UMA HEURÍSTICA BASEADA EM DENSIDADE PARA O PROBLEMA DE AGRUPAMENTO AUTOMÁTICO

Gustavo Silva Semaan[1], Augusto César Fadel[2], José André de Moura Brito[3], Luiz Satoru Ochi[4]

[1] Instituto do Noroeste Fluminense de Educação Superior - Universidade Federal Fluminense (INFES - UFF)
[2] Fundação Instituto Brasileiro de Geografia e Estatística (IBGE)
[3] Escola Nacional de Ciências Estatísticas - Instituto Brasileiro de Geografia e Estatística (ENCE - IBGE)
[4] Instituto de Computação - Universidade Federal Fluminense (IC - UFF)

{*gustavosemaan@id.uff.br, augustofadel@gmail.com, jose.m.brito@ibge.gov.br, satoru@ic.uff.br*}

Resumo

A análise de agrupamentos agrega vários métodos que visam identificar grupos dentro de um conjunto de dados. Este artigo apresenta novas heurísticas baseadas na metaheurística Busca Local Iterada para resolver o Problema de Agrupamento Automático, qual seja o problema de determinar o número ideal de grupos para uma base dados. Para tal, em uma das fases da aplicação desta heurística, foi utilizado o índice silhueta, que combina conceitos de coesão e separação e é considerado pelas heurísticas propostas para avaliar a qualidade das soluções. De acordo com os experimentos computacionais reportados neste trabalho, verifica-se que a nova heurística ILS-DBSCAN é muito eficiente no que concerne ao tempo de processamento e muito eficaz quanto à qualidade das soluções obtidas, quando comparado com outros métodos da literatura. Em geral, os resultados desta nova heurística foram superiores aos resultados relatados na literatura. Dessa maneira o ILS-DBSCAN apresenta-se como um algoritmo promissor para a resolução do problema abordado.

Palavras-chave: Agrupamento Automático; Densidade; Índice Silhueta; Busca Local Iterada.

Abstract

The cluster analysis has several methods that aim to identify groups within a dataset. This paper presents new heuristics based on Iterated Local Search metaheuristic to solve the Automatic Clustering Problem. The Silhouette index combines both cohesion and separation concepts, and it is considered by proposed heuristics to evaluate the solutions. According the computational experiments, the effectiveness and efficiency of the ILS-DBSCAN are reflected in substantially lower runtimes and in the solutions' quality, which are competitive with best results reported in the literature. This way, the use of the ILS-DBSCAN is a new promising way to solve the problem.

Keywords: *Automatic Clustering; Density; Silhouette Index; Iterated Local Search.*

1. Introdução

A análise de agrupamentos é uma técnica de análise multivariada [Hair et al, 2009] que agrega um conjunto de métodos que são aplicados à determinação de grupos a partir de um conjunto de objetos definidos por certas características (atributos). Basicamente, o objetivo dessa análise é produzir grupos que sejam formados por objetos (características) semelhantes e que possam refletir a forma como os dados são estruturados. Nesse sentido**,** busca-se maximizar a similaridade (homogeneidade) entre os objetos de um mesmo grupo e minimizar a similaridade entre objetos de grupos distintos [Han & Kamber, 2006] [Larose, 2005] [Goldschimidt & Passos, 2005] [Naldi, 2011].

Formalmente, o problema clássico de agrupamento (PA) pode ser definido da seguinte maneira: dado um conjunto X formado por n objetos, tal que $X=\{x_1,x_2,...,x_i,...,x_n\}$, com cada objeto $x_i \in X$ possuindo f atributos (dimensões ou 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. Além disso, devem ser respeitadas as restrições concernentes a cada problema

particular abordado [Han & Kamber, 2006] [Baum, 1986] [Hruschka & Ebecken, 2001] [Dias & Ochi, 2003].

As Equações 1, 2 e 3 representam o conjunto de restrições que definem o problema clássico de agrupamento, e determinam, respectivamente: (1) que o conjunto *X* corresponde à união dos objetos dos grupos; (2) cada objeto pertence a exatamente um grupo; (3) todos os grupos possuem ao menos um objeto.

$$\bigcup_{j=1}^{k} C_j = X \tag{1}$$

$$C_i \bigcap C_l = \varnothing \text{ i,l=1, .., k e i } \neq \text{ l} \tag{2}$$

$$C_j \neq \varnothing \text{ i=1, .., k} \tag{3}$$

Para o PA o número de soluções possíveis, ou seja, o total de maneiras em que os *n* objetos podem ser agrupados, considerando um número fixo de *k* grupos, é dado pelo número de *Stirling* (*NS*) de segundo tipo [Johnson e Wichern, 2001] (Equação 4). Para o Problema de Agrupamento Automático (PAA) o número de soluções possíveis é ainda maior e corresponde ao somatório da Equação 4 para o número de grupos variando no intervalo [1,k_{max}], sendo k_{max} o número máximo de grupos (Equação 5).

