 5 - Busca local para vGrau 1, 2 e 4.

Figura 6 - Alcance da busca local.

10. A solução é então formada no formato *groupnumber*, em que em um vetor o índice é o objeto em questão e o valor é o grupo ao qual o objeto pertence. Objetos do tipo ruídos são alocados em um grupo especifico, de rótulo -1(que é desconsiderado). Ao concluir a busca local é verificado a qual grupo cada objeto foi alocado. Para avaliar a qualidade da solução obtida aplica-se o índice silhueta tradicional, apresentado na seção 2.1.
11. Em alguns casos, a maneira como a grade é dividida pode unir grupos que deveriam estar separados ou preservar objetos que deveriam ter sido desconsiderados, mas que ficaram no limite interior de uma célula densa, fazendo com que não fossem eliminados no passo 7. De forma a mitigar esse efeito, para cada m_{Cell}, calculado no passo 2, além de

[2] É possível executar o algoritmo sem verificar e eliminar objetos ruído. Para tal, basta desconsiderar os passos 6 e 7 ou definir *vGrau* com apenas um elemento com valor 1, uma vez que, pela definição do problema de agrupamentos, um grupo deve possuir pelo menos um objeto.

gerar grades na formação normal, são geradas grades deslocadas em ½ de m_{Cell} para cima e ½ de m_{Cell} para a direita. Na prática a grade receberá ½ de m_{Cell} em cada borda, aumentando a região associada às células (Figura 7). A divisão das células é então refeita e são executados novamente os passos 5 a 9. Ao concluir a execução desses passos o algoritmo retorna para o passo 2**,** para continuar a iteração com um novo valor de m_{Cell} e novas grades.

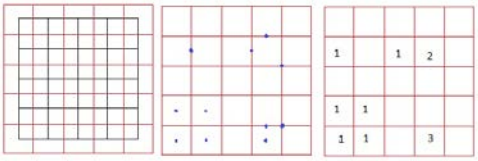

Figura 7 - Deslocamento de Grade e a nova formação.

4. Experimentos Computacionais

O algoritmo proposto no presente trabalho foi implementado em linguagem Java. Além disso, para a execução dos experimentos computacionais foi utilizado um computador dotado de um processador Intel I3 3,1GHz, com 4 GB de memória RAM e Sistema Operacional Ubuntu 12.04.4LTS. Em relação à quantidade de grupos das soluções, foram consideradas válidas as soluções cuja quantidade de grupos pertence ao intervalo [2, $n^{1/2}$], sendo n a quantidade de objetos da instância. A consideração de tal intervalo é uma prática que tem sido comumente adotada nos trabalhos da literatura, conforme [Naldi, 2011] e [Semaan et al., 2012a]. No que diz respeito às instâncias utilizadas, ao todo foram consideradas 63 instâncias conhecidas na literatura, com número de atributos variando entre 2 e 4 e número de objetos variando entre 75 e 1000. As instâncias Ruspini, Maronna e 200DATA são artificiais com 75, 200 e 200 objetos e 4, 4 e 3 grupos, respectivamente, e têm 2 atributos. A instância Iris possui 4 atributos e 150 objetos, onde os objetos representam uma amostra de 3 espécies da planta do gênero íris e os atributos representam medições realizadas. A instância Vowel, descrita em [Wang et al., 2007]**,** possui 2 atributos e 528 objetos. As demais instâncias são artificiais e foram geradas conforme descrito em [Cruz, 2010]. Nesse conjunto, o nome da instância indica suas características, por exemplo, a instância 100p3c1 possui 100 objetos organizados em 3 grupos. A presença do numeral 1 indica que a instância é "comportada", ou seja, possui grupos bem definidos, coesos e bem separados.

Os experimentos foram executados em duas fases: a primeira não levou em consideração o tratamento de ruídos; a segunda buscou identificar e eliminar objetos do tipo ruído. A primeira fase compreende cinco experimentos (Exp1, Exp2, Exp4, Exp5 e Exp6), nos quais o número de iterações foi, respectivamente, 10, 50, 100, 1000 e 5000. Cada iteração foi executada considerando um valor diferente atribuído à variável *fator*, conforme apresentado na seção 3. Os resultados obtidos foram comparados com seis algoritmos da literatura, quais sejam: AECBL e AECBL+KM [Dib et. al., 2014]; Algoritmo CLUES [Wang et. al., 2007]; método sistemático baseado em densidade, MRDBSCAN [Semaan et. al., 2012a] [Semaan et. al., 2013]; algoritmo MSH_ACP [Cruz et. al., 2015a]; algoritmo baseado em grades, apresentado em [Semaan et. al., 2015], denomindado GradesSW.

