
**GUSTAVO HENRIQUE DE ARAÚJO E SOUSA**

# ESTUDO DE ALGORITMOS HÍBRIDOS PARA CLUSTERIZAÇÃO DE DADOS USANDO PSO

**LAVRAS - MG**

**2010**

**GUSTAVO HENRIQUE DE ARAÚJO E SOUSA**

# ESTUDO DE ALGORITMOS HÍBRIDOS PARA CLUSTERIZAÇÃO DE DADOS USANDO PSO

Monografia apresentada ao Colegiado do Curso de Ciência da Computação, para a obtenção do título de Bacharel em Ciência da Computação.

Orientador

Prof.Dr. Ahmed Ali Abdalla Esmin

**LAVRAS - MG**

**2010**

**GUSTAVO HENRIQUE DE ARAÚJO E SOUSA**

# ESTUDO DE ALGORITMOS HÍBRIDOS PARA CLUSTERIZAÇÃO DE DADOS USANDO PSO

Monografia apresentada ao Colegiado do Curso de Ciência da Computação, para a obtenção do título de Bacharel em Ciência da Computação.

Aprovada em *19 de Novembro de 2010*

Profa.Dra. Marluce Rodrigues Pereira

Prof.Dr. Denilson Alves Pereira

Prof.Dr. Ahmed Ali Abdalla Esmin

Orientador

**LAVRAS - MG**

**2010**

*Dedico esta monografia unicamente à minha mãe Maria Aparecida, pois sem ela, nada disso teria acontecido.*

## AGRADECIMENTOS

Agradeço especialmente aos meus pais, Maria Aparecida e José Renato, por todo o amor, carinho, orgulho, incentivo e confiança depositada em mim.

Ao meus irmãos, Ana Carolina e Luís Felipe, por todo amor, carinho, amizade, apoio e companheirismo.

Aos meus avós, Zinha e Antônio, que sempre me incentivaram a continuar estudando, pelo amor e carinho dos melhores avós do mundo.

Aos meus tios, Vera, Elisabete, Isabel e Marcos, por todo seu amor, carinho, apoio, compreensão, ajuda e companheirismo.

Aos professores do curso que me proporcionaram conhecimento no decorrer de todos esses anos.

Agradeço especialmente o professor Ahmed, que me acolheu e me orientou neste trabalho.

Agradeço a todos da República Taverna e agregados, Mário, Luciano, Euler, Lucas, Rafael, Pedro, Ana Carolina, Ana Elisa, Ana Carolina, Marianna, Tainara, Élcio, Wilker, João Victor, pelo companheirismo, amizade, por me abrigar enquanto eu não tinha casa, por me ajudar em tempos difíceis, pelas bebedeiras, pelas risadas e pelas maluquices.

Ao pessoal da Mitah Technologies, por tudo que me ensinaram.

Obrigado a todos vocês!

**RESUMO**

A quantidade de informação disponível e coletada atualmente é muito maior do que a capacidade humana de processá-la. Esse problema motiva o desenvolvimento de métodos computacionais capazes de extrair conhecimento útil desses dados de modo eficiente. O problema de *Clustering*, ou Clusterização, consiste em encontrar grupos em um conjunto de dados e é uma das principais tarefas de descoberta de conhecimento a partir de bancos de dados. O *Particle Swarm Optimization* (PSO) é uma técnica baseada no comportamento social bastante nova que vem sendo aplicada com sucesso a diversos tipos de problemas. Recentemente, foram propostas algumas abordagens que modificam o PSO para o problema de Clusterização. Tais métodos são recentes e ainda podem fazer parte de objetos de pesquisa. Este trabalho visa o estudo da combinação de algoritmos com o *Particle Swarm Optimization* para resolver o problema de Clusterização de dados.

Palavras-chave: Mineração de dados; Particle Swarm Optimization;Clusterização.

**ABSTRACT**

The amount of information available and collected nowadays is much greater than the human capacity of processing it. This problem motivates the development of computational methods capable of extracting useful knowledge such data efficiently. The Clustering problem consists on finding groups on a certain data set and is one of the main tasks on knowledge discovery from data bases. The Particle Swarm Optimization is a rather new method based on social behavior that has been successfully applied to many kinds of problems. Recently, some approaches were proposed modifying the PSO to the Clustering problem. These methods are recent and still can be part of research objects. This work aims to studying of hybrid algorithms using Particle Swarm Optimization to solve the problem of clustering data.

Keywords: Data Mining; Particle Swarm Optimization; Clustering.

## LISTA DE FIGURAS

## LISTA DE TABELAS

**SUMÁRIO**

# 1 Introdução

## 1.1 Contextualização e Motivação

A quantidade de informação, disponível e coletada hoje em dia é maior que a capacidade humana de análise e extração de conhecimento a partir dela. A quantidade excessiva de dados é um problema em quase todas as áreas, como instituições financeiras, astronomia, biologia (bioinformática) e *Web*. Este aumento dramático de informação disponível necessitando de análise motivou o desenvolvimento de técnicas e ferramentas que possuam a capacidade de extrair conhecimento de forma automática e eficiente (COHEN & CASTRO, 2006).

Ester *et al.* (1998) destacam que muitas organizações têm reconhecido o valor estratégico do conhecimento escondido em suas grandes bases de dados.

A cada dia, um volume maior de dados é coletado e armazenado em bases de dados. Com isso, surge a necessidade de métodos eficientes e efetivos para descobrir e usar a informação contida nos dados (ANKERST *et al.*, 1999).

*Data Mining* é uma etapa no processo de descoberta de conhecimento consistindo de aplicações de análise de dados e algoritmos de descoberta que, sobre limitações de eficiência computacional aceitáveis, produz uma enumeração particular dos padrões sobre os dados (ESTER *et al.*, 1998).

De acordo com Cohen e Castro (2006), *Clustering*, também conhecido como Agrupamento ou Clusterização, é uma das técnicas mais utilizadas na descoberta de conhecimento a partir de bancos de dados. Este método consiste em encontrar grupos nos quais os seus elementos são similares entre si e diferentes dos elementos de outros grupos. Como dito por Merwe e Engelbrecht (2003), Clusterização de dados é o processo de reunir vetores multidimensionais em grupos, tendo em vista obter a maior similaridade dentro de cada grupo.

O *Particle Swarm Optimization* (PSO) é uma técnica de otimização estocástica proposta inicialmente por Kennedy e Eberhart (1995) inspirado pelo comportamento social de cardumes de peixes e bandos de aves. Kennedy e Eberhart (1995) descobriram o algoritmo através da simulação de um modelo social simplificado, onde a intenção original foi simular a graciosa, porém imprevisível, coreografia de um bando de pássaros.

De acordo com Reynolds *et al.* (2005), no PSO, as soluções em potencial para o problema, chamadas partículas, se movem pelo espaço de busca, simulando o vôo dos pássaros, tendo como objetivo obter a melhor posição, ou seja, a melhor solução para o problema.

O *Particle Swarm Optimization* é um algoritmo relativamente novo e muitas vezes é aplicado com sucesso a diversos problemas de otimização. Recentemente surgiram na literatura algumas abordagens do PSO para resolver o problema de Clusterização de dados com resultados satisfatórios. Já que a aplicação do PSO em *Clustering* é recente, suas abordagens ainda podem ser trabalhadas a fim de encontrar melhores resultados.

## 1.2 Objetivos

Este trabalho tem como objetivo o estudo de algoritmos híbridos usando o *Particle Swarm Optimization* para resolver o problema de Clusterização de dados. Em outras palavras, o presente trabalho objetiva combinar o PSO com outros algoritmos a fim de encontrar melhores resultados.

Para que seja possível uma análise e comparação entre as combinações de algoritmos, foram feitos vários testes e experimentos utilizando diversos *benchmarks* famosos na área como *Iris*, *Glass* e *Wine*. Espera-se que este trabalho possa contribuir para que o PSO se torne uma ferramenta de otimização e Clusterização

de Dados consolidada além de tornar o uso de algoritmos híbridos mais comum, visando o aperfeiçoamento dos algoritmos convencionais.

Além disso, se os resultados deste trabalho forem satisfatórios, é esperado que este trabalho seja descrito em um artigo cientifico, para que seja publicado em congressos ou periódicos.

## 1.3 Tipo de Pesquisa

O presente trabalho pode ser considerado uma pesquisa Aplicada, pois visa a geração conhecimentos para uma aplicação prática, dirigindo-se à solução do problema de Clusterização. Para tal são analisadas as técnicas de Clusterização para que seja possível identificar os pontos positivos e negativos de cada técnica. Feito isso, unir dois métodos a fim de propôr um novo método que possui as qualidades de ambos que o compõem e, em seguida, a divulgação dos conhecimentos e resultados obtidos. É, também, uma pesquisa Exploratória, pois visa o aprimoramento de idéias já existentes e a descoberta de novas informações/conhecimentos.

## 1.4 Procedimentos Metodológicos

As técnicas de Clusterização foram estudadas e implementadas no período entre fevereiro e setembro de 2010, utilizando a linguagem de programação Java e o ambiente de desenvolvimento Eclipse. Os métodos de *Clustering* foram profundamente estudados, analisados e testados a fim de obter uma comparação entre cada método e a possibilidade de encontrar deficiências e melhorias que pudessem ser sugeridas a partir da fusão de algoritmos, objetivando obter melhor desempenho.

## 1.5 Organização do Texto

Este trabalho é organizado para que qualquer pessoa, possa entender todos os passos, métodos e resultados obtidos. No Capitulo 2, são explanados todos os conceitos julgados necessários para a melhor compreensão deste trabalho. Já no Capítulo 3, é descrito a metodologia utilizada para solucionar o problema, ou seja, serão abordados a Clusterização de dados utilizando o PSO e o modelo híbrido *Hybrid Particle Swarm Optimization with Mutation*. O Capítulo 4 descreve os software implementado, onde foram implementados e testados uma série de algoritmos e os resultados obtidos a partir de uma análise critica dos dados. Na conclusão (Capítulo 5) são expostos os resultados obtidos a proposta de hibridização do PSO. Também são propostos trabalhos futuros que visam melhorar o método proposto e aplica-lo a um problema real.

# 2 Referencial Teórico

## 2.1 *Data Mining*

### 2.1.1 Introdução

Esta seção tem como objetivo introduzir o assunto *Data Mining*, ou Mineração de Dados. Assim, será abordado uma definição ampla sobre o assunto, a explicação acerca do porquê do seu surgimento, quais as suas aplicações e qual é o processo de aquisição de conhecimento a partir de grandes quantidades de dados.

### 2.1.2 Mineração de Dados

A mineração de dados tem estado em ascensão nos últimos anos graças à larga disponibilidade de enormes quantidades de dados e à necessidade eminente de transformar esses dados em informação utilizável (HAN & KAMBER, 2001).

De acordo com Han e Kamber (2001), *Data Mining* é tida como o resultado da evolução natural das Tecnologias de Informação (TI). Os passos da evolução no ramo de banco de dados podem ser descritos na seguinte sequência (Figura 1): coleta de dados e criação de banco de dados (BD), gerenciamento de dados (incluindo armazenamento de dados, recuperação e controle de transação), e análise de dados e compreensão (que envolve o armazenamento de dados e mineração de dados). Como se pode observar, na Figura 1, cada fase da evolução das tecnologias de persistência depende diretamente da anterior. Por exemplo, o mecanismo de coleta e criação de banco de dados é um pré-requisito para o desenvolvimento de aplicações que permitam o armazenamento de dados, recuperação e controle de transação.