$$NS(n,k) = \frac{1}{k!}\sum_{j=0}^{k}(-1)^{j-i}\binom{k}{j}j^n \tag{4}$$

$$NS(n) = \sum_{j=1}^{k_{\max}} NS(n, j) \tag{5}$$

Para ilustrar a ordem de grandeza da quantidade de soluções possíveis, no caso de 10 objetos (n=10) e 3 grupos (k=3) devem ser consideradas 9.330 soluções (Equação 4). Mas**,** considerando apenas dobro de objetos (n=20 e k=3)**,** o número de soluções possíveis aumenta para 580.606.446.

No PAA a quantidade de soluções possíveis cresce exponencialmente com o aumento da quantidade de objetos (n), o que torna proibitiva a obtenção da solução ótima global mediante a aplicação de procedimentos de enumeração exaustiva. A Figura 1 ilustra tal situação através de um comparativo do número de *Stirling* acumulado para dois conjuntos de dados, um com 10 objetos e outro com 15 objetos.

Conforme [Kumar *et al.*, 2009], as últimas décadas têm sido marcadas pelo desenvolvimento de diversos algoritmos de agrupamento, que são aplicados nos mais diversos domínios como: marketing, economia, ecologia, estatística, pesquisas médicas, ciências políticas etc. Entretanto, torna-se necessário o desenvolvimento de algoritmos específicos conforme a aplicação, uma vez que nenhum algoritmo trabalha com todos os tipos de dados, características dos grupos e contempla todas as especificidades dos problemas.

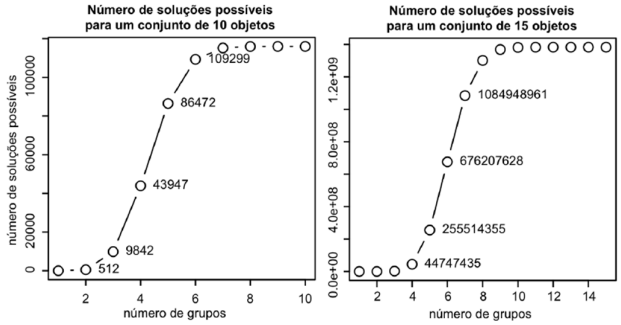

Figura 1. Soluções possíveis por quantidade de grupos para 10 e 15 objetos.

Os métodos baseados em densidade, por exemplo, permitem a identificação de grupos de formatos arbitrários. Nesse contexto esses métodos classificam como grupos as regiões onde há o maior número de elementos (objetos) no espaço de dados que são, naturalmente, separados pelas áreas de baixa densidade [Han & Kamber, 2006]. Um clássico algoritmo da literatura que considera conceitos de densidade é o algoritmo DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) [Ester *et al.*, 1996].

O presente trabalho está dividido em cinco 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 os algoritmos heurísticos propostos nesse trabalho. A Seção 4 traz os resultados computacionais obtidos, enquanto a Seção 5 apresenta as conclusões do trabalho e sugere trabalhos futuros.

2. Revisão da Literatura

Conforme foi apresentado na seção 1, no PAA a identificação (ou determinação) da quantidade de grupos (*k*) faz parte do problema. Segundo [Kumar *et al.*, 2009], a identificação de *k* talvez seja um dos problemas de seleção de parâmetros de entrada mais conhecidos. Neste sentido, a presente seção apresenta uma descrição de diversos algoritmos e métodos da literatura para a resolução do PAA.

Os índices relativos são frequentemente utilizados com o objetivo de avaliar soluções provenientes de diversos algoritmos e métodos que consideram diferentes abordagens. Esses índices não têm a propriedade de monotonicidade, ou seja, não são afetados pelo aumento ou pela redução do número de grupos da solução. A seção 2.1 apresenta o índice relativo Silhueta, que foi utilizando em algumas propostas relatadas nessa seção bem como no algoritmo proposto nesse trabalho [Rousseeuw, 1987] [Cruz, 2010] [Naldi, 2011].

Duas abordagens sistemáticas que utilizam o clássico algoritmo da literatura *k-means* foram propostas por [Naldi, 2011]: o *MRk-means* (do inglês *Multiple Runs of k-means*) e o *OMRk-means* (do inglês *Ordered MRk-means*). Elas atuam na maximização do índice silhueta, consideram conceitos de agrupamento baseados em protótipos, e consistem em múltiplas execuções do *k-means*. A distinção entre elas é que enquanto na *MRk-means* o valor do parâmetro *k* é obtido de maneira aleatória dentro de um intervalo de inteiros, na *OMRk-means* todos os valores inteiros pertencentes ao referido intervalo são considerados.