As Tabelas 1 e 2 apresentam um comparativo entre os resultados produzidos pelos algoritmos da literatura e os produzidos através do método proposto, denominado GradeBL. É possível observar que o método proposto foi capaz de encontrar o maior índice silhueta em todas as instancias classificadas como *comportadas*, conforme definição de [Cruz, 2010] e também utilizado por [Semaan, 2013]. Destaca-se, ainda, que em 9 das 14 instâncias submetidas ao algoritmo GradesSW, que possui metodologia similar à metodologia do algoritmo atualmente proposto, as soluções produzidas pelo método proposto foram superiores, no que diz respeito ao índice de validação adotado. Nos demais casos, os resultados foram equivalentes. Uma análise desses resultados indica a melhoria das silhuetas, mediante a aplicação do novo algoritmo.

Quanto às instancias classificadas como não *comportadas*, os resultados obtidos foram pouco satisfatórios no que diz respeito ao valor da silhueta. Esse fato indicou a necessidade de identificação e classificação de objetos do tipo ruído. Em três instâncias (600p3c1, 800p4c1 e

800p10c1) o algoritmo produziu a solução com a menor silhueta, quando comparado com os demais algoritmos. A melhor silhueta foi obtida apenas nas instâncias 300p4c1 e 500p4c1.

A segunda fase compreende três experimentos (Exp3, Exp7 e Exp8), nos quais o número de iterações foi, respectivamente, 10, 100 e 50. Para o parâmetro de densidade (*l*), foram considerados os valores 5, 10 e 10, respectivamente, em que a densidade representada pela quantidade de iterações no vetor vGrau (vetor com quantidades mínimas distintas de objetos em cada célula), limite inferior para eliminar quantidades de objetos do tipo ruído. As Tabelas 3, 4 e 5 relatam os resultados obtidos. Cada coluna indica a porcentagem de objetos ruído que foram desconsiderados, variando de 0% a 10%. A coluna *Fx* traz o índice silhueta obtido e a coluna *k* mostra a quantidade final de grupos em cada caso.

Tabela 1 – Silhuetas obtidas para instâncias da literatura e instâncias *comportadas*

Instâncias	AECBL	AECBL+KM (melhor)	CLUES	MRDBSCAN	MSH_ACP	GradesSW	GradeBL				
							Exp1	Exp2	Exp4	Exp5	Exp6
Ruspini	**0,738**	**0,738**	**0,738**	**0,738**	**0,738**	-	**0,738**	**0,738**	**0,738**	**0,738**	**0,738**
Iris	0,686	0,686	0,563	**0,687**	0,686	-	-	-	-	-	-
Maronna	**0,575**	**0,575**	**0,575**	0,562	**0,575**	-	0,562	0,562	0,562	0,562	0,562
200DATA	**0,823**	**0,823**	0,462	**0,823**	**0,823**	-	**0,823**	**0,823**	**0,823**	**0,823**	**0,823**
Vowel	0,425	0,425	**0,448**	0,417	0,418	-	0,345	0,368	0,419	0,419	0,419
100p3c	**0,786**	**0,786**	**0,786**	**0,786**	**0,786**	0,062	**0,786**	**0,786**	**0,786**	**0,786**	**0,786**
100p7c	**0,834**	**0,834**	**0,834**	**0,834**	**0,834**	**0,834**	**0,834**	**0,834**	**0,834**	**0,834**	**0,834**
100p10c	**0,834**	**0,834**	**0,834**	0,691	**0,834**	**0,834**	**0,834**	**0,834**	**0,834**	**0,834**	**0,834**
200p4c	**0,773**	**0,773**	**0,773**	**0,773**	**0,773**	0,020	**0,773**	**0,773**	**0,773**	**0,773**	**0,773**
300p3c	**0,766**	**0,766**	0,552	**0,766**	**0,766**	0,026	**0,766**	**0,766**	**0,766**	**0,766**	**0,766**
400p3c	**0,799**	**0,799**	**0,799**	**0,799**	**0,799**	0,301	**0,799**	**0,799**	**0,799**	**0,799**	**0,799**
500p3c	**0,825**	**0,825**	**0,825**	**0,825**	**0,825**	**0,825**	**0,825**	**0,825**	**0,825**	**0,825**	**0,825**
600p15c	**0,781**	**0,781**	**0,753**	**0,781**	**0,781**	**0,781**	**0,781**	**0,781**	**0,781**	**0,781**	**0,781**
700p4c	**0,797**	**0,797**	**0,797**	**0,797**	**0,797**	**0,333**	**0,797**	**0,797**	**0,797**	**0,797**	**0,797**
800p23c	**0,787**	**0,787**	0,739	**0,787**	**0,787**	**0,787**	**0,787**	**0,787**	**0,787**	**0,787**	**0,787**
900p5c	**0,716**	**0,716**	0,613	**0,716**	**0,716**	0,447	**0,716**	**0,716**	**0,716**	**0,716**	**0,716**
900p12c	**0,841**	**0,841**	0,798	**0,841**	**0,841**	0,651	**0,841**	**0,841**	**0,841**	**0,841**	**0,841**
1000p6c	**0,736**	**0,736**	**0,736**	**0,736**	**0,736**	0,399	**0,736**	**0,736**	**0,736**	**0,736**	**0,736**
1000p14c	**0,831**	**0,831**	0,767	0,808	**0,831**	0,605	**0,831**	**0,831**	**0,831**	**0,831**	**0,831**