Devido a esse grande aperfeiçoamento das tecnologias de informação e também ao incrível progresso das tecnologias de *hardware* voltadas para o arma-

zenamento de dados ao longo de mais de três décadas, tornou-se muito fácil e rápido a aquisição de grandes quantidades de dados. No entanto, é impraticável para um ser humano interpretar de forma útil uma quantidade imensa de dados manualmente, ou seja, sem a utilização de ferramentas de obtenção de conhecimento. Como resultado desse imenso fluxo de dados, os BD's com larga quantidade de dados podem se tornar os chamados *data tombs*, ou túmulos de dados, que são bancos de dados com grande quantidade de informações que são raramente utilizadas. Assim, surge a necessidade de se obter ferramentas de análise de dados (HAN & KAMBER, 2001).

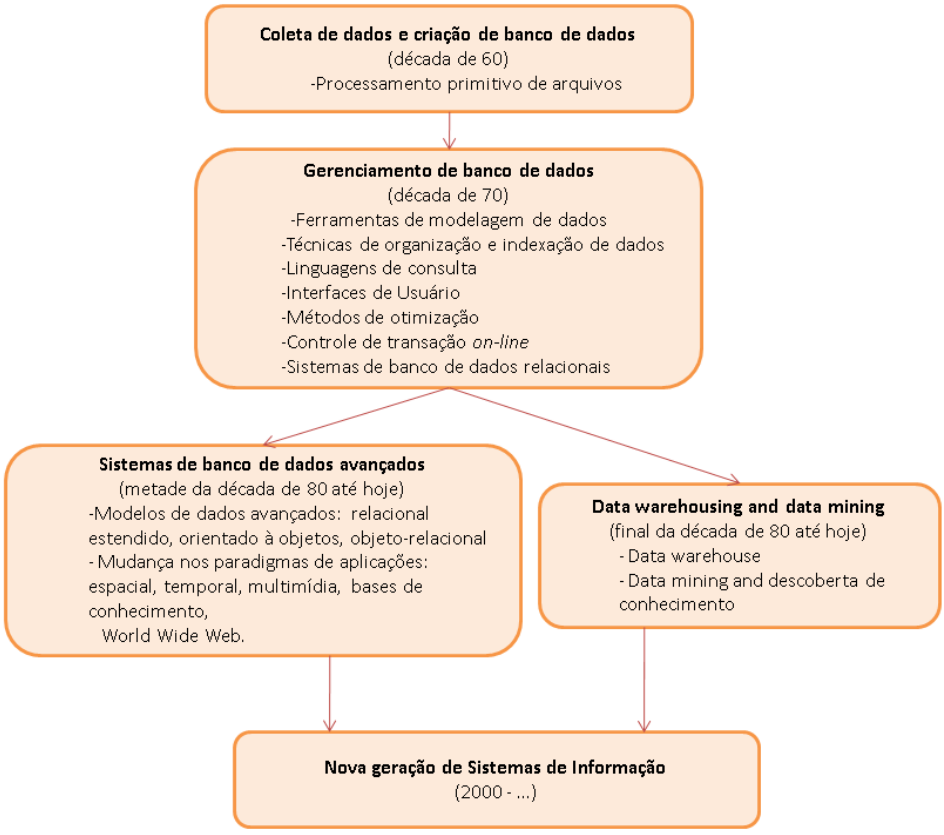


Figura 1: Evolução da tecnologia de banco de dados

No mercado global atual, a competição entre as empresas para atingir seus clientes é tão grande, que uma boa organização da produção, redução de custos e bom atendimento não garantem o sucesso da empresa. É preciso, cada vez mais, obter ferramentas que aumentem a proximidade com o cliente. A mineração de dados permite, a partir de grandes quantidades de dados adquiridos dos clientes, ser capaz de interpretar e compreender seus objetivos, expectativas e desejos. Assim, é possível prover produtos que suprem cada vez melhor as necessidades do cliente, trazendo assim melhor satisfação para o mesmo e maior competitividade para a empresa.

*Data Mining* é o conjunto de métodos que provê, automaticamente, padrões em uma base de dados livre da tendenciosidade e limitação de uma análise baseada meramente na observação humana (BRAGA, 2005). "Presidentes de grandes corporações como IBM, Microsoft e Harley-Davidson não foram capazes de prever que o mercado ia preferir PC's, Internet e motos populares." (BRAGA, 2005)

De acordo com Han e Kamber (2001), mineração de dados é definida como extração, ou "mineração", de conhecimento de grandes quantidades de dados. Em outras palavras, minerar dados é vasculhar uma grande quantidade de dados à procura de padrões, regras de associação e grupos com similaridade, com o objetivo de obter conhecimento a cerca dos dados em questão. O termo *Data Mining* faz referência aos mineradores de ouro, que dentre enormes rochas, conseguiam extrair tesouros. Além disso, existem outros termos que possuem sentido similar, tais como mineração de conhecimento, extração de conhecimento e análise de dados/padrões.

Alguns autores afirmam que mineração de dados possui o mesmo significado que KDD, ou *Knowledge Discovery in Databases* (Figura 2), porém, outros

dizem que *Data Mining* é simplesmente uma das fases que envolvem o processo de KDD, que, de acordo com Han e Kamber (2001), consiste numa sequência iterativa dos seguintes passos:

- **Limpeza de dados** - Consiste na retirada de ruídos e dados irrelevantes. Essa fase é muito importante, pois esses dados retirados da base poderiam, além de tornar a mineração mais lenta, atrapalhar na aquisição de conhecimento (SILVA, 2000).

- **Integração de dados** - Algumas vezes, os dados, os quais se quer analisar, estão separados em diversas bases de dados. Assim, essa fase visa integrar todos os bancos de dados envolvidos em apenas um, a fim de facilitar o uso das ferramentas de *Data mining*.

- **Seleção dos dados** - Onde os dados relevantes à tarefa de análise são obtidos através do banco de dados.

- **Mineração dos dados** - Uma fase essencial, onde se seleciona os métodos a serem utilizados para localizar os padrões nos dados. Em seguida, ocorre a efetiva busca pelos padrões de interesse numa forma particular de representação ou conjunto de representações.

- **Avaliação dos padrões encontrados** - Nessa fase, é feita uma avaliação dos resultados da mineração dos dados, a fim de constatar que a mesma reconheceu padrões que são interessantes e/ou úteis.

- **Apresentação do conhecimento** - Visualização dos padrões encontrados na mineração para o usuário final.

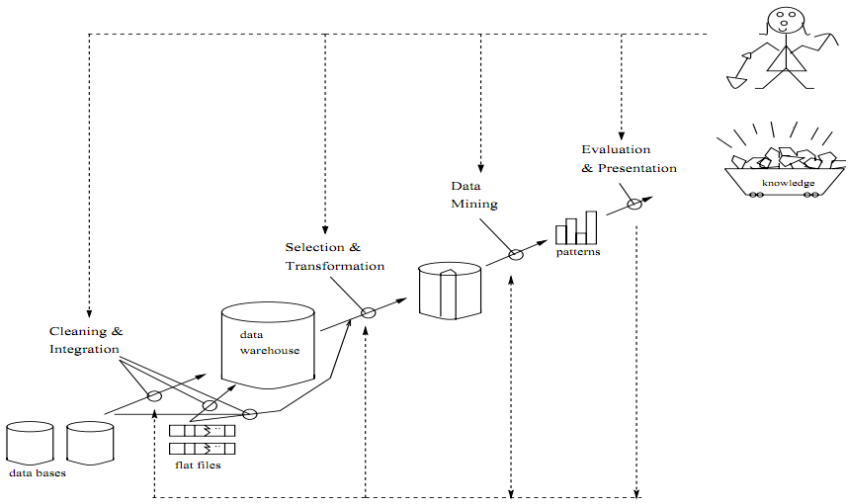


Figura 2: Fluxograma do KDD (HAN & KAMBER, 2001)

## 2.2 Clusterização

### 2.2.1 Introdução

Esta seção tem como assunto principal o problema de *Clustering*, ou Clusterização. Na subseção 2.2.2, o problema será discutido, juntamente com a sua importância e dificuldade dos algoritmos de Clusterização, como por exemplo encontrar o conjunto correto de *clusters*. Já na subseção 2.2.3, é apresentada a definição formal do problema. Será visto na subseção 2.2.4 que o problema de *Clustering* tem aplicações em diversas áreas do conhecimento. Na subseção 2.2.5 é demonstrado como medir a similaridade entre dois dados e como normalizar os dados para que a unidade de medida não influencie no agrupamento. Na subseção 2.2.6, será visto que os métodos de Clusterização podem ser divididos em várias classificações. Por último, dois algoritmos de Clusterização vão ser explanados, *k-means* e *k-harmonic means*, que são algoritmos já conhecidos na literatura e serão utilizados neste trabalho.

### 2.2.2 Clusterização

O verbo *to cluster*, em inglês, significa agrupar. No contexto de *Data Mining*, clusterizar corresponde ao processo de encontrar grupos em um conjunto de dados de modo que os elementos de um grupo possuam maior similaridade entre si do que com elementos de outros grupos (CARLANTONIO, 2001).

De acordo com Carlantonio (2001), se qualquer ser humano observar o conjunto de objetos da Figura 3 e fosse pedido-lhe que agrupasse os objetos, o mesmo certamente formaria três grupos, um com retângulos, um com elipses e outro com triângulos. Em outras palavras, para um ser humano, muitas vezes é trivial agrupar objetos, no entanto, o computador não tem esta mesma facilidade.

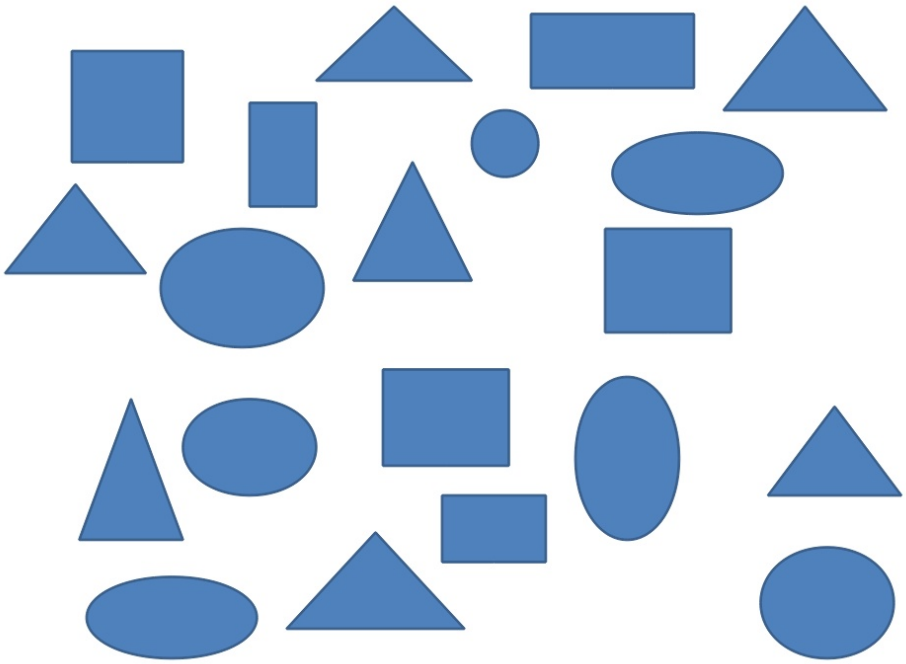

Figura 3: Conjunto de Objetos

Cole (1998) define *Clustering* como o conjunto de técnicas usadas a fim de obter uma estrutura "natural"dentro de um conjunto de dados. Os dados são arranjados com base na similaridade entre eles, indiferente à estrutura de dados na qual os dados estão representados.

Entre as aplicações de Clusterização está a redução de dados (técnica que reduz uma grande quantidade de dados em um conjunto de elementos que caracterizam os seus grupos), desenvolvimento de esquemas de classificação (também conhecido como taxonomias), e sugestão ou suporte de hipóteses de estruturas de dados (COLE, 1998).