A abordagem sistemática baseada em densidade MRDBSCAN (do inglês *Multiple Runs of DBSCAN)* foi proposta por [Semaan et al., 2012] e considera o clássico algoritmo DBSCAN [Ester et al., 1996]. Ela também atua na maximização do índice silhueta para a resolução do PAA e realiza múltiplas execuções do DBSCAN. Uma vez que o DBSCAN é determinístico, a técnica *Distk* [Kumar et al., 2009] para calibração de seus parâmetros de entrada foi utilizada e diversos conjuntos de parâmetros de entrada são usados com o objetivo de obter diferentes soluções.

Diversos trabalhos da literatura trazem propostas de algoritmos baseados em meta-heurísticas para a resolução do PAA. Estes algoritmos têm por objetivo encontrar um número ideal de grupos e a sua solução correspondente. [Soares and Ochi, 2004] [Cruz, 2010] [Hruschka and Ebecken, 2003] [Hruschka et al., 2004] [Hruschka *et al.*, 2006] [Ma *et al.*, 2006] [Alves *et al.* 2006] [Tseng & Yang, 2001] [Naldi & Carvalho, 2007].

Em [Maulik & Bandyopadhyay, 2000], assim como em diversos trabalhos da literatura como [Cruz, 2010] e [Semaan, 2013], o algoritmo de agrupamento *k-means* [MacQueen, 1967] foi utilizado para formação de soluções, em especial, em conjunto a procedimentos de busca local. Na fase de avaliação dos cromossomos, o cálculo da aptidão (*fitness*) no AG corresponde ao cálculo das distâncias entre os centroides codificados no cromossomo e cada objeto. Quando um determinado objeto está mais próximo do centroide de outro grupo, ele deve ser migrado para o grupo do centroide mais próximo, assim como ocorre no *k-means*.

Em [Garai & Chaudhuri, 2004] os dados são decompostos em alguns grupos fragmentados (grupos parciais). O Algoritmo Genético é aplicado sobre esses fragmentos e realiza uniões entre grupos adjacentes com base em uma análise de vizinhanças. Já em [Oliveira, 2007] é proposto o Algoritmo Evolutivo EDACluster baseado no algoritmo EDA (Algoritmo de Estimativa de Distribuição, do inglês *Estimation of Distribution Algorithms*) [Larrañaga and Lozano, 2002]. O EDACluster considera uma metodologia baseada nos métodos de densidade e grade, e não utiliza os operadores de cruzamento e mutação.

O algoritmo baseado em grade (e em densidade) CLIQUE [Rakesh et al., 1999] particiona o conjunto de dados em subespaços (grade de células) para encontrar agrupamentos suficientemente densos, ou seja, com grande concentração de objetos. Cada grade contém um conjunto de dados, separando os valores contínuos de cada atributo em um número de intervalos discretos. Por fim, cada objeto é atribuído a uma célula a qual seu intervalo contém o valor original do objeto. Os agrupamentos são formados a partir da junção de células densas adjacentes [Oliveira, 2007].

Em [Semaan *et al.*, 2014] foi apresentado um método baseado em densidade e grade que também atua na maximização o índice silhueta. Já [Semaan *et al.*, 2013] propõe um método baseado em combinação de soluções com coassociação para o PAA. Nesse artigo, em particular, foram utilizados conceitos de comitê de agrupamentos (do inglês *cluster ensembles*).

2.1. Índice Silhueta

O Índice Silhueta foi proposto por [Rousseeuw, 1987] e possibilita avaliação da qualidade das soluções com base na proximidade entre os objetos de determinado grupo e na distância desses objetos ao seu grupo mais próximo. Ele é calculado para cada objeto, sendo possível identificar se o objeto está alocado ao grupo mais adequado. Em síntese, esse índice combina as ideias de coesão e de separação. Os quatro passos a seguir explicam, sucintamente, como calculá-lo:

1. Nesse trabalho d_{ij} (Equação 7) corresponde à distância euclidiana entre os objetos x_i e x_j, e f é a quantidade de atributos dos objetos. Para cada objeto x_i calcula-se a sua distância média $a(x_i)$ (Equação 8) em relação aos demais objetos do mesmo grupo. Na Equação 8, $|C_w|$ representa a quantidade de objetos do grupo C_w ao qual o objeto x_i pertence.
2. A Equação 9 apresenta a distância entre o objeto x_i e os objetos de outro grupo C_t, sendo $|C_t|$ correspondente à 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 e $b(x_i)$ (Equação 10) armazena a distância média em relação a todos os objetos do grupo mais próximo.
3. O coeficiente silhueta de cada objeto x_i ($s(x_i)$) é obtido mediante a aplicação da Equação 11.
4. O cálculo da silhueta de uma solução $\pi=\{C_1,...,C_k\}$ é a média das silhuetas dos objetos, conforme apresenta a Equação 6, em que n é a quantidade de objetos da solução. Essa função deve ser maximizada.

$$\max Silhueta(\pi) = \frac{1}{n}\sum_{i=1}^{n} s(x_i) \tag{6}$$

$$d_{ij} = \sqrt{\sum_{q=1}^{f} (x_i^q - x_j^q)^2} \tag{7}$$

$$a(x_i) = \frac{1}{|C_w| - 1} \sum d_{ij} \quad \forall x_j \neq x_i, \quad x_j \in C_w \tag{8}$$

$$d(x_i, C_t) = \frac{1}{|C_t|} \sum d_{ij} \quad \forall x_j \in C_t \tag{9}$$

$$b(x_i) = \min d(x_i, C_t) \quad C_t \neq C_w \quad C_t \in C \tag{10}$$

$$s(x_i) = \frac{b(x_i) - a(x_i)}{\max\{b(x_i), a(x_i)\}} \tag{11}$$

Os objetos que têm os valores da silhueta correspondentes a valores positivos e próximos de 1 estão bem localizados em seu grupo, enquanto valores negativos indicam que o objeto está mais próximo de outro(s) grupo(s). Ou seja, este índice permite avaliar a pertinência dos objetos em relação aos seus respectivos grupos.

3. Heurísticas para o Problema de Agrupamento Automático

Os algoritmos apresentados na presente 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*. O trabalho de [Cruz, 2010] justifica a construção desses grupos como um procedimento de pré-processamento que tem por finalidade a redução da cardinalidade da instância. Nesse sentido, o procedimento de construção desses grupos trabalha com conceitos de densidade, sendo composto por duas etapas, a saber: *Formação* e *União (ou Junção) de Grupos Parciais*.

Nos algoritmos *CLUSTERING* [Tseng and Yang 2001], *AEC* e *SAPCA* [Soares 2004b] o conceito de *Grupos Parciais* refere-se aos componentes conexos de *Grafos* construídos utilizando diferentes técnicas. Enquanto no algoritmo *CLUSTERING* esses componentes são construídos com base na formação do *Grafo de Vizinhança G*, nos algoritmos *AEC* e *SAPCA* é utilizado um *Grafo de Vizinhança Relativa*. No algoritmo *CLUES* [Wang 2007] o procedimento responsável pela formação dos *Grupos Parciais* é denominado *Encolhimento*, sendo considerados nesse caso os *k*-vizinhos mais próximos a cada objeto.

Conforme o contexto de cada algoritmo, a *Redução da cardinalidade* citada por [Cruz, 2010] significa que cada *Grupo Parcial* será manipulado pelos demais procedimentos dos algoritmos como um objeto, com eventuais exceções. Os procedimentos aplicados à avaliação da qualidade de cada solução (Índice Silhueta), por exemplo, devem considerar os objetos separadamente para que o cálculo seja realizado corretamente.

3.1 – ALGORITMOS HEURÍSTICOS PROPOSTOS

O presente trabalho propõe heurísticas baseadas na *metaheurística Busca Local Iterada* (ILS, do inglês *Iterated Local Search*) [Lourenco et al. 2010] (Algoritmo 1). Em especial, foram incorporados a essas novas heurísticas os procedimentos utilizados pelo eficiente Algoritmo Evolutivo proposto por [Cruz, 2010]. Essa metaheurística atua em um subespaço de soluções ótimas definido por procedimentos de otimização, e baseia-se na simples ideia de que uma solução ótima local pode ser melhorada por meio da aplicação de um procedimento de busca local em uma nova solução, sendo essa resultante da aplicação de um procedimento de perturbação.

```
ILS()
   s₀ ← GeraSoluçãoInicial();
   s ← BuscaLocal(s₀);
   Enquanto não(critérios de parada) faça
      s' ← Perturbação(s);
      s" ← Busca Local(s');
      s ← Criterio de Aceitação(s, s");
   Fim-Enquanto;
   Retorne s;
Fim ILS();
```

Algoritmo 1. Pseudo-código da Busca Local Iterada.

No presente trabalho 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.