Tabela 2 - Silhuetas obtidas para instâncias não *comportadas*

Instâncias	AECBL	AECBL+KM (melhor)	CLUES	MRDBSCAN	MSH_ACP	GradeBL				
						Exp1	Exp2	Exp4	Exp5	Exp6
100p3c1	0,580	0,580	**0,597**	0,104	0,580	0,043	0,258	0,290	0,336	0,336
100p5c1	0,696	0,696	**0,703**	0,424	0,696	0,620	0,655	0,654	0,668	0,668
100p7c1	0,491	0,495	**0,551**	-0,012	0,487	0,244	0,378	0,388	0,388	0,388
200p2c1	**0,764**	**0,764**	0,591	0,624	**0,764**	0,599	0,615	0,754	0,758	0,758
200p3c1	**0,680**	**0,680**	0,674	0,648	**0,680**	0,648	0,662	0,662	0,662	0,662
200p4c1	**0,745**	**0,754**	**0,754**	0,622	**0,745**	0,687	0,676	0,685	0,693	0,693
200p7c1	0,576	**0,578**	0,555	0,392	0,570	0,392	0,423	0,423	0,445	0,445
200p12c1	**0,575**	**0,575**	0,562	0,403	0,570	0,430	0,443	0,430	0,455	0,455
300p2c1	0,776	**0,777**	0,490	0,620	0,776	0,610	0,771	0,771	0,774	0,774
300p3c1	0,667	**0,677**	0,592	0,639	**0,677**	0,640	0,658	0,656	0,657	0,657
300p4c1	0,591	0,592	0,592	0,269	0,591	**0,607**	**0,607**	**0,607**	**0,607**	**0,607**
300p6c1	**0,664**	**0,664**	0,579	0,548	0,661	0,337	0,533	0,533	0,536	0,567
400p4c1	0,599	0,607	**0,620**	0,379	0,602	0,127	0,382	0,382	0,382	0,382
400p17c1	0,514	0,514	**0,552**	0,183	0,514	0,169	0,328	0,325	0,460	0,460
500p4c1	0,658	0,660	0,515	0,305	0,660	0,605	0,605	0,569	**0,661**	**0,661**
500p6c1	0,629	0,632	**0,669**	0,494	0,629	0,522	0,545	0,545	0,549	0,549
600p3c1	**0,721**	**0,721**	0,703	0,686	**0,721**	0,397	0,484	0,530	0,532	0,532
700p15c1	0,680	**0,692**	0,660	0,122	0,680	0,121	0,331	0,293	0,441	0,441
800p4c1	0,702	0,705	**0,714**	0,508	0,703	0,291	0,382	0,382	0,382	0,393
800p10c1	0,468	0,478	**0,507**	0,079	0,468	0,065	0,065	0,065	0,065	0,065
800p18c1	0,691	0,691	**0,694**	0,265	0,691	0,086	0,460	0,482	0,482	0,505
1000p5c1	0,639	**0,643**	0,558	0,164	0,639	0,278	0,331	0,340	0,429	0,500
1000p27c1	0,516	0,516	**0,563**	-0,293	0,516	0,006	0,006	0,170	0,170	0,191

Tabela 3 - Silhuetas obtidas para instâncias da literatura e instâncias *comportadas*, aplicando etapa de identificação de objeto do tipo ruído