A Clusterização é uma tarefa preliminar à execução da Classificação, pois não é possível classificar dados que não possuam classe. Assim, *Clustering* também pode ser entendido como o ato de classificar por inteiro o conjunto de dados. A partir do agrupamento, é possível analisar os elementos que compõem cada um dos *clusters*, identificando assim as características comuns aos seus elementos. Feito isso, é possível criar um nome que represente cada grupo, dando origem a uma classe (CARLANTONIO, 2001).

Han e Kamber (2001) afirmam que a Clusterização de Dados é uma nova disciplina científica e está em um forte desenvolvimento nos últimos anos. Existe uma grande quantidade de pesquisas sendo feitas, em sua maioria na área de *Data Mining*, Estatística, Aprendizagem de Máquina, Banco de Dados Espaciais, Biologia (Bioinformática), Marketing, etc. Essas pesquisas frequentemente se tornam artigos em periódicos.

De acordo com Ankerst *et al.* (1999) o propósito de identificar *clusters* é obter a partição de um banco de dados de registros tal que os registros possuam similaridade entre si. Isso, então, permite que as características de cada grupo possam ser descritas.

### 2.2.3 Definição Formal

De acordo com Cole (1998) devemos definir o problema de Clusterização da seguinte forma:

Considerando a clusterização de um conjunto de dados de n objetos X = $\{X_1, X_2, ..., X_n\}$ , onde cada $X_i \in \Re^p$ é um vetor de $p$ medidas reais que dimensionam as características do objeto, estes devem ser clusterizados em grupos disjuntos C = $\{C_1, C_2, ... ,C_k\}$, onde $k$ é o número de *clusters*, de forma que tenhamos as seguintes condições respeitadas:

- $C_1 \cup C_2 \cdots \cup C_k = X$, ou seja, todos os elementos contidos em $X$ devem estar em algum *cluster* $C_i$.
- $C_i \neq \varnothing$, isto é, não deve haver nenhum *cluster* vazio.
- $C_i \cap C_j = \varnothing$, para $i \neq j$, em outras palavras, os elementos de $X$ não podem estar contidos em dois *clusters* diferentes.

Os objetos que compõem cada grupo devem possuir maior similaridade entre si do que com objetos de outros grupos. O valor de $k$ pode ser conhecido, se este for o caso, o problema é chamado de $k$-Clusterização.

### 2.2.4 Aplicações

O agrupamento de dados pode gerar conhecimento em praticamente qualquer área do conhecimento, sejam eles relativos às compras efetuadas em um supermercado, às especificações físicas e químicas de petróleo, aos sintomas de doenças, às características dos seres vivos ou aos documentos existentes na *Web* (CARLANTONIO, 2001).

De acordo com Cole (1998), inúmeros problemas já foram resolvidos com o uso de *Clustering*. Na Psiquiatria, Clusterização já foi usada como forma de

classificar se o paciente possui ou não depressão. Em Pesquisa de Mercado, para identificar grupos homogêneos de mercados de teste. Assim como na Arqueologia, para classificar machados de mão ingleses. No Reconhecimento de Padrões, para segmentar imagens. Na Medicina, como método de aquisição de conhecimento para sistemas de diagnósticos.

Han e Kamber (2001) dizem que na área de negócios, Clusterização pode ajudar a encontrar grupos de clientes que possuem comportamento similar baseando-se no padrão de compra. Na Biologia, pode ser usado para derivar taxonomias de plantas e animais, categorizar genes com funcionalidades semelhantes e analisar melhor estruturas inerentes às populações. Na Geografia, para encontrar *clusters* de casas em uma cidade de acordo com o seu tipo, valor, localização geográfica. Pode ajudar também na classificação de documentos da web.

Ankerst *et al.* (1999) descrevem como aplicações: a criação de mapas temáticos em sistemas de informação geográfica por clusterizar espaços característicos; a detecção de *clusters* de objetos em sistemas de informação geográfica; e a clusterização de base de dados de *Web-Log* para descobrir grupos de padrões de acessos similares que podem corresponder a diferentes padrões de usuários.

### 2.2.5 Medidas de Similaridade

*Clusters* são formados a partir de elementos que possuem maior similaridade entre eles do que com elementos de outros *clusters*. Para que seja possível criar grupos, é necessário primeiramente definir como é medida a distância entre dois objetos. Uma pequena distância significa grande similaridade, portanto a medida de distância pode, também, ser usada para quantificar a dissimilaridade (COLE, 1999).

Considerando que $X_i$ e $X_j$ são dois objetos do conjunto de dados, $p$ é a quantidade de atributo que cada elemento possui. $X_{it}$ e $X_{jt}$ são os valores do $t^{-ésimo}$ atributo de $X_i$ e $X_j$, HAN e KAMBER (2001) destacam que a medida de similaridade mais utilizada é a distância Euclidiana:

$$d(X_i, X_j) = \sqrt{\sum_{t=1}^{p} (X_{it} - X_{jt})^2} \tag{1}$$

que é a distância em linha direta entre os dois pontos que representam os objetos.

Outra medida bastante usada é a distância *Manhattan* (ou *City Block*) definida por:

$$d(X_i, X_j) = \sum_{t=1}^{p} |X_{it} - X_{jt}|. \tag{2}$$

Ambos os métodos são generalizações da distância *Minkowski*:

$$d(X_i, X_j) = (\sum_{t=1}^{p} (X_{it} - X_{jt})^n)^{\frac{1}{n}}. \tag{3}$$

Em que $n$ é um valor arbitrário. É fácil perceber que a distância Euclidiana é a distância *Minkowski* com o valor de $n$ igual a 2. Também, a distancia *Manhattan* é tida como *Minkowski* com o $n$ valendo 1.

Segundo Han e Kamber (2001), existem alguns atributos que podem, além de aumentar o custo computacional do algoritmo, prejudicar a precisão do resul-

tado final do algoritmo. Por isso, nem todos os atributos devem ser, necessariamente, incluídos na Clusterização. Além disso, pesos podem ser atribuídos aos atributos de acordo com a sua relevância. A distância Euclidiana ponderada pode ser calculada da seguinte forma:

$$d(X_i, X_j) = \sqrt{\sum_{t=1}^{p} w_t (X_{it} - X_{jt})^2}, \tag{4}$$

onde $w_t$ é um peso atribuído à variável $t$. O mesmo pode ser aplicado com as distâncias *Manhattan* e *Minkowski*.

Pereira (2007) destaca que a unidade de medida pode afetar conclusivamente o resultado na Clusterização. Por exemplo, a alteração de metros para quilômetros ou de gramas para quilos pode afetar o desempenho do *Clustering*. Para evitar esse problema, os dados devem ser alterados de modo que as variáveis tenham pesos iguais.

Uma das soluções mais comuns é transformar as medidas em variáveis sem escala, o que pode ser feito da seguinte maneira:

1. Calcular o desvio absoluto médio, $s_f$:

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \cdots + |x_{nf} - m_f|), \tag{5}$$

onde $x_{1f}, x_{2f}, \ldots, x_{nf}$ são medidas do atributo $f$ e $m_f$ é o valor médio de $f$:

$$m_f = \frac{1}{n}(x_{1f} + x_{2f} + \cdots + x_{nf}), \tag{6}$$

2. Calcular a medida padronizada, conhecida como *z-core*, da seguinte maneira:

$$z_{if} = \frac{x_i f - m_f}{s_f}. \tag{7}$$

A padronização pode ou não ser feita, dependendo da aplicação. Por exemplo, numa aplicação de classificação de jogadores de basquete, pode não ser interessante padronizar a altura dos jogadores, pois é um atributo que possui maior importância.

### 2.2.6 Métodos de Clusterização

De acordo com Carlantonio (2001), o método ideal de *Clustering* deveria seguir os seguintes requisitos:

- Descobrir *clusters* de forma arbitrária. Os métodos de *Clustering* baseados nas medidas de distância Euclidiana ou Manhattan tendem a encontrar clusters esféricos de tamanho e densidade similares.
- Identificar *clusters* de tamanho variado. Algumas técnicas tendem a encontrar grupos de tamanho homogêneo.
- Aceitar os diversos tipos de variáveis possíveis (numérico, booleano, enumerado).

- Não possuir sensibilidade à ordem de apresentação dos elementos. Independente da ordem a qual os objetos forem apresentados, deseja-se que não haja mudança na Clusterização.

- Aceitar objetos com qualquer número de dimensões. Em outras palavras, não deve importar a quantidade de atributos que cada objeto possui.

- Ser escalável para trabalhar com qualquer quantidade de objetos.

- Prover dados interpretáveis e utilizáveis.

- Ser robusto na presença de ruídos. A maioria das bases de dados reais possui quantidade considerável de ruídos ou dados perdidos, então a qualidade dos *clusters* não deve ser comprometida por estes dados.

- Exigir o mínimo de conhecimento para definir os parâmetros de entrada. Para executar métodos de Clusterização, os valores dos argumentos de entrada apropriados são, frequentemente, desconhecidos e/ou difíceis de determinar, especialmente, para conjuntos de objetos de alta dimensionalidade e de grande número de objetos. Em alguns métodos, os resultados do *Clustering* são bastante sensíveis aos parâmetros de entrada.

- Trabalhar com restrições. Nos problemas do mundo real, frequentemente são encontradas situações onde é necessário haver restrições. Os métodos devem agrupar os dados de forma a satisfazer as restrições.

- Encontrar o número "natural" de *clusters*. Encontrar o número adequado de grupos em um conjunto de dados é muitas vezes uma tarefa difícil. Alguns métodos precisam de um valor de referência.

Agrawal *et al.* (1998) destacam que nenhum dos métodos de Clusterização atuais atende a todos os requisitos da técnica ideal. No entanto, um trabalho considerável tem sido feito para atender a cada ponto separadamente.

Os principais métodos de Clusterização podem ser classificados nas seguintes categorias (HAN & KAMBER, 2001):

1. **Métodos de Particionamento:**

   Dado uma base de dados de $n$ objetos ou tuplas, os métodos de particionamento constroem $k$ partições dos dados, sendo que $k \leq n$. O método cria um particionamento inicial e, iterativamente, vai realocando os dados de uma partição para outra de modo que eles se ajustem melhor. Em outras palavras, o bom particionamento é tido quando os dados do mesmo *cluster* estão próximos e os dados de *clusters* diferentes estão longe (HAN & KAMBER, 2001).

   Algoritmos são efetivos se o número $k$ de *clusters* puder ser razoavelmente estimado, se os *clusters* são de forma convexa e se possuem tamanho e densidade similares (ANKERST *et al.*, 1999).

   Após a divisão inicial, há duas possibilidades na escolha do "elemento" que vai representar o centro do *cluster*, e que será a referência para o cálculo da medida de similaridade. Se utilizarmos a média dos objetos que pertencem ao *cluster* em questão, também chamada de centro de gravidade do *cluster*, esta é a abordagem conhecida como *k-means*. Alternativamente, se for escolhido como representante o objeto que se encontra mais próximo ao centro de gravidade do *cluster*. Esta abordagem é conhecida como *k-medoids*, e o elemento mais próximo ao centro é chamado de *medoid* (CARLANTONIO, 2001).

Os métodos de particionamento tem dificuldades de trabalhar com *clusters* de formas complexas e tamanhos diferentes (HAN & KAMBER, 2001).

2. **Métodos Hierárquicos:**

   De acordo com Han e Kamber (2001), os métodos hierárquicos criam uma decomposição hierárquica de um dado conjunto de dados. A decomposição hierárquica pode ser representada por um dendrograma, uma árvore que iterativamente divide a base de dados em subconjuntos menores até que cada subconjunto consista de somente um objeto. Existem duas classificações para os métodos de Clusterização Hierárquicos, eles podem ser Aglomerativos ou Divisivos.