3.1.1 – Características

- **Procedimentos de Construção:** ambos os procedimentos construtivos utilizados consideram conceitos de agrupamentos baseados em densidade. São eles:
  - **DBSCAN**: procedimento que implementa uma variante do algoritmo DBSCAN [Ester et al., 1996] [Semaan *et al.*, 2012] em que todos os objetos devem pertencer a um grupo, inclusive os objetos classificados como *ruídos*.
  - **FJGP** (Formação e Junção de Grupos Parciais): procedimento para construção de soluções iniciais utilizado pelo algoritmo AECBL1[Cruz, 2010]. 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 é atribuído a cada grupo parcial. O *bit* 1 indica 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*.
- **Avaliação**: foi utilizado o *Índice Silhueta Tradicional* (seção 2.1).
- **Buscas Locais**:
  - ***Inversão Individual*:** atua na manipulação de *bits* do vetor binário *auxB*. Consiste, basicamente, em a cada posição (*bit*) *i* desse vetor ($auxB_i$), 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. Tendo em vista que esse procedimento realiza a inversão de um *Grupo Parcial* por vez, e que essa inversão resulta no incremento ou decremento da quantidade de *Grupos Parciais* classificados como "*pais*", 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.
  - ***Troca entre Pares**:* também atua na manipulação de *bits* do vetor binário *auxB*. Esse procedimento realiza uma busca em que para cada posição *i* do vetor *auxB*, seu valor $auxB_i$ é trocado com o valor de $auxB_j$, quando $i \neq j$ e $auxB_i \neq auxB_j$. Uma vez que a troca é realizada apenas quando os valores no vetor binário forem diferentes, além de tratar-se de uma busca intensiva, a quantidade de grupos da solução permanece inalterada ($k + 1 - 1 = k$).

- ***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 *bits*. Com o objetivo de percorrer novos caminhos de busca um novo procedimento foi proposto apenas tornando aleatória a seleção da posição 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.*

- **Filtro**: são formadas *q* soluções iniciais, em que *q* é submetido como parâmetro. Essas soluções são refinadas 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.
- **União de Grupos Parciais**: foi utilizado o procedimento otimizado para o recálculo de centroides apresentado por [Soares 2004a] com o objetivo de reduzir o custo computacional de $O(n.f)$ para $O(f)$, em que n é a quantidade de objetos da instância e f a quantidade de atributos dos objetos.
- **Perturbação**: inversão aleatória de *bits* do vetor *auxB* e aplicação do procedimento de formação de soluções.
- **Critérios de Parada**: alcance ao alvo, ou seja, 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**,** foram propostos 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**.

4. Experimentos Computacionais

Para a realização dos experimentos computacionais, os algoritmos propostos foram implementados em Linguagem C++, utilizando o paradigma orientação a objetos. É uma prática comum em PAA considerar k no intervalo de inteiros $[2,k_{max}]$, em que $k_{max}=\sqrt{n}$ [Pal and Bezdek, 1995][Pakhira et al., 2005][Campello et al., 2009]. Em [Han and Kamber, 2012], 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 a ambos os intervalos apresentados na literatura, neste trabalho foi considerado *k* no intervalo de inteiros $[2, \sqrt{n}]$. Foram utilizadas as 82 instâncias dos três conjuntos de dados (DS1, DS2 e DS3) apresentados na seção 4.1.

4.1 – Instâncias Utilizadas

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 variando entre *30* e *2000*, a quantidade de dimensões (atributos) entre *2* e *13* e diferentes características relacionadas, por exemplo, com a coesão, separação, os formatos e as 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

[Fisher 1936] [Ruspini 1970] [Maronna and Jacovkis 1974][Hastie et al. 2001] [Naldi 2011] [Wang 2007].

O segundo conjunto (DS2) contempla 51 instâncias que foram construídas por [Cruz, 2010]. 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 [Cruz, 2010]). A Figura 2 apresenta uma instância *100p7c* (nome com final "*c*") considerada "comportada" com 100 objetos e 7 grupos, e 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 [Soares 2004a][Soares 2004b]. Essas instâncias possuem a quantidade de objetos entre *30* e *2000* e duas dimensões.

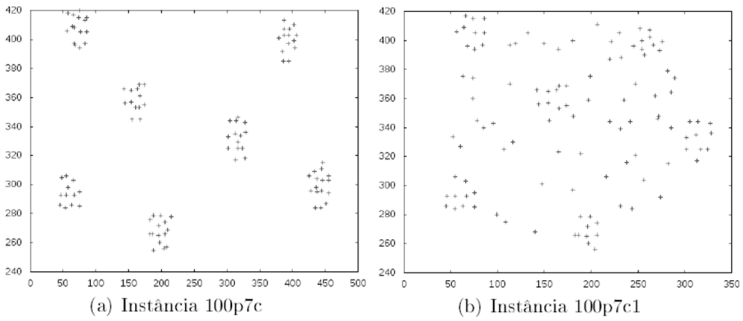

Figura 2. Instâncias consideradas "comportadas" (100p7c) e não comportadas (100p7c1).