Instâncias	0%		1%		2%		3%		4%		5%		6%		7%		8%		9%		10%	
	Fx	k	Fx	k	Fx	k	Fx	k	Fx	k	Fx	k	Fx	k	Fx	k	Fx	k	Fx	k	Fx	k
Ruspini	0,738	4	0,738	4	0,738	4	0,753	4	0,760	4	0,760	4	0,764	4	0,764	4	0,778	4	0,778	4	0,778	4
Maronna	0,562	2	0,566	2	0,567	2	0,567	2	0,569	2	0,570	2	0,575	2	0,575	2	0,575	2	0,575	2	0,575	2
200DATA	0,823	3	0,828	3	0,829	3	0,835	3	0,835	3	0,835	3	0,840	3	0,840	3	0,846	3	0,847	3	0,847	3
Vowel	0,368	2	0,369	2	0,369	2	0,369	2	0,369	2	0,369	2	0,369	2	0,369	2	0,369	2	0,369	2	0,369	2
Gauss9	0,171	3	0,250	2	0,250	2	0,250	2	0,250	2	0,250	2	0,250	2	0,250	2	0,250	2	0,250	2	0,250	2
100p3c	0,786	3	0,790	3	0,794	3	0,798	3	0,802	3	0,802	3	0,806	3	0,806	3	0,808	3	0,808	3	0,808	3

Tabela 3 - Silhuetas obtidas para instâncias da literatura e instâncias comportadas, aplicando etapa de identificação de objeto do tipo ruído *(Continuação)*

Instâncias	0%		1%		2%		3%		4%		5%		6%		7%		8%		9%		10%	
	Fx	k	Fx	k	Fx	k	Fx	k	Fx	k	Fx		Fx	k	Fx	k	Fx	k	Fx	k	Fx	k
100p7c	0,834	7	0,835	7	0,835	7	0,837	7	0,838	7	0,838	7	0,838	7	0,838	7	0,842	7	0,842	7	0,842	7
100p10c	0,834	10	0,834	10	0,834	10	0,834	10	0,834	10	0,834	10	0,834	10	0,834	10	0,834	10	0,834	10	0,834	10
200p4c	0,773	4	0,776	4	0,778	4	0,780	4	0,782	4	0,782	4	0,784	4	0,784	4	0,785	4	0,787	4	0,790	4
300p3c	0,766	3	0,769	3	0,770	3	0,772	3	0,772	3	0,775	3	0,775	3	0,778	3	0,780	3	0,780	3	0,780	3
400p3c	0,799	3	0,802	3	0,802	3	0,802	3	0,807	3	0,810	3	0,813	3	0,813	3	0,813	3	0,813	3	0,813	3
500p3c	0,825	3	0,825	3	0,826	3	0,826	3	0,826	3	0,826	3	0,827	3	0,827	3	0,827	3	0,827	3	0,827	3
600p15c	0,781	15	0,784	15	0,787	15	0,787	15	0,792	15	0,792	15	0,793	15	0,793	15	0,795	15	0,806	15	0,806	15
700p4c	0,797	4	0,800	4	0,802	4	0,802	4	0,804	4	0,806	4	0,807	4	0,807	4	0,807	4	0,807	4	0,807	4
800p23c	0,787	23	0,789	23	0,792	23	0,793	23	0,793	23	0,793	23	0,793	23	0,793	23	0,793	23	0,793	23	0,793	23
900p5c	0,716	5	0,719	5	0,720	5	0,721	5	0,721	5	0,721	5	0,723	5	0,723	5	0,724	5	0,724	5	0,724	5
900p12c	0,841	12	0,843	12	0,844	12	0,844	12	0,845	12	0,845	12	0,845	12	0,845	12	0,845	12	0,845	12	0,845	12
1000p6c	0,736	6	0,738	6	0,741	6	0,744	6	0,745	6	0,748	6	0,748	6	0,753	6	0,753	6	0,753	6	0,753	6
1000p14c	0,807	15	0,833	14	0,835	14	0,837	14	0,838	14	0,838	14	0,841	14	0,842	14	0,842	14	0,844	14	0,844	14

A Tabela 3 traz os resultados obtidos a partir de instâncias bem conhecidas da literatura e das instâncias consideradas *comportadas*. Com exceção das instâncias vowel2 e gauss9, que não possuem grupos bem separados, o método obteve bons resultados mesmo sem aplicar a etapa de identificação de ruído. A melhora sutil observada em todas as instâncias pode ser atribuída à remoção de objetos mais dispersos, o que aumenta a coesão dos grupos finais e eleva o índice silhueta. É esperado que, quanto maior a proporção de objetos do tipo ruído a desconsiderar, maior será o valor obtido para o índice silhueta, pois mais coesos serão os objetos remanescentes. E a Tabela 5 apresenta os resultados para as instâncias não *comportadas*. Houve uma melhora nas soluções encontradas, em especial na instância 600p3c1, onde o algoritmo não produziu um bom resultado na primeira fase, sem utilizar a técnica de identificação de ruídos. Ao classificar 6% dos objetos como sendo do tipo ruído**,** foi obtido um resultado superior ao melhor resultado obtido anteriormente e o algoritmo pôde determinar o número ideal de grupos relatado em [Cruz, 2010].