   Os métodos Aglomerativos, também conhecidos como *botton-up*, toma, inicialmente, cada elemento do conjunto de dados como um *cluster*. A partir daí o algoritmo vai iterando e fundindo os *clusters* que são mais similares até que todos os grupos tenham sido fundidos em um *cluster* final, o nível mais alto da hierarquia. O Algoritmo também pode parar se um número máximo de *clusters* tenha sido alcançado ou se a distância entre dois *clusters* mais próximos esteja acima dos limites estipulados (HAN & KAMBER, 2001).

   Já os métodos Divisivos, também chamados de *top-down* iniciam com o nível mais alto da hierarquia, isto é, um único *cluster* que contém todos os objetos e se divide em *clusters* menores até que cada grupo contenha apenas um objeto ou algum critério de parada seja alcançado (CARLANTONIO, 2001).

3. **Métodos Baseados em Densidade:**

   Segundo Carlantonio (2001), a maioria dos métodos de Clusterização se baseia na distância entre os objetos. Assim, esses métodos são bons para

encontrar *clusters* apenas na forma esférica e, portanto, têm dificuldades em encontrar *clusters* de formato arbitrário.

As abordagens baseadas em densidade aplicam um critério de *cluster* local e são muito utilizados em *Data Mining*. *Clusters* são considerados regiões do espaço de dados nas quais a distribuição de objetos é densa e são separados por regiões do espaço onde há baixa densidade, conhecidas como *noise*, ou ruído. Essas regiões podem ter forma arbitrária e os dados em uma região podem estar distribuídos arbitrariamente. Em outras palavras, os métodos baseados em densidade vêm para sanar a necessidade de encontrar *clusters* de tamanho e forma irregular, que é um problema de outras abordagens (ANKERST *et al.*, 1999).

4. **Métodos Baseados em *Grid*:**

   Um método baseado em *Grid* cria no espaço de dados uma estrutura de grade (*grid*) multidimensional com um número finito de células. Este método executa todas as operações de *Clustering* na estrutura de grade. A principal vantagem desta abordagem é a velocidade de processamento que é normalmente independente do número de elementos do conjunto de dados, dependendo apenas do número de células da estrutura de *grid*.

   A ideia é obter informações estatísticas associadas com as células, e não aos elementos, de maneira que classes inteiras de consultas e problemas de Clusterização possam ser resolvidos sem recorrer aos objetos individuais (PEREIRA, 2007).

5. **Métodos Baseados em Modelos:**

   De acordo com Han e Kamber (2001), métodos baseados em modelos hipotetiza um modelo para cada um dos *clusters*, e encontra o melhor encaixe

dos dados ao modelo. Um algoritmo baseado em modelos pode localizar *clusters* construindo uma função de densidade que reflete a distribuição espacial dos pontos. Isso nos leva a um modo de determinar automaticamente o número de *clusters*, mesmo se levarmos os ruídos em conta. Portanto, os métodos de Clusterização baseados em modelos podem ser considerados como métodos robustos, uma vez que não são influenciados por *noise*.

### 2.2.7 *K-Means*

*K-means* (KM) é uma técnica de Clusterização de Particionamento proposta inicialmente por MacQueen (1967) e tem o propósito de particionar uma população *n*-dimensional em *k* grupos em uma dada base de dados. Este método será usado no trabalho tanto como medida de comparação com outros algoritmos, que serão visto na Seção 4.2, quanto para a hibridização de algoritmos, que será abordado na Subseção 3.2.1.

De acordo com Carlantonio (2001), o algoritmo KM toma um parâmetro de entrada, k, e particiona um conjunto de *n* objetos em *k clusters* de modo que a similaridade *intracluster* resultante é alta, mas a similaridade entre *clusters* diferentes é baixa. A similaridade de *clusters* é medida em respeito ao valor médio dos objetos em um *cluster*, que pode ser visto como o centro de gravidade do *cluster*.

O algoritmo *k-means* (Figura 4) trabalha da seguinte maneira. Inicialmente, o método, aleatoriamente, seleciona *k* objetos, cada um dos quais, inicialmente, representa os centróides, ou centros dos *clusters*. Para cada um dos objetos da base de dados, é feita a atribuição ao *cluster* ao qual o objeto é mais similar, baseado na distância entre o objeto e a média do *cluster*. Ele, então, computa o novo valor dos centróides, que é tido pela média dos valores dos objetos que estão

atribuídos à cada *cluster*. Este processo se repete até que o valor dos centros não se altere significativamente (CARLANTONIO, 2001).

**Algoritmo *k-means*** – o algoritmo *k-means* para particionamento baseia-se no valor médio dos objetos no cluster.
**Entrada:** O número de clusters, *k*, e a base de dados com *n* objetos.
**Saída:** Um conjunto de *k* clusters que minimizam o critério do erro quadrado.
**Método:**

1. Escolha arbitrariamente *k* objetos da base de dados como os centros inicias dos clusters;
2. Repita
3. (re)atribua cada objeto ao cluster ao qual o objeto é mais similar, de acordo com o valor médio dos objetos no cluster;
4. atualize as médias dos clusters, isto é, calcule o valor médio dos objetos para cada cluster;
5. até que não haja mudança de objetos de um cluster para outro.

Figura 4: Algoritmo *k-means*

O método *k-means*, entretanto, pode ser aplicado somente quando a média de um *cluster* é definida. Isto pode não ser o caso em algumas aplicações, tais como quando dados com atributos categóricos (nominais) estão envolvidos (CARLANTONIO, 2001).

Zhang *et al.* (1999) comentam que a performance do algoritmo KM é sensível à partição inicial, gerada pela escolha aleatória dos centros iniciais.

Segundo Carlantonio (2001), a necessidade de o usuário ter que especificar k, o número de *clusters*, com antecedência é uma desvantagem. O método *k-means* não é adequado para descobrir *clusters* com formas não convexas ou grupos de dimensões muito diferentes. Além disso, ele é sensível a ruídos, visto que pequeno

número de tais dados pode influenciar, substancialmente, o valor resultado final do algoritmo.

Apesar de ser bastante rápido e de fácil implementação, o método *k-means* possui alguns problemas aos quais existem estudos tentando resolver (HAN & KAMBER, 2001).

### 2.2.8 *K-Harmonic Means*

*K-Harmonic Means*, ou KHM, é um algoritmo bastante novo e foi proposto por Zhang *et al.* (1999) e modificado por Hammerly e Elkan (2002). O método é, assim como o *k-means* um algoritmo de Clusterização de particionamento baseado em centróides e iterativo, que refina seus *clusters* definidos pelos seus *k* centros. A diferença entre o KM e o KHM é que, a cada iteração do método, quando ocorre o recalculo dos *k* centros (passo 4 da Figura 4), o *k-means* dá pesos iguais a todos os elementos atribuídos a cada *cluster* devido à média aritmética. Alternativamente, o *k-harmonic means* dá, a cada iteração do algoritmo, peso dinâmico a cada elemento graças ao uso da média harmônica. A média harmônica atribui um peso grande para os elementos que não estão próximos de qualquer centro e um peso pequeno ao elementos que estão perto de um ou mais centros (JIANG *et al.*, 2010). Güngör e Ünler (2007) destacam que este princípio é importante por que queremos evitar zonas muito povoadas com múltiplos centros. Ao aumentar o peso dos centros que não estão perto de outros, o algoritmo pode atrair centróides para longe daquelas áreas densas, sem ter que, consequentemente, aumentar o peso dos elementos que estão em zonas mais densas.

Zhang *et al.* (1999) demonstram em seu trabalho que *K-harmonic means* é insensível à inicialização dos centróides. Além disso, em alguns casos, o KHM

melhorou significativamente a qualidade dos *clusters* encontrados, se comparado com *k-means*.

Antes de introduzirmos o algoritmo de Clusterização por *k-harmonic means*, é necessária a explicação de algumas notações usadas no procedimento (GÜNGÖR & ÜNLER, 2007).

- $x_i$: $i$-ésimo elemento do conjunto de dados de tamanho $N$, onde $i = 1,...,N$.
- $c_j$: $j$-ésimo centróide, onde $j = 1,...,k$.
- $m(c_j/x_i)$: valor do grau de adesão $x_i$ ao *cluster* $j$.
- $w(x_i)$: valor do grau de influencia do elemento $x_i$ à posição do centróide $c_j$ na próxima iteração.

De acordo com Jiang *et al.* (2010), o algoritmo do *k-harmonic means* pode ser detalhado como:

1. Inicializa-se o algoritmo KHM escolhendo os $k$ centróides aleatoriamente.
2. Calcule o valor função objetivo de acordo com:

$$KHM(X,C) = \sum_{i=1}^{N} \frac{k}{\sum_{j=1}^{k} \frac{1}{\|x_i - c_i\|^p}}, \tag{8}$$

onde $X$ é o conjunto de dados, $C$ é o conjunto de centróides, $x_i$ é o $i^{-ésimo}$ elemento do conjunto $X$, $c_i$ é o centroide ao qual $x_i$ pertence. $p$, por sua vez, é um parâmetro de entrada e é provado que KHM funciona melhor com $p > 2$;

3. Calcule a adesão de cada elemento $x_i$ a cada centróide $c_j$ de acordo com:

$$m(c_j/x_i) = \frac{\| x_i - c_j \|^{-p-2}}{\sum_{j=1}^{k} \| x_i - c_j \|^{-p-2}}, m(c_j/x_i) \in [0,1]. \quad (9)$$

4. Calcule o peso da cada elemento de acordo com:

$$w(x_i) = \frac{\sum_{j=1}^{k} \| x_i - c_j \|^{-p-2}}{(\sum_{j=1}^{k} \| x_i - c_j \|^{-p})^2}. \quad (10)$$

5. Calcule a localização do novo centro usando o peso e a adesão de cada elemento de acordo com:

$$c_j = \frac{\sum_{i=1}^{N} m(c_j/x_i).w(x_i).x_i}{\sum_{i=1}^{N} m(c_j/x_i).w(x_i)}. \quad (11)$$

6. Repita os passos 2-5 até um número de iterações predefinido ou até o valor da função objetivo $KHM(X,C)$ não se alterar significativamente.

7. Atribua o elemento $x_i$ ao *cluster* $j$ com o maior $m(c_j/x_i)$.

Jiang *et al.* (2010) ainda destacam que devido à $m(c_j/x_i)$, o algoritmo KHM é particularmente útil quando as fronteiras dos *clusters* não são bem separadas e, portando, ambíguas. Além disso, o algoritmo KHM é menos sensível à inicialização que o algoritmo KM.

## 2.3 *Particle Swarm Optimization*

### 2.3.1 Introdução

*Particle Swarm Optimization* (PSO) é um método de otimização baseado em população inicialmente proposto por Kennedy e Eberhart (1995) inspirado no comportamento social de bando de pássaros. A busca por alimento ou pelo ninho e a interação entre os pássaros ao longo do vôo são modelados como um mecanismo de otimização (SARAMAGO & PRADO, 2005).

O PSO tem obtido sucesso na aplicação de problemas em diversos domínios, particularmente na otimização de funções numéricas. Além disso, tem capacidade de resolver a maior parte dos problemas resolvidos por Algoritmos Genéticos, que é um outro método de otimização muito utilizado (KENNEDY & EBERHART, 1995).

No PSO, cada indivíduo da população busca pela solução ótima levando em consideração o melhor individuo, em uma certa vizinhança, e a melhor posição pessoal já encontrada (KENNEDY & EBERHART, 1995).