4.2 – Experimentos com Heurísticas Propostas

O primeiro experimento consiste em executar o algoritmo com os melhores resultados relatados na literatura. Trata-se do algoritmo evolutivo com busca local (AECBL1) apresentado em [Cruz, 2010] que, em experimentos preliminares realizados nesse trabalho, foi comparado aos algoritmos CLUSTERING [Tseng and Yang 2001], SAPCA e AEC-RC [Soares 2004b] e MRDBSCAN [Semaan, 2013].

As implementações do algoritmo AECBL1 foram gentilmente cedidas pelo autor e também serviram de base para as análises e pesquisas. Para a realização dos experimentos foram utilizados os mesmos parâmetros relatados na literatura. O AECBL1 foi executado 5 vezes, cada uma com 50 iterações para a obtenção dos dados a seguir:

- ***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 propostos ILS-FJGP e ILS-DBSCAN utilizando os resultados obtidos no experimento anterior, realizado com o AECBL1. Nos experimentos apresentados na presente seção, os critérios de parada utilizados foram: alcançar o maior valor de silhueta obtido pelo AECBL1 ou processar até 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 refinamento. Nesse

sentido, as gerações do Algoritmo Evolutivo AECBL1 utilizam-se de buscas locais e de um procedimento de reconexão por caminhos [Reeves 2010], 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 [Semaan et al., 2012], observou-se que para as instâncias denominadas *comportadas* o método obteve 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 2 grupos.

O ILS-DBSCAN, assim como o método sistemático MRDBSCAN, utiliza como procedimento de construção um algoritmo baseado no DBSCAN. A heurística, entretanto, não utiliza a técnica *Distk* [Kumar et al., 2009] 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 o a variável *z* (valor fracionário aleatório no intervalo [1.5, 4.5]). Já o parâmetro *qtdeObjetos* corresponde a um valor inteiro selecionado aleatoriamente no intervalo [2,5].

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*.

Em relação aos comparativos dos resultados realizados entre 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 a expensas de um baixo custo computacional. Para isso, a Tabela 1 apresenta percentuais em que os algoritmos alcançaram as melhores soluções. É possível observar que o algoritmo ILS-FJGP obteve os menores percentuais em todos os comparativos apresentados.

Tabela 1. Alcance à melhor solução obtida por algoritmo.

Instância	AECBL1	ILS-FJGP	ILS-DBSCAN
Todas	**96,30%**	4,94%	50,62%
DS2	**96,00%**	4,00%	46,00%
DS2 - Comportadas	88,89%	11,11%	**100%**

Ainda com base na Tabela 1, o AECBL1 destacou-se obtendo 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 unidades, enquanto o AECBL1 obteve em cerca de 89% das instâncias.

Com o objetivo de analisar também os custos computacionais (tempo de processamento) do algoritmo ILS-DBSCAN para as instâncias comportadas, a Tabela 2 apresenta: o ***Gap total***, razão entre o tempo total de processamento do AECBL1 e o ILS-DBSCAN e o ***Gap Iteração***, razão entre o tempo em que foi obtida a melhor solução do AECBL1 e o ILS-DBSCAN.

Em relação ao **Gap Iteração**, o ILS-DBSCAN foi, na média, abaixo de 0,9% do tempo consumido pelo AECBL1. No pior caso, o tempo computacional do algoritmo foi de apenas cerca de 3%. Quando a análise realiza considera o ***Gap Total*** (tempo total de processamento), a diferença é ainda maior, em que o maior percentual foi de apenas 0,57%.

Conforme os resultados apresentados pelas Tabelas 1 e 2 o algoritmo ILS-DBSCAN obteve os melhores resultados, empatando ou melhorando 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 AECBL.

Tabela 2. 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%*

Com o objetivo de verificar a eficiência do ILS-DBSCAN foi realizado um experimento baseado na *Análise de Probabilidade Empírica* (TTTPlots, do inglês *Time-To-Target Plots*) proposta por [Aiex et al. 2007]. 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).