Tabela 4 - Silhuetas obtidas para instâncias não comportadas, aplicando etapa de identificação de ruído

Instâncias	0%		1%		2%		3%		4%		5%		6%		7%		8%		9%		10%	
	Fx	k	Fx	k	Fx	k	Fx	k	Fx	k	Fx	k	Fx	k	Fx	k	Fx	k	Fx	k	Fx	k
100p3c1	0,043	2	0,043	2	0,043	2	0,043	2	0,043	2	0,043	2	0,043	2	0,043	2	0,043	2	0,043	2	0,043	2
100p5c1	0,655	5	0,655	5	0,655	5	0,655	5	0,655	5	0,655	5	0,655	5	0,655	5	0,655	5	0,655	5	0,769	5
100p7c1	0,378	8	0,378	8	0,378	8	0,378	8	0,378	8	0,378	8	0,378	8	0,378	8	0,378	8	0,378	8	0,378	8
200p2c1	0,612	7	0,612	7	0,612	7	0,612	7	0,802	2	0,802	2	0,822	2	0,822	2	0,844	2	0,847	2	0,848	2
200p3c1	0,648	2	0,648	2	0,649	2	0,649	2	0,649	2	0,649	2	0,649	2	0,649	2	0,688	2	0,765	3	0,768	3
200p7c1	0,423	2	0,423	2	0,424	2	0,424	2	0,424	2	0,424	2	0,424	2	0,424	2	0,424	2	0,424	2	0,424	2
200p8c1	0,410	6	0,410	6	0,410	6	0,410	6	0,410	6	0,410	6	0,410	6	0,410	6	0,410	6	0,410	6	0,410	6
200p12c1	0,446	14	0,446	14	0,446	14	0,446	14	0,446	14	0,446	14	0,446	14	0,446	14	0,446	14	0,446	14	0,446	14
300p2c1	0,589	6	0,589	6	0,589	6	0,791	2	0,814	2	0,814	2	0,835	2	0,836	2	0,838	2	0,838	2	0,838	2
300p3c1	0,640	2	0,640	2	0,640	2	0,667	2	0,670	2	0,670	2	0,681	2	0,742	3	0,744	3	0,744	3	0,750	3
300p6c1	0,490	9	0,490	9	0,490	9	0,538	2	0,538	2	0,538	2	0,538	2	0,538	2	0,554	2	0,554	2	0,677	6
300p10c1	0,510	4	0,510	4	0,510	4	0,510	4	0,518	4	0,518	4	0,527	4	0,527	4	0,537	6	0,537	6	0,549	4
300p13c1	0,432	2	0,432	2	0,432	2	0,432	2	0,432	2	0,453	4	0,453	4	0,453	4	0,453	4	0,453	4	0,453	4
400p4c1	0,127	3	0,127	3	0,127	3	0,127	3	0,127	3	0,127	3	0,127	3	0,131	2	0,131	2	0,131	2	0,131	2
400p17c1	0,206	2	0,206	2	0,206	2	0,206	2	0,206	2	0,218	2	0,218	2	0,218	2	0,218	2	0,218	2	0,218	2
500p6c1	0,485	13	0,485	13	0,485	13	0,485	13	0,485	13	0,485	13	0,583	9	0,583	9	0,622	5	0,667	7	0,667	7
500p19c1	0,134	3	0,137	3	0,137	3	0,137	3	0,137	3	0,137	3	0,137	3	0,137	3	0,212	2	0,212	2	0,212	2
600p3c1	0,487	6	0,487	6	0,487	6	0,487	6	0,487	6	0,487	6	0,745	3	0,745	3	0,745	3	0,745	3	0,745	3
800p10c1	0,065	2	0,066	2	0,066	2	0,066	2	0,066	2	0,066	2	0,066	2	0,066	2	0,066	2	0,066	2	0,066	2
800p18c1	0,359	27	0,359	27	0,359	27	0,359	27	0,513	18	0,513	18	0,710	18	0,710	18	0,739	17	0,759	20	0,781	18
1000p5c1	0,356	10	0,356	10	0,356	10	0,356	10	0,420	6	0,420	6	0,420	6	0,570	10	0,571	4	0,711	5	0,712	5
1000p27c1	0,006	2	0,169	2	0,169	2	0,169	2	0,169	2	0,169	2	0,169	2	0,169	2	0,169	2	0,169	2	0,169	2
1100p6c1	0,378	26	0,378	26	0,378	26	0,378	26	0,378	26	0,378	26	0,654	9	0,658	9	0,733	6	0,736	6	0,740	6
1500p6c1	0,177	38	0,177	38	0,177	38	0,177	38	0,275	8	0,462	9	0,566	4	0,673	5	0,677	5	0,701	6	0,701	6
2000p9c1	0,048	43	0,048	43	0,280	2	0,280	2	0,439	3	0,439	3	0,439	3	0,449	3	0,546	18	0,546	18	0,630	9