De acordo com Kennedy e Eberhart (1995), suas principais características incluem baixa complexidade na implementação, baixo custo de memória e velocidade.

A subseção 2.3.2 fala sobre como o algoritmo foi descoberto, durante a simulação da movimentação de bandos de pássaros. Adiante, a subseção 2.3.3 apresenta a definição formal do método. Finalmente, a seção 2.3.4 dá uma classificação para o PSO, que ocupa algum lugar entre o Algoritmo Genético e a Programação Evolutiva.

### 2.3.2 Origem

O algoritmo PSO nasceu da simulação de um ambiente social simplificado onde a intenção era simular o movimento de um bando de pássaros.

De acordo com Kennedy e Eberhart (1995), algumas simulações propostas anteriormente se baseiam na distância entre indivíduos, onde imagina-se que a sincronia do comportamento do grupo está relacionada aos esforços dos indivíduos em manter uma distância ótima entre si e sua vizinhança. Segundo os autores, é sensato supor, com certa abstração, que algumas das mesmas regras fundamentam o comportamento social animal, incluindo rebanhos, cardumes, bandos e até mesmo humanos. Além disso, Kennedy e Eberhart (1995) afirmam que o compartilhamento de informações em um ambiente social é uma vantagem evolutiva, pois indivíduos de uma população podem obter vantagens das descobertas e experiências passadas de outros indivíduos. Os humanos tendem a ajustar crenças e atitudes conforme outros humanos, na maioria das vezes aquele homem que possui maior influência, e usam experiências passadas na tomada de decisões e comportamentos (KENNEDY & EBERHART, 1995).

### 2.3.3 *Particle Swarm Optimization*

De acordo com Merwe e Engelbrecht (2003), dado um problema, o algoritmo mantém uma enxame de partículas de tamanho $s$, em que cada partícula representa um candidato potencial para a solução do problema. Cada partícula se encontra em uma posição em um espaço de busca multidimensional e mantém as seguintes informações:

- $f$ – *fitness* da partícula. Este valor indica quão bem a partícula está posicionada no espaço de busca. Em outras palavras, o *fitness* de cada partícula

significa quão boa ela é para resolver o problema. Quanto mais próximo de zero, melhor.

- $x_i$ – Posição atual da partícula.
- $v_i$ – Velocidade Atual da partícula.
- $y_i$ – Melhor posição pessoal da partícula (melhor posição em que a partícula já esteve).

Assim como na natureza, para que as partículas circulem no espaço de busca, é necessário que cada partícula possua uma velocidade. Assim, a cada iteração a partícula tem sua velocidade ajustada de acordo com a seguinte equação:

$$v_{i,k}(t+1) = wv_{ik}(t) + c_1 r_{1,k}(t)(y_{i,k}(t) - x_{i,k}(t)) + c_2 r_{2,k}(t)(\hat{y}_k(t) - x_{i,k}(t)). \quad (12)$$

Onde $v_{i,k}$ denota a $k^{-ésima}$ dimensão do vetor velocidade associado à $i^{-ésima}$ partícula. O valor $w$ é um peso fornecido pelo usuário, que pode-se manter fixo ao longo de todo o algoritmo ou variar entre um $w_{min}$ e um $w_{máx}$ de acordo com a seguinte equação:

$$w = w_{min} + i\frac{w_{máx} - w_{min}}{t}. \quad (13)$$

Onde $i$ é o numero da iteração e $t$ é o número máximo de iterações do algoritmo.

A velocidade é modificada separadamente em cada dimensão $k \in i \ldots n$. $r_1$ e $r_2$ são valores aleatórios, $r_1 \sim U(0,1)$ e $r_2 \sim U(0,1)$, ou seja, entre zero e um, que

contribuem para a natureza estocástica do algoritmo. $c_1$ e $c_2$, $0 < c_1, c_2 \leq 2$, são coeficientes de aceleração, $c_1$ regula o passo máximo na direção da melhor posição pessoal e $c_2$ na direção da posição global (*gbest*, ou *global best*) ou da vizinhança (*lbest*, ou *local best*) (BERGH, 2001). O termo $c_1 r_{1,k}(t)(y_{i,k}(t) - x_{i,k}(t))$ leva em conta as experiências passadas da partícula e é associado a cognição. O termo $c_2 r_{2,k}(t)(\hat{y}_k(t) - x_{i,k}(t))$ é o termo social, pois a partícula se inspira na melhor solução ao seu redor (BERGH, 2001; COHEN e CASTRO, 2006).

Na versão *gbest* do PSO, $\hat{y}$ representa a melhor posição que foi encontrada dentre todas as partículas do enxame. Bergh (2001) explica que essa versão fornece uma taxa de convergência mais rápida, mas que, no entanto, é menos robusta. Na versão *lbest*, a população é dividida em vizinhanças, e $\hat{y}_j$ representa a melhor posição encontrada na vizinhança da partícula $j$ , para uma população de tamanho $s$ e vizinhança de tamanho $l$. A definição de $\hat{y}$, como é utilizado na versão *gbest*, é representada pela Equação (14).

$$\hat{y}_i(t) \in \{y_0(t), y_1(t), \ldots, y_s(t)\} \mid f(\hat{y}_i(t+1)) = min\{y_0(t), y_1(t), \ldots, y_s(t)\}, \quad (14)$$

onde $f$ é a função objetivo do problema. A melhor posição global é aquela que possui menor valor da função objetivo.

A posição da partícula é atualizada usando o novo vetor velocidade de acordo com a seguinte equação:

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (15)$$

A melhor posição pessoal da partícula $i$ é calculada como:

$$x(t+1) = \begin{cases} y_i(t) & \text{se } f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) & \text{se } f(x_i(t+1)) < f(y_i(t)) \end{cases}$$

(16)

O algoritmo consiste em repetidas aplicações das equações de atualização de posição apresentadas (Figura 5). A ação de alterar a posição de cada partícula simula o movimento de um pássaro no espaço, que é influenciado pela própria memória da partícula, ou melhor posição pessoal (*personal best position*), e pela partícula que está melhor posicionada, ou melhor posição global (*global best position*). O algoritmo segue a heurística que a melhor resposta sempre se encontra próxima à melhor posição global encontrada por uma partícula, por isso, todas as outras tendem a procurar a melhor solução próxima ao *gbest*.

A inicialização da população de colônia normalmente é obtida com as partículas dispostas aleatoriamente sobre o espaço de projeto, $[-x_{max}, x_{max}]$. As velocidades $v_{i,j}$ geralmente também são inicializadas aleatoriamente em um intervalo $[-v_{max}, v_{max}]$. O critério de parada pode ser um número determinado de iterações ou outro critério dependendo do problema (BERGH, 2001).

### 2.3.4 Classificação

Para Kennedy e Eberhart (1995), criadores do PSO, a classificação do algoritmo parece estar em algum lugar entre a Programação Evolutiva e o Algoritmo Genético.

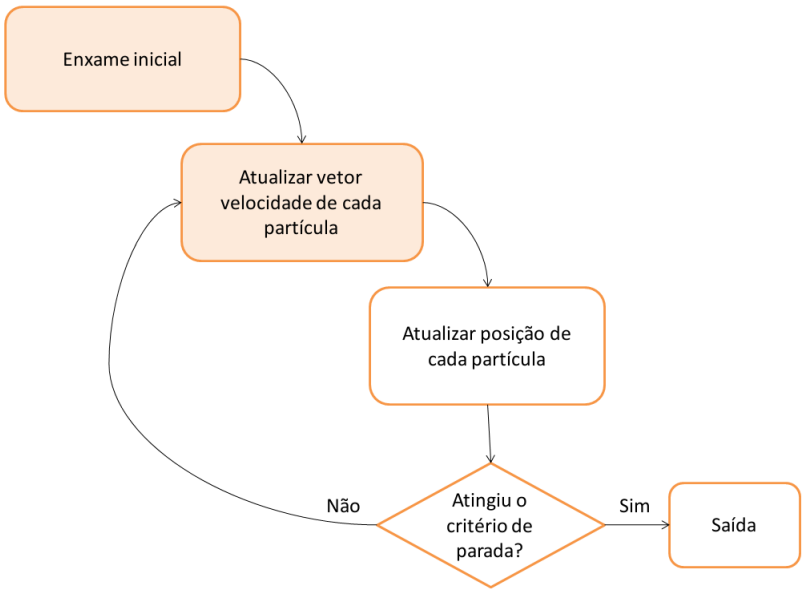


Figura 5: Fluxograma para o algoritmo PSO básico

Programação Evolutiva tem como ideia principal utilizar a simulação da evolução para desenvolver inteligência artificial que não dependesse de heurísticas, mas ao invés disso, gerasse organismos de intelecto crescente (PORTO, 1997). Assim como na Programação Evolutiva, o PSO é dependente de processos estocásticos. A ideia básica do Algoritmo Genético consiste em dada uma população de indivíduos, a pressão do ambiente provoca uma seleção natural, o que causa o aumento no *fitness* , ou aptidão, da população. Com isso, o ajuste da posição das partículas em relação à sua melhor posição pessoal e à melhor posição da vizinhança pode ser comparado com a operação de sobrevivência do mais apto no Algoritmo Genético. Assim como na Computação Evolutiva, o conceito de *fitness* também é usado. Único ao PSO é o vôo das soluções potencias pelo espaço de busca acelerando em direção às melhores soluções (KENNEDY & EBERHART, 1995).

O PSO pode ser visto como um processo de adaptação ao invés de substituição das populações anteriores. Bergh (2001) diz que isso deixa a diferença entre o PSO e os Algoritmos Evolutivos mais clara. O PSO mantém informações sobre a posição e velocidade, enquanto os Algoritmos Evolutivos tradicionais apenas rastreiam a posição.

# 3 Metodologia

## 3.1 PSO Usado para Clusterização de Dados

No método proposto por Merwe e Engelbrecht (2003), cada partícula do PSO é composta por um vetor de centróides de tamanho Nc, onde Nc é o número de grupos a serem criados, e representa uma solução completa para o problema. A cada iteração, os dados são atribuídos ao centróide ao qual estão mais próximos, as soluções são avaliadas e então atualizadas de acordo com sua melhor posição no espaço de busca e a melhor posição já encontrada por alguma partícula.

O método proposto por Merwe e Engelbrecht (2003) é o mais simples e também aquele que mais se adéqua às ideias do PSO, onde cada indivíduo representa uma possível solução do problema. Por isso, decidiu-se trabalhar sobre esse método.

De acordo com Merwe e Engelbrecht (2003), cada partícula $x_i$ é tido como um conjunto de *clusters*, pois só assim, cada particula representaria uma resposta completa da Clusterização. Em outras palavras, no enxame do PSO, cada partícula possue um vetor de $k$ centróides que são construídos da seguinte forma:

$$x_i = (m_{i1}, m_{i2}, \ldots, m_{ij}, \ldots, m_{iN_c},),$$

onde $N_c$ é o número de *clusters* a serem criados em $m_{i,j}$ corresponde ao $j^{-ésimo}$ centróide da $i^{-ésima}$ partícula. Assim, uma única partícula representa um candidato à solução do problema de Clusterização.

Cada partícula é avaliada através da seguinte equação denominada Função Objetivo:

$$J_e = \frac{\sum_{j=1}^{N_c} [\sum_{\forall Z_p \in c_{i,j}} d(z_p.m_{i,j}) / \mid C_{i,j} \mid]}{N_c} \tag{17}$$

Onde $Z_p$ é o $p^{-ésimo}$ dado, $\mid C_{ij} \mid$ é o número de dados pertencentes ao *cluster* $C_{ij}$ e $d$ é a distância euclidiana entre $Z_p$ e $m_{ij}$.