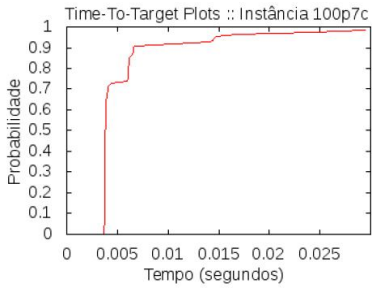

Figura 3. Grático TTTPlots para a Instância 100p7c :: Alvo Silhueta=0,83

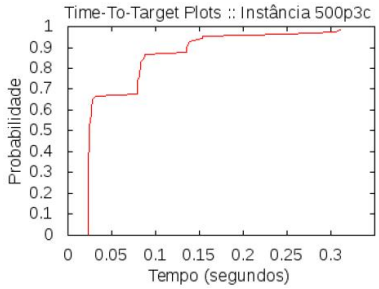

Figura 4. Grático TTTPlots para a Instância 500p3c :: Alvo Silhueta=0,82

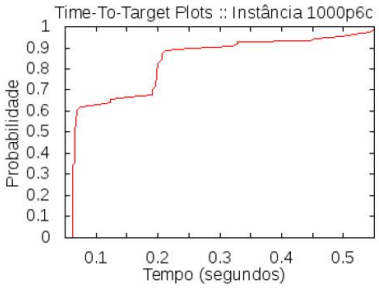

Figura 5. Grático TTTPlots para a Instância 1000p6c :: Alvo Silhueta=0,73

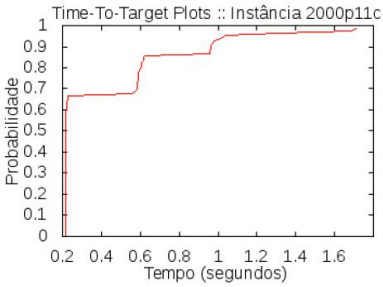

Figura 6. Grático TTTPlots para a Instância 2000p11c :: Alvo Silhueta=0,71

A Figura 3 apresenta o gráfico *TTTPlots* para a Instância 100p7c, cujo valor do alvo (silhueta) utilizado foi 0,83. É possível observar que em menos de 0,005 segundos a probabilidade de o algoritmo alcançar o alvo foi superior a 70%. Além disso, em cerca de 0,03 segundos o alvo foi alcançado em todos os experimentos. O mesmo experimento foi realizado nas instâncias 500p3c, 1000p6c e 2000p11c, conforme ilustram as Figuras 4, 5 e 6, respectivamente.

5. Conclusões e Trabalhos Futuros

Com o objetivo de resolver o PAA, o presente trabalho propôs dois Algoritmos Heurísticos baseados na metaheurística *Busca Local Iterada*: o ILS-FJGP e o ILS-DBSCAN

A heurística ILS-FJGP não obteve resultados satisfatórios, e 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 (MRDBSCAN), 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. Além disso, quando o número de grupos não foi o mesmo do melhor resultado apresentado na literatura, a diferença máxima foi de apenas 2 unidades. Com base nos resultados relatados, um novo 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 obtendo 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.

O algoritmo ILS-DBSCAN, mesmo quando comparado com o algoritmo que apresenta os melhores resultados na literatura, teve desempenho superior tanto na qualidade das soluções produzidas quanto na substancial redução do tempo de processamento, considerando o subconjunto de instâncias "comportadas". Não obstante, de forma a reforçar ainda mais esta análise, seguem algumas propostas para trabalhos futuros:

- Realizar mais experimentos computacionais com novos conjuntos de instâncias, que possuem mais atributos e diferentes características.
- O índice Silhueta não produz bons resultados para grupos com formatos arbitrários. Estudar novas medidas relativas ou até mesmo propor uma nova medida que seja mais adequada às soluções com características específicas de grupos baseados em densidade.
- Realizar pesquisas sobre *Tendência à Formação de Agrupamentos*, como o uso da *Estatística de Hopkins* [Banerjee 2004] com o objetivo de identificar situações em que determinadas heurísticas são mais adequadas. Por exemplo, nesse trabalho o ILS-DBSCAN apresentou-se como mais adequado (eficiente e eficaz) em relação ao subconjunto de instâncias consideradas comportadas.

Agradecimentos

Os autores agradecem ao apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), da Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ) e do Instituto de Computação da Universidade Federal Fluminense.

Referências

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.

Alves, V., R. Campello, & E. Hruschka (2006). *Towards a fast evolutionary algorithm for clustering*. *In IEEE Congress on Evolutionary Computation*, 2006, Vancouver, Canada, pp. 1776–1783.

Banerjee, A. *Validating clusters using the hopkins statistic*. *IEEE International Conference on Date of Conference*, 2004.

Baum, E.B. *Iterated descent: A better algorithm for local search in combinatorial optimization problems. Technical report Caltech*, Pasadena, CA. Manuscript, 1986.

Campello, R. J. G. B., Hruschka, E. R., and Alves, V. S. (2009). On the efficiency of evolutionary fuzzy clustering. Journal of Heuristics 15, page 43-75.