A Tabela 5 mostra resultados adicionais, obtidos para instâncias não consideradas na primeira fase do experimento. Essas instâncias foram construídas e utilizadas por [Cruz, 2010] [Soares e Ochi, 2004]. Nesse conjunto, destaca-se o resultado obtido para a instância outliers, que possui 150 objetos e 2 grupos. Sem a etapa de identificação de ruídos, o algoritmo identificou 7 grupos, porém, a partir da remoção de 4% de objetos considerados do tipo ruído, o número de grupos relatado em [Soares e Ochi, 2004] foi encontrado. Outro resultado de destaque ocorreu na instância 3dens, onde o índice silhueta foi consideravelmente elevado e o número de grupos foi corrigido de 3 para 2 grupos.

Tabela 5 - Silhuetas obtidas para instâncias adicionais, aplicando etapa de identificação de ruído

Instâncias	0%		1%		2%		3%		4%		5%		6%		7%		8%		9%		10%	
	Fx	k	Fx	k	Fx	k	Fx	k	Fx	k	Fx	k	Fx	k	Fx	k	Fx	k	Fx	k	Fx	k
30p	0,354	5	0,354	5	0,354	5	0,354	5	0,368	4	0,368	4	0,368	4	0,368	4	0,368	4	0,368	4	0,368	4
97p	0,648	4	0,648	4	0,648	4	0,648	4	0,648	4	0,648	4	0,648	4	0,648	4	0,648	4	0,648	4	0,648	4
181p	0,737	6	0,737	6	0,741	6	0,741	6	0,744	6	0,744	6	0,744	6	0,744	6	0,744	6	0,744	6	0,751	6
300p4c	0,750	4	0,752	4	0,754	4	0,755	4	0,755	4	0,755	4	0,755	4	0,755	4	0,755	4	0,767	4	0,767	4
350p5c	0,759	5	0,760	5	0,762	5	0,762	5	0,762	5	0,762	5	0,762	5	0,766	5	0,766	5	0,768	5	0,771	5
450p4c	0,766	4	0,768	4	0,770	4	0,774	4	0,774	4	0,774	4	0,777	4	0,777	4	0,777	4	0,777	4	0,777	4
500p3c	0,825	3	0,825	3	0,826	3	0,826	3	0,826	3	0,826	3	0,826	3	0,826	3	0,826	3	0,826	3	0,826	3
600p3c	0,751	3	0,753	3	0,754	3	0,754	3	0,756	3	0,756	3	0,756	3	0,756	3	0,759	3	0,759	3	0,759	3
900p5c	0,716	5	0,718	5	0,719	5	0,721	5	0,722	5	0,723	5	0,723	5	0,723	5	0,726	5	0,726	5	0,726	5
1000p6c	0,736	6	0,738	6	0,742	6	0,744	6	0,746	6	0,747	6	0,750	6	0,750	6	0,755	6	0,757	6	0,759	6
2000p11c	0,713	11	0,716	11	0,717	11	0,718	11	0,718	11	0,718	11	0,723	11	0,723	11	0,726	11	0,726	11	0,726	11
2face	0,667	2	0,667	2	0,669	2	0,671	2	0,671	2	0,671	2	0,674	2	0,674	2	0,674	2	0,674	2	0,674	2
3dens	0,689	3	0,693	3	0,697	3	0,777	2	0,783	2	0,787	2	0,787	2	0,787	2	0,787	2	0,806	2	0,808	2
Convdensity	0,847	4	0,847	4	0,877	3	0,879	3	0,881	3	0,882	3	0,882	3	0,882	3	0,882	3	0,882	3	0,882	3
Convexo	0,668	3	0,668	3	0,668	3	0,668	3	0,668	3	0,668	3	0,668	3	0,668	3	0,668	3	0,668	3	0,668	3
Face	0,089	13	0,089	13	0,089	13	0,097	2	0,097	2	0,097	2	0,097	2	0,097	2	0,097	2	0,097	2	0,097	2
Moreshapes	0,726	7	0,734	6	0,740	6	0,741	6	0,742	6	0,742	6	0,742	6	0,742	6	0,742	6	0,742	6	0,742	6
Numbers	0,560	9	0,561	9	0,561	9	0,561	9	0,561	9	0,561	9	0,561	9	0,561	9	0,566	9	0,572	9	0,572	9
Numbers2	0,600	10	0,602	10	0,602	10	0,605	10	0,605	10	0,605	10	0,612	10	0,612	10	0,612	10	0,612	10	0,612	10
Outliers	0,607	7	0,607	7	0,607	7	0,607	7	0,634	2	0,637	2	0,637	2	0,637	2	0,637	2	0,637	2	0,637	2
Outliers_ags	0,665	4	0,665	4	0,665	4	0,665	4	0,665	4	0,665	4	0,665	4	0,665	4	0,665	4	0,665	4	0,665	4