O algoritmo proposto por Merwe e Engelbrecht (2003) pode ser descrito da seguinte forma:

1. Inicializar o valor dos centróides de cada partícula aleatoriamente
2. Para t = 1 até $t_{max}$ faça
   - $(a)$ Para cada partícula $i$ faça
   - $(b)$ Para cada dado $z_p$
     - $i$. calcule a distância Euclidiana $d(z_p, m_{ij})$ para todos os centróides $m_{ij}$
     - $ii$. atribua $z_p$ ao *cluster* $C_{ij}$tal que $d(z_p, m_{ij}) = min_{\forall k=1...N_c}\{d(z_p, m_{ij})\}$
     - $iii$. Calcule o *fitness* da partícula utilizando a função objetivo
   - $(c)$ Atualize o melhor global e os melhores locais
   - $(d)$ Atualize os centróides dos *clusters*

## 3.2 *Hybrid Particle Swarm Optimization with Mutation*

Muitos problemas na área da engenharia envolvem a minimização de uma função. Dessa forma, é interessante estudar algoritmos de otimização eficientes. Algoritmos Genéticos (AGs) são uma família de modelos computacionais que é

inspirada na evolução de seres vivos. Esses algoritmos codificam uma potencial solução para um problema específico em um cromossomo simples, como estrutura de dados e aplica operadores de recombinação a estas estruturas de modo a procurar uma solução ótima para o problema (ESMIN *et al.* 2006).

Em comparação com AGs, o PSO tem a ideia fundamental muito mais inteligente e pode ser aplicado com mais facilidade. De acordo com Esmin *et al.* (2006), o PSO tem sido amplamente aplicada na otimização da função, formação de redes neurais artificiais, controle difuso e alguns outros campos. Desde então, vários melhoras no algoritmo do PSO têm sido desenvolvidas.

Alguns algoritmos híbridos PSO foram propostos adicionando a idéia do Algoritmo Genético, como o trabalho de Angeline (1998) e Løvbjerg *et al.* (2001). O método seleciona um determinado número de partículas de acordo com uma certa probabilidade em cada iteração. As partículas são aleatoriamente combinadas em pares. Cada casal produz dois filhos por *crossover*. Em seguida, os filhos substituem os pais, a fim de manter o número de partículas inalterado.

O trabalho de Esmin *et al.* (2006) propõe um novo modelo chamado *Hybrid Swarm Optimizer with Mutation* (HPSOM), que tem como objetivo resolver o problema de estagnação e impedir que as partículas fiquem presas em mínimos locais. Para resolver este problema, o HPSOM integra o processo de mutação muitas vezes utilizado em GA no PSO. A cada iteração, as partículas tem uma probabilidade, que é escolhida pelo usuário, de sofrer mutação. O processo de mutação é empregado utilizando a seguinte equação:

$$mut(p[k]) = p([k]x - 1) + \omega \tag{18}$$

Onde $p[k]$ é uma partícula escolhida aleatoriamente do enxame e $\omega$ é um valor, também aleatório, que está no intervalo $[0, 0.1x(x_{max} - x_{min})]$, que representa 10% do espaço de busca.

Simulações em uma série *benchmarks* de teste mostram que os algoritmos propostos híbrido possuem uma melhor capacidade de encontrar o ótimo global do que algoritmo PSO padrão.

O presente trabalho, no entanto, testa o método HPSOM aplicado ao problema de Clusterização, o qual ainda não existe na literatura.

### 3.2.1 Algoritmos Híbridos

De acordo com Merwe e Engelbrecht (2003), o método *k-means* converge para o resultado mais rapidamente que o *Particle Swarm Optimization*. Em outras palavras, o algoritmo *k-means* alcança a convergência em poucas iterações. Entretanto, na maioria dos casos, o PSO encontra grupos que possuem maior *fitness*.

Em seu trabalho, Merwe e Engelbrecht (2003) mostram que o desempenho do algoritmo PSO para Clusterização de dados pode ser melhorado. Isto pode ser conseguido fornecendo no enxame inicial o resultado do algoritmo *k-means* a fim de obter pelo menos uma partícula próxima ao melhor particionamento. O resultado do método *k-means* é utilizado como uma das particulas, enquanto o resto do enxame é inicializado aleatoriamente e a partir daí, buscam refinar o resultado trazido pelo *k-means*. O método híbrido utilizando PSO e *k-means* foi chamado de PSOKM.

O presente trabalho faz uso da proposta de Merwe e Engelbrecht (2003), chamando-o de PSOKM. Além disso, é proposto uma nova abordagem para a solução do problema de Clusterização, um algoritmo que usa a mesma ideia de hibridização do trabalho de Merwe e Engelbrecht (2003) (PSOKM), mas que, no

entanto, utiliza o HPSOM ao invés do PSO clássico. Esta abordagem busca a união de duas abordagens, tanto a proposta por Esmin *et al.* (2005) quanto a proposta por Merwe e Engelbrecht (2003), a fim de obter melhores resultados que ambos. Esta combinação de métodos foi chamado neste trabalho de HPSOMKM (*Hybrid Particle Swarm Optimizatiom with Mutation and K-Means*) e é ilustrada na Figura 6.

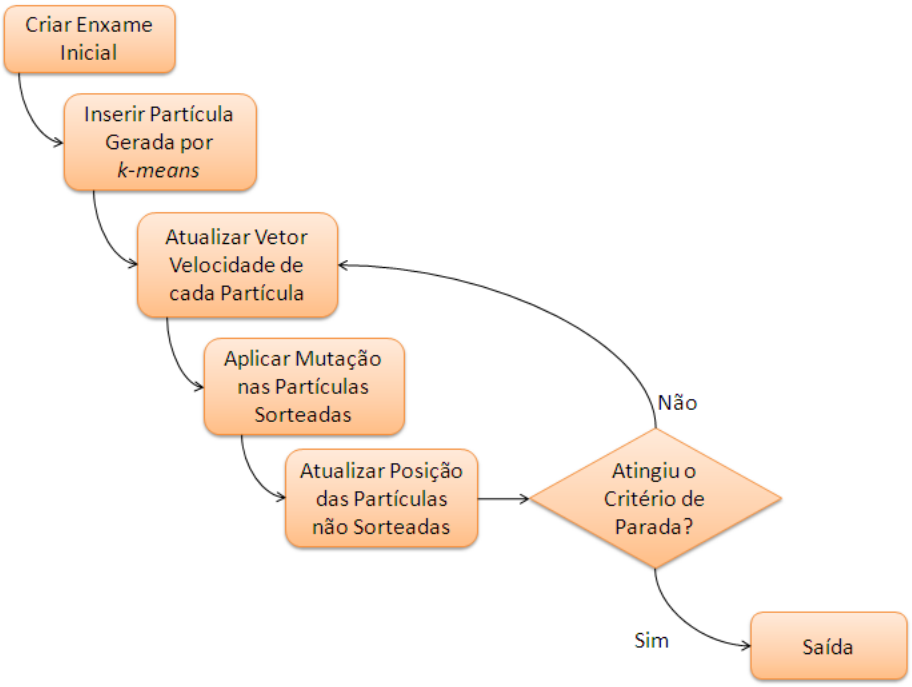


Figura 6: Fluxograma para o algoritmo HPSOMKM

# 4 Resultados

## 4.1 Implementação

O aplicativo, juntamente com os algoritmos, tanto puros quanto híbridos, foram implementados utilizando a linguagem de programação Java, com o intuito de obter maior velocidade de desenvolvimento devido à grande quantidade de algoritmos a ser implementado. O programa criado é orientado a objetos, dando assim uma melhor estrutura ao código, pois é interessante que seja de fácil manutenção e possua boa extensibilidade, ou seja, que suporte a introdução de novas ideias, algoritmos e até mesmo outras combinações de métodos.

O programa (Figura 7) permite que os parâmetros do método sejam alterados, além da escolha do algoritmo a ser utilizado, o conjunto de dados que se deseja usar e o número de vezes que se deseja executar o algoritmo, para que se possa obter o resultado médio de várias execuções (Figura 8). É capaz de exibir, em duas dimensões, os dados e os grupos gerados pelo método, onde os dados pertencentes ao mesmo grupo possuem a mesma cor e forma (Figura 9). Além disso, é possível executar o algoritmo escolhido passo a passo, para que seja possível entender o comportamento das partículas ao longo das interações.

Ao final de sua execução, o programa exibe a taxa de acerto e o *fitness* da melhor partícula em cada execução do algoritmo escolhido e, em seguida, a taxa média do método para o número de execuções escolhido, juntamente com desvio padrão e variância (Figura 10).

Foram criadas as classes: *Particle*, que representa uma partícula, onde o *Swarm* é, simplesmente, um conjunto de objetos da classe *Particle*; *Centroid*, que representa um dos centroides que está contido em cada partícula; *Velocity*, que representa a velocidade de cada partícula em cada uma de suas $n$ dimensões, ou

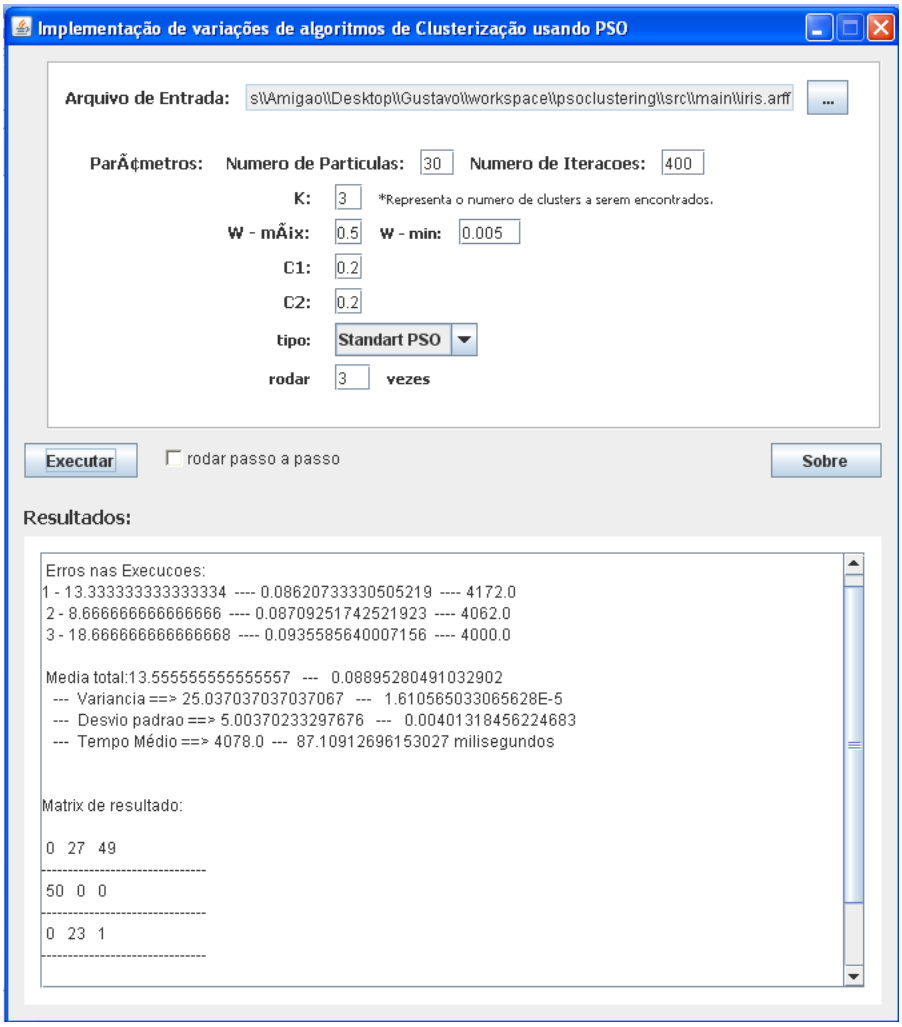


Figura 7: O Aplicativo

seja, cada partícula têm uma instância da classe *Velocity* para dimensão cada que possui; *DataVector*, que representa um elemento do conjunto de dados.