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

Dias, C.R.; & Ochi, L.S.. *Efficient Evolutionary Algorithms for the Clustering Problems in Directed Graphs*. Proc. of the IEEE Congress on Evolutionary Computation (IEEE-CEC), 983-988. Canberra, Austrália, 2003.

Ester, M., H.-P. Kriegel, J. Sander, & X. Xu (1996). *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.

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

Garai, G. & Chaudhuri, B. (2004), *A novel genetic algorithm for automatic clustering, Pattern Recognition Letters*, Ed. 25, pg. 173–187.

Goldschimidt R.; Passos, E. Data Mining: um guia prático. Editora Campus, Rio de Janeiro: Elsevier, 2005.

Hair, J.F, Black, W.C, Babin, B.J., Anderson, R.E. e Tatham, R.L. Análise Multivariada de Dados, Bookman, Sexta Edição, 2009.

Han, J., e Kamber, M., *Cluster Analysis*. In: Morgan Kaufmann. Publishers (eds.), *Data Mining: Concepts and Techniques*, 2 ed., chapter 8, New York, USA, Academic Press, 2006.

Hastie, t., Tibshirani, R., and Friedman, J. (2001). The elements of statistical learning. Data Mining, Inference, and prediction.

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

Hruschka, E. R. & Ebecken, N. F. F. (2003). *A genetic algorithm for cluster analysis*. Intelligent Data Analysis 7 (1), 15–25.

Hruschka, E. R., R. J. G. B. Campello, & L. N. de Castro (2004a). *Evolutionary algorithms for clustering gene-expression data*. In Proc. IEEE Int. *Conf. on Data Mining*, Brighton/England, pp. 403–406.

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

Johnson, R.; Wichern, D. Applied multivariate statistical analysis. 3.ed. New Jersey: Pretince-Hill, 2001. 642p.

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

Larose, D. T. Discovering *Knowledge in Data, An Introduction to Data Mining*. John Wiley & Sons, 2005.

Larrañaga, Pedro; & Lozano, Jose A. *Estimation of distribution algorithms: A new tool for evolutionary computation*. Kluwer Academic Publishers, Boston, 2002.

Lourenco, H. R., Martin, O. C., and Stuzle, T. (2010). Iterated local search: Framework and applications. In Glover, F. and Kochenberger, G., editors, Handbook of Metaheuristics, pages 363-397. Kluwer Academic Publishers.

Ma, P. C. H., Chan, K. C. C., Yao, X., and Chiu, D. K. Y. (2006). An evolutionary clustering algorithm for gene expression microarray data analysis. Evolutionary Computations 10, page 296-314.

Maulik, U. & Bandyopadhyay, S. (2000), Genetic Algorithm-based Clustering Technique, Pattern Recognition p.33,1455-1465.

Macqueen, J. B. (1967). Some Methods for Classification and Analysis of MultiVariate Observations. Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability. University of California Press. P. 281-297, V. 1.

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

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. 2007.

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

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

Pakhira, M., S. B. and Maulik, U. (2005). A study of some fuzzy cluster validity indices, genetic clustering and application to pixel classification. Fuzzy Sets Systems 155, page 191-214.

Pal, N. and Bezdek, J. (1995). On cluster validity for the fuzzy c-means model. IEEE Transactions of Fuzzy Systems 3, page 370-379.

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

Reeves, C. R. Genetic algorithms. In Glover, F. and Kochenberger, G., editors, Handbook of Metaheuristics, Kluwer Academic Publishers. 2010.

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

Ruspini, E. H. Numerical methods for fuzzy clustering. Information Science, 1970.

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* v.10 n4, 2012.

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

Semaan, G. S. ; Vasconcelos, R. B. ; Brito, J. A. M. ; Ochi, L. S. Proposta de um Método Baseado em Densidade e Grade para o Problema de Agrupamento Automático. XVII Simpósio de Pesquisa Operacional e Logística da Marinha. *Ed. Edgard Blücher*, 2014.

Semaan, G. S., Torres, C., Brito, J. A. M., Ochi, L. S. . Um Método Baseado em Combinação de Soluções com Coassociação para o Problema de Agrupamento Automático. Revista Brasileira de Estatística, v. 74, p. 43-68, 2013.

Soares, S. S. R. F., Ochi, L. S. *Um Algoritmo Evolutivo com Reconexão de Caminhos para o Problema de Clusterização Automática*. in *XII Latin Ibero American Congress on Operations Research*, Proc. of the XII CLAIO, 2004.

Soares, S. R. F. (2004b). Metaheurísticas para o problema de clusterização automática. Dissertação de mestrado, Universidade Federal Fluminense.

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

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