5. Conclusões e Trabalhos Futuros

O método apresentado no presente trabalho possibilita solucionar o PAA através de conceito híbrido baseado em grade e densidade, combinado aos procedimentos de deslocamento de grade e identificação de objetos do tipo ruído. Um fator de perturbação é considerado a fim de produzir soluções de maneira não determinística, ampliando o espaço de busca e possibilitando a determinação de soluções melhores. A silhueta é o índice de validação considerado na escolha da solução mais adequada. Os parâmetros relacionados ao problema de agrupamento são determinados automaticamente. Os parâmetros que devem ser fornecidos são o número máximo de iterações desejadas e a proporção estimada de objetos do tipo ruído presentes na instância. Como a definição de tal parâmetro não é trivial, um vetor de valores possíveis deve ser obtido a partir de uma análise exploratória preliminar dos dados, o que implica ampliação do espaço de soluções e favorece a obtenção da solução adequada.

Os resultados obtidos nos experimentos realizados indicam que o método, quando aplicado às instâncias “comportadas”, ou seja, com grupos bem definidos, coesos e bem separados, produz resultados de elevada qualidade no que diz respeito à formação dos grupos. Em todas as instâncias com essa característica, a melhor solução produzida pelo método proposto possui silhueta equivalente à melhor silhueta obtida pelos algoritmos da literatura considerados. Quanto aos resultados produzidos para as instâncias não “comportadas”, é possível verificar a contribuição do recurso de identificação de ruídos na melhoria da qualidade das soluções. Todavia, é fato conhecido da literatura que o índice silhueta, adotado para avaliar a qualidade das soluções produzidas, não é adequado para grupos com formatos arbitrários, como, por exemplo, grupos muito alongados ou aninhados [Naldi, 2011].

A comparação com o método GradesSW aponta a contribuição dos procedimentos de deslocamento de grade e identificação de ruído, implementados no método GradesBL. No que diz respeito à comparação com os métodos AECBL e AECBL+KM, embora os resultados não tenham sido superiores em termos valor do índice silhueta da solução final, o método apresentado é de fácil implementação e custo computacional controlado, uma vez que o número de operações do algoritmo não é uma função do número de objetos na instância, mas do número de células da grade, parâmetro que pode ser especificado em função da capacidade de processamento disponível. Tal propriedade é importante à manipulação de instâncias com nº elevado de objetos.

Destacamos alguns pontos para futuro aprimoramento: (i) estudar outros índices em critérios relativos, de forma a adotar um que seja mais adequado às soluções com características específicas de grupos baseados em densidade; (ii) realizar experimentos computacionais com novos conjuntos de instâncias com diferentes características, a fim de verificar a versatilidade do método e (iii) investigar novas formas de obter a variável m_{cell} de maneira automática; (iv) comparar o desempenho do método em termos de custo computacional.

Referências

Bezdeck, J. C., Hathaway, R. J. (2002). VAT: A tool for visual assessment of (cluster) tendency, in: Proceedings of IJCNN, IEEE Press, Piscataway, NJ, pp. 2225–2230.

Cruz, M.D. (2010). O Problema de Clusterização Automática (Tese de Doutorado) – UFRJ, Rio de Janeiro.