Foi criado, também, uma classe abstrata chamada *PSO*, que possui todos os métodos e atributos básicos para a execução do método. Assim, todas as combinações de algoritmos que usam o *Particle Swarm Optimization* teriam que es-

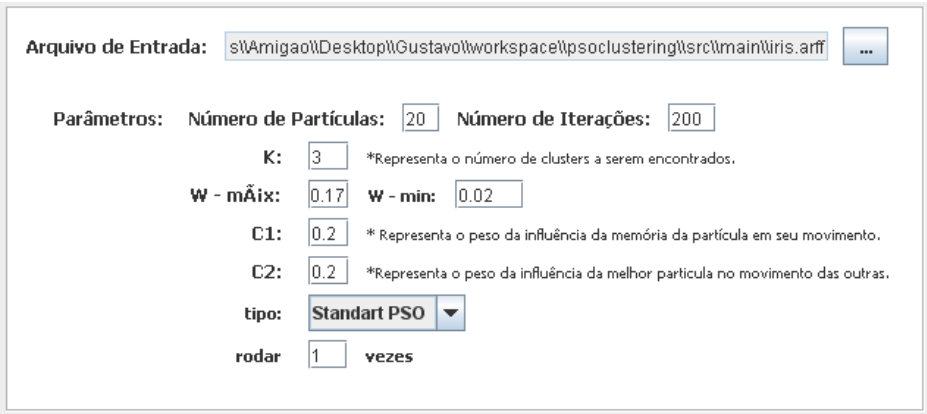


Figura 8: Parâmetros do Algoritmo

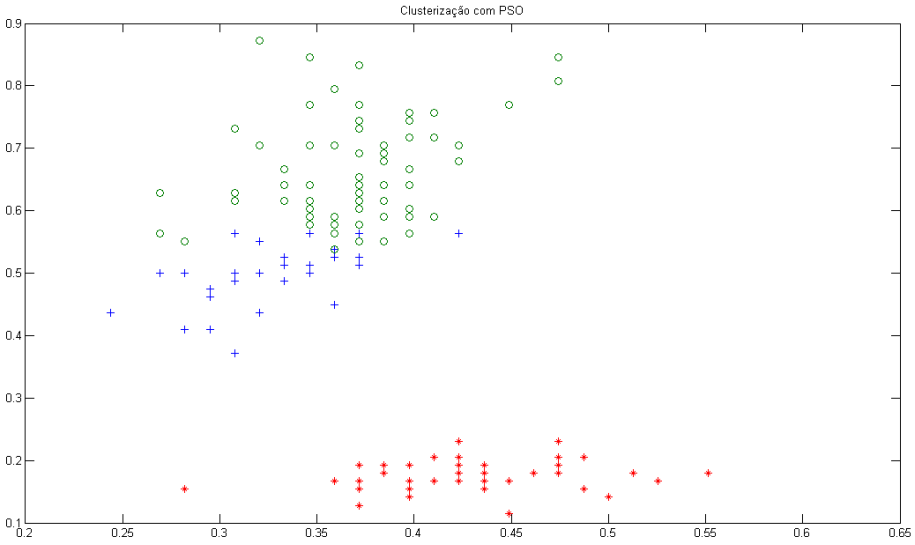


Figura 9: Gráfico do Resultado da Clusterização do *benchmark Iris*

tender esta classe. As classes que estendem PSO são *StandartPSO*, que executa o PSO padrão; *PSOKMeans*, que representa o algoritmo híbrido de PSO e *k-means*; *PSOKHamonicMeans*, que representa o algoritmo híbrido de PSO e *k-harmonic means*; *HPSOM* que realiza o *Hybrid Particle Swarm Optimization with Muta-*

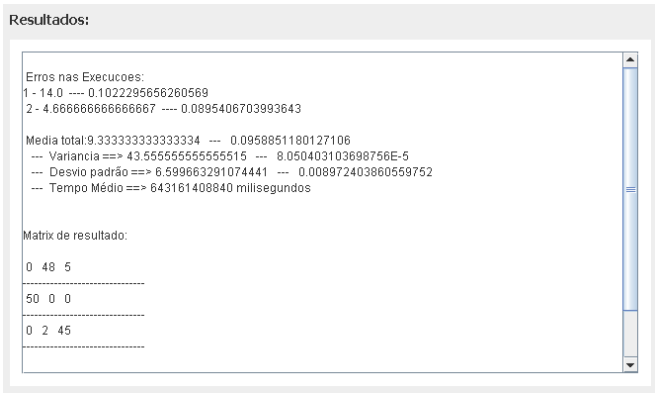


Figura 10: Resultados da Implementação

*tion*; e por último *HPSOMKMeans* que representa o algoritmo híbrido de HPSOM e *k-means*, HPSOMKM.

Para critério de comparação com os algoritmos híbridos, também foram implementados os algoritmos *k-means*, na classe *SimpleKMeans*, e *k-harmonic means* na classe *KHarmonicMeans*, que não utilizam PSO.

## 4.2 Resultados

Para avaliar e comparar os métodos, foram usados três benchmarks amplamente abordados na literatura: *Iris*, *Glass* e *Wine* (ASUNCION & NEWMAN, 2007). Todos podem ser encontrados no UCI *Repository of Machine Learning Databases* e suas caracteristicas estão descritas na Tabela 1. O *benchmark Iris* apresenta cento e cinquenta instâncias da flor Iris, divididas em três classes com cinqüenta instâncias cada. A primeira classe corresponde ao tipo Setosa, a segunda ao tipo Versicolour e a terceira Virginica. As instâncias possuem quatro atributos de valores reais, *sepal length* (comprimento da sépala), *sepal width* (largura da sépala), *petal length* (comprimento da pétala), *petal width* (largura da pétala). Uma

| *benckmark* | Número de Instâncias | Número de Atributos | Número de Classes |
|---|---|---|---|
| *Iris* | 150 | 4 | 3 |
| *Wine* | 178 | 13 | 3 |
| *Glass* | 214 | 9 | 7 |

Tabela 1: Características dos *benchmarks*

das classes (Setosa) é linearmente separável das outras duas, que não são linearmente separáveis entre si.

O *benchmark Wine* possui cento e setenta e oito instâncias de vinho. Esses dados são resultados de uma análise química realizada em vinhos da mesma região da Itália, mas vindos de diferentes cultivares. A análise determinou as quantidades de treze constituintes encontrados em cada um dos três tipos de vinho. Os atributos são: *alcohol , malic acid, ash, alcalinity of ash, magnesium, total phenols, flavanoids, noflavanoid fenols, proanthocyanins, color intensity, hue, OD280/OD315 of diluted wines, praline*. A primeira classe contém cinqüenta e nove instâncias, a segunda classe contém setenta e uma e a terceira quarenta e oito.

O *benchmark Glass* apresenta duzentas e quatorze instâncias de vidros divididas em sete tipos. Setenta instâncias em *building windows float processed*, dezessete em *vehicle windows float processed*, setenta e seis em *buiding windows non-float processed*, nenhuma em *vehicle windows non-float processed*, treze em *containers*, 9 em *tableware*, vinte e nove em *headlamps*. Os atributos dos dados são: *refrative index, sodium, magnesium, aluminium, silicon, potassium, calcium, barium, iron*.

Para cada conjunto de dados, o programa foi executado por 100 vezes, com 400 iterações, 30 partículas, $w$ variando entre 0.5 e 0.005, $c1 = 0.2$ e $c2 = 0.2$, utilizando cada um dos métodos. Após uma grande quantidade de testes, foram escolhidos estes valores por obterem bons resultados em todos os métodos.

Cada execução é avaliada de acordo com o *fitness* da melhor partícula, ou seja, é calculada a média da distância entre cada elemento do conjunto de dados e o centroide ao qual ele pertence.

De acordo com Pereira (2007), para saber se determinado elemento foi agrupado de maneira correta, é preciso saber a que classe do *benchmark* o *cluster* encontrado pelo algoritmo corresponde. Dado que:

- $G = \{g_1, g_2, \ldots, g_k\}$ é o conjunto dos grupos gerados pelo algoritmo.
- $B = \{b_1, b_2, \ldots, b_k\}$é o conjunto de classes do *benchmark*.
- $r_i$ é a classe representada pelo grupo $i$.
- $n_{ij}$ é o grupo de dados da classe $j$ no grupo $i$.

O algoritmo de mapeamento de *clusters* pode ser descrito da seguinte forma:

**Enquanto** $G \neq \varnothing$

**Encontrar** maior $n_{ij} \mid g_i \in G$ e $b_j \in B$
$r_i = j$
$G = G - g_i$
$B = B - b_j$

Dessa maneira, todos os dados que estiverem em $c_i$ e também pertencerem a classe $r_i$ estão agrupados corretamente, os outros estão agrupados incorretamente.

A Tabela 2 resume os resultados obtidos nos sete algoritmos de agrupamento para os *benchmarks* já descritos na questão da distância intracluster, ou seja, a média do valor do *fitness* da melhor partícula em todas as execuções e seu desvio

| Método | *Iris* | *Wine* | *Glass* |
|---|---|---|---|
| PSO | $0.08696 \pm 0.00484$ | $0.05903 \pm 0.00153$ | $0.01911 \pm 0.00108$ |
| KM | 0.08307 | 0.06155 | 0.01393 |
| KHM | 0.09961 | 0.07526 | 0.04285 |
| PSOKM | $0.08210 \pm 0.00203$ | $0.05820 \pm 0.00069$ | $0.01382 \pm 0.00026$ |
| HPSOM 10% | $0.08203 \pm 0.00289$ | $0.00289 \pm 0.00148$ | $0.01442 \pm 0.00123$ |
| HPSOM 5% | $0.08295 \pm 0.00320$ | $0.05838 \pm 0.00141$ | $0.01446 \pm 0.00116$ |
| HPSOMKM 10% | $0.08222 \pm \mathbf{0.00114}$ | $0.05810 \pm \mathbf{0.00074}$ | $0.01385 \pm \mathbf{0.00022}$ |
| HPSOMKM 5% | $\mathbf{0.08164} \pm 0.00250$ | $\mathbf{0.05801} \pm 0.00075$ | $\mathbf{0.01377} \pm 0.00037$ |

Tabela 2: Resultados dos testes experimentais

| Método | *Iris* | *Wine* | *Glass* |
|---|---|---|---|
| PSO | $4516.5 \pm 145.44$ | $11943.43 \pm 115.73$ | $20274.5 \pm 254.18$ |
| KM | $\mathbf{6.75 \pm 7.91}$ | $\mathbf{33.18 \pm 7.83}$ | $\mathbf{51.81 \pm 9.29}$ |
| KHM | $100.5 \pm 36.38$ | $775.5 \pm 362$ | $3193.37 \pm 2026.22$ |
| PSOKM | $4370.06 \pm 200.37$ | $11157.18 \pm 246.25$ | $18771.5 \pm 580.02$ |
| HPSOM 10% | $4514.75 \pm 223.71$ | $11244.25 \pm 264.12$ | $19143.56 \pm 288.55$ |
| HPSOM 5% | $4526.25 \pm 119.73$ | $11217.81 \pm 417.12$ | $18730.5 \pm 222.78$ |
| HPSOMKM 10% | $4426.81 \pm 36.77$ | $11103.37 \pm 122.08$ | $21744.25 \pm 452.76$ |
| HPSOMKM 5% | $4434.62 \pm 49.96$ | $11372.06 \pm 213.71$ | $21022.56 \pm 179.47$ |

Tabela 3: Tempo de execução dos testes experimentais

padrão ($f \pm \sigma$) para cada um dos métodos descritos neste trabalho. Vale destacar que nos algoritmos que possuem mutação, foram testados dois tipos de porcentagem de mutação, 5% e 10%. Em outras palavras, os algoritmos HPSOM 5%, HPSOM 10%, HPSOMKM 5% e HPSOMKM 10% na Tabela 2 são os testes dos algorítmos HPSOM e HPSOMKM em 5% e 10% de mutação, respectivamente.