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

Cruz, M. D. e Ochi, L. (2015b). Hybrid Method Using Evolutionary Algorithm and a Linear Integer Model to Solve the Automatic Clustering Problem. *Learning & Nonlinear Models*, v. 13, n. 2.

Ester, M., Kriegel, H. P., Sander, J. e Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In: *International Conference On Knowledge Discovery And Data Mining (KDD'96)*, Portland, Oregon, p. 226-231.

Han, J. e Kamber, M. Cluster Analysis. (2006). Data Mining: Concepts and Techniques. 2nd ed. New York: Academic Press.

Hruschka, E. R., Campello, R. J. G. B. e Castro, L. N. (2004a). Evolutionary algorithms for clustering gene-expression data. In: *International Conference On Data Mining*, Brighton/England, p. 403-406.

Hruschka, E. R. e Ebecken, N. F. F. (2001). A Genetic algorithm for cluster analysis. IEEE Transactions on Evolutionary Computation.

Hruschka, E. R., Ebecken, N. F. F. (2003) A genetic algorithm for cluster analysis. Intelligent Data Analysis, v. 7, n. 1, p. 15-25.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, v. 31, n. 8, p. 651-666.

Johnson A. R. e Wichern D. W. (2002). Applied Multivariate Statistical Analysis. Prentice Hall. Fifth Edition.

Liu, R., Jiao, L., Zhang, X. e Li, Y. (2012). Gene transposon based clone selection algorithm for automatic clustering. *Information Sciences*, v. 204, p. 1-22.

Ma, E. W. M. e Chow, T. W. S. (2004). A new shifting grid clustering algorithm. *Pattern Recognition*, v. 37, n. 3, p. 503-514.

Naldi, C. N. (2011). Técnicas de Combinação para Agrupamento Centralizado e Distribuído de Dados. (Tese de Doutorado) – Universidade de São Paulo.

Nascimento, M. C. V., Toledo, F. M. B e Carvalho, A. C. P. L. F. (2010). Investigation of a new GRASP-based clustering algorithm applied to biological data. *Computers & Operations Research*, v.37, n.8, Special Issue, p.1381-1388.

Oliveira, C. (2007). Edacluster: Um Algoritmo Evolucionário para Análise de Agrupamentos Baseados em Densidade e Grade. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal do Pará.

Rakesh, A., Johanners, G., Dimitrios, G. e Prabhakar, R. (1999). Automatic subspace clustering of high dimensional data for data mining applications. In: ACM SIGMOD, p. 94-105.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, v. 20, p. 53-65.

Semaan, G. S. (2013). Algoritmos para o Problema de Agrupamento Automático. (Tese de Doutorado) – Instituto de Computação, Universidade Federal Fluminense, Rio de Janeiro.

Semaan, G. S., Brito, J. A. M. e Ochi, L. S. (2012b). Um Método Hierárquico para a Determinação do Número Ideal de Grupos. *Revista Brasileira de Estatística*, v. 73, p. 81-113.

Semaan, G. S., Cruz, M. D., Brito, J. A. M. e Ochi, L.S. (2012a). Proposta de um método de classificação baseado em densidade para a determinação do número ideal de grupos em problemas de clusterização. *Learning & Nonlinear Models*, v. 10, n. 4.

Semaan, G. S., Vasconcelos, R. B., Brito, J. A. M. e Ochi, L. S. (2015). Proposta de um Método Baseado em Densidade e Grade para o Problema de Agrupamento Automático. Anais do XVII Simpósio de Pesquisa Operacional e Logística da Marinha (SPOLM). São Paulo: Editora Edgard Blücher. p. 153.

Soares, S. S. R. F. e Ochi, L. S. (2004). Um Algoritmo Evolutivo com Reconexão de Caminhos para o Problema de Clusterização Automática. In: XII LATIN IBERO AMERICAN CONGRESS ON OPERATIONS RESEARCH.

Steinbach, M., Karypis, G. e Kumar, V. (2000). A comparison of document clustering techniques. Minneapolis: University of Minnesota. (Technical Report 34).

Tan, P., Steinbach, M. e Kumar, V. (2009). Introdução ao Data Mining. Editora Ciência Moderna.

Tseng, L. e Yang, S. B. (2001). A genetic approach to the automatic clustering problem. *Pattern Recognition*, v. 34.

Wang, X., Qiu, W. e Zamar, R. H. (2007). CLUES: A non-parametric clustering method based on local shrinking. *Computational Statistics & Data Analysis*, v. 52.

Zalik, K. R. (2008). An efficient k'-means clustering algorithm. *Pattern Recognition Letters*, v. 29, n. 9, p. 1385-1391.