O tempo de execução médio de cada método, em 100 execuções, foi medido em milissegundos e é apresentado na Tabela 3, que também o desvio padrão, para a melhor análise dos dados.

Observando a Tabela 3, é fácil perceber que método *k-means* executa muito mais rapidamente que todos os outros algoritmos testados neste trabalho.

Na Tabela 2, KM conseguiu bons resultados, muito competitivos em relação ao PSO em todos os *benchmarks* testados, e sempre chega ao mesmo resultado devido a não ser um método estocástico.

O método *k-harmonic means*, apesar de também ser bastante rápido, apresentou os piores resultados entre todos os testes. Assim, a ideia de propor um algoritmo híbrido usando *k-harmonic means* não faz sentido, pois é melhor utilizar o *k-maens*, que executa mais rápido e possui melhores resultados experimentais.

Nos testes com *Iris* e *Wine*, PSO obteve resultados bem competitivos, se comparados com KM, inclusive nos testes com *Wine*, PSO obtém melhores resultados que KM. No entanto, PSO apresentou resultados muito inferiores que KM nos testes com *Glass*.

Os métodos HPSOM 5% e HPSOM 10% obtiveram melhores resultados que o PSO em todos os testes e valores muito próximos entre si, com exceção dos testes feitos com *Wine*, onde o HPSOM 10% obteve resultados largamente melhores que HPSOM 5%. Também, conseguiram melhores resultados que KM no *Iris* e *Wine*. Além disso, HPSOM 5% e HPSOM 10% conseguiram menor desvio padrão que o PSO, o que significa maior confiabilidade no método.

PSOKM, como dito por Merwe e Engelbrecht (2003), de fato, refinou o resultado do KM, ou seja, nenhum dos resultados do método *k-means* foi melhor que PSOKM. Além disso, PSOKM foi melhor que os métodos HPSOM 5% e HPSOM 10% nos testes com *Wine* e *Glass* e , também, possui menos desvio padrão.

Por fim, o método HPSOMKM obteve os melhores resultados deste trabalho tanto na distância intracluster quanto no desvio padrão. HPSOMKM 5% conseguiu os melhores *clusters* (Figura 11 ) e HPSOMKM 10% obteve o melhor desvio padrão.

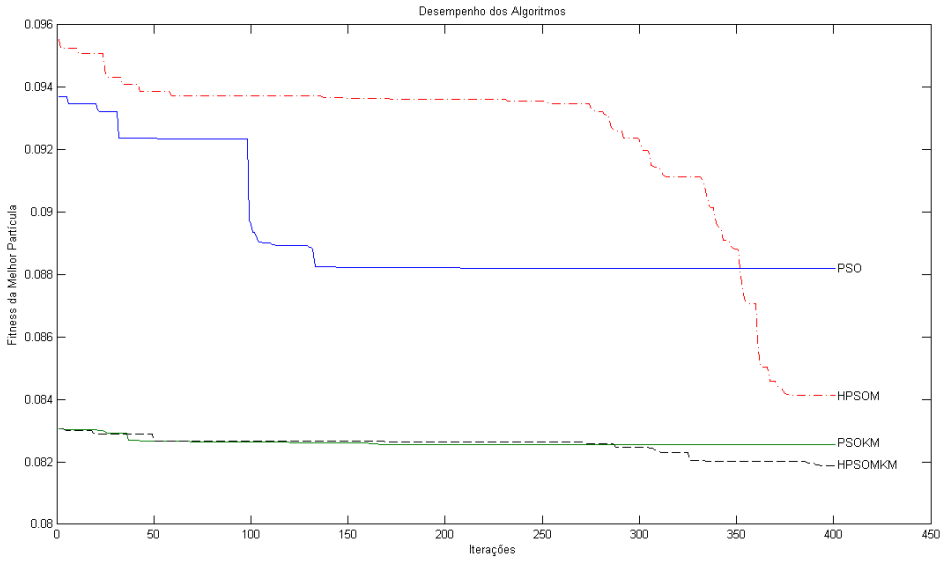


Figura 11: *Fitness* da melhor partícula ao longo das iterações (valores multiplicados por 100, para se obter melhor visibilidade)

A Tabela 3 nos mostra que à medida que vamos combinando algoritmos, o tempo de execução aumenta proporcionalmente. Contudo, há uma melhora na Clusterização, que é um problema muito complexo.

# 5 Conclusão e Trabalhos Futuros

Nesse trabalho, foi apresentado um estudo do problema de Clusterização, uma das principais tarefas de descoberta de conhecimento em bancos de dados e aplicado em diversas áreas. Foi visto que as técnicas para a resolução do problema de *Clustering* podem ser divididas em: Métodos por Particionamento, Métodos Hierárquicos, Métodos Baseados em Densidade, Métodos Baseados em Grid, Métodos Baseados em Modelos.

Adiante, foi apresentado o PSO, *Particle Swarm Optimization*, um algoritmo baseado em comportamento social que tem sido aplicado com sucesso em diversos problemas. Foram apresentados alguns métodos de resolução do problema de Clusterização baseados no PSO que tem surgido na literatura nos últimos anos. Dentre eles, o método desenvolvido por Merwe e Engelbrecht (2003), um método simples e aderente às ideias do PSO, foi implementado e seu funcionamento estudado um pouco mais a fundo. Então, foram propostos e testados uma série algoritmos híbridos, que usam a ideia de Merwe e Engelbrecht (2003) em conjunto com *k-means*, *k-harmonic means* e um esquema de mutação proposto por Esmim (2006), para a Clusterização de dados.

Três *benchmarks* conhecidos da área foram usados para comparar a eficiência desses métodos. O HPSOM demonstrou-se capaz de encontrar boas soluções para o problema, melhores que todas as outras testadas neste trabalho, principalmente em *clusters* com formato circular. Os resultados demonstram que é interessante utilizar técnicas híbridas para resolver problemas de *Clustering*, pois foram encontrados resultados muito superiores.

Em trabalhos futuros, deseja-se o desenvolvimento de um método de Clusterização, baseado no PSO, que obtenha bons resultados na Clusterização textual, para que possa ser utilizado como ferramenta de *Clustering* de páginas *Web*.

# 6 Referencia Bibliográfica


AGRAWAL, R., GEHRKE, J., GUNOPULOS, D., RRAGHAVAN, P. **Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications**, ACM SIGMOD Int'l Conf. on Management of Data, Seattle, Washington, June 1998.

ANGELINE, P. J. **Evolutionary Optimization Versus Particle Swarm Optimization: Philosophy and Performance Differences.** Evolutionary Programming VII (1998) Lecture Notes in Computer.

ANKERST, M., BREUNIG, M., M., KRIEGEL, H.-P., et al. **OPTICS: Ordering Points to Identify the Clustering Structure**, In: Proceedings of the ACM SIGMOD Conference on Management of Data, pp. 49-60, Philadelphia, PA, USA, June, 1999.

ASUNCION, A.; NEWMAN, D.J. . **UCI Machine Learning Repository** [http://www.ics.uci.edu/ mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science, 2007.

BERGH, F. van der. **An Analysis of Particle Swarm Optimizers**. 300 p. Tese (PhD in the Faculty of Natural and Agricultural Science) – University of Pretoria, Pretoria, 2001.

BRAGA, L. P. V. **Introdução à Mineração de Dados**. 2ª edição revisada e ampliada. Rio de Janeiro: E-Papers Serviços Editoriais, 2005.

CARLANTONIO, L. M. **Novas Metodologias para Clusterização de Dados, Coordenação de Programas de Pós-Graduação em Engenharia**, COPPE/UFRJ, 2001.

COHEN, S. C. M.; CASTRO, L. N. de. **Data Clustering with Particle Swarms.** In: Congress on Evolutionary Computation, 2006. Proceedings of IEEE Congress on Evolutionary Computation 2006 (CEC 2006). Vancouver: IEEE Computer Society, 2006. p. 1792- 1798.

COLE, R. M., **Clustering with Genetic Algorithms**, M. Sc., Department of Computer Science, University of Western Australia, Australia, 1998.

ESMIN, A. A. A. ; TORRES, Germano Lambert ; ALVARENGA, G. B. . **Hybrid Evolutionary Algorithm Based on PSO and GA Mutation.** In: 6th International Conference on Hybrid Intelligent Systems 4th Conference on Neuro-Computing and Evolving Intelligence. Auckland - Nova Zelândia : HIS-NCEI Press, 2006. p. 57-60.

ESTER, M., KRIEGEL, H.-P., SANDER, J., et al, **Incremental Clustering for Mining in a Data Warehousing Environment**, In: Proceedings of the 24th International Conference on Very Large Data Bases (VLDB), pp. 323-333, New York City, New York, USA, August, 1998

GÜNGÖR, Z., ÜNLER, A. **K-harmonic Means Data Clustering with Simulated Annealing Heuristic** Applied Mathematics and Computation, Volume 184, Issue 2, 15 January 2007, Pages 199-209, ISSN 0096-3003, DOI: 10.1016/j.amc.2006.05.166. (http://www.sciencedirect.com/science/article/B6TY8-4KMYKYX-2/2/8857941a2f5c05095eaccc73c34a9930)

HAMMERLY, G., ELKAN, C. (2002). **Alternatives to the K-means Algorithm that Find Better Clusterings.** In: Proceedings of the 11th international conference on information and knowledge management (pp. 600–607).

HAN, J., KAMBER, M. **Data Mining: Concepts and Techniques**. San Francisco: Morgan Kaufmann, 2001. 550 p.

JIANG, H., YI,S., Li, J., YANG, F., HU, Y. **Ant clustering algorithm with K-harmonic means clustering** In: Expert Systems with Applications, Volume 37, December 2010, Pages 8679-8684.

KENNEDY, J., EBERHART, R. C. **Particle Swarm Optimization.** In: IEEE International Conference on Neural Networks, 1995, Perth. Proceedings of IEEE International Conference on Neural Networks. 1995, p. 1942-1948.

LΦBJERG, M., RASMUSSEN, T. K., KRINK, T. **Hybrid Particle Swarm Optimiser with Breeding and Subpopulations.** Proceedings of the Genetic and Evolutionary Computation Conference (GECCO). San Francisco, USA, July 2001.

MACQUEEN, J. **Some methods for classification and analysis of multivariate observations**. In: L. M. Le Cam J. Neyman [eds.] Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Vol. 1. University of California Press, Berkeley, 1967.

MERWE, D. W., ENGELBRECHT, A. P. **Data Clustering using Particle Swarm Optimization.** In: Congress on Evolutionary Computation, 2003. Proceedings of IEEE Congress on Evolutionary Computation 2003 (CEC 2003), Caribella: IEEE Computer Society, 2003. p. 215-220.

PORTO, V. W. **Evolutionary Programming**. In: BÄCK, T.; FOGEL, D. B.; MICHALEWICZ, Z. .Handbook of Evolutionary Computation. 1. ed. Bristol: IOP Publishing, 1997.Cap. B1.4, p. 1-10.

PEREIRA, D. L. **Estudo do PSO na Clusterização de dados**. Monografia apresentada na UFLA (Universidade Federal de Lavras), 2007.

REYNOLDS, R. G., PENG, B.,WHALLON, R. **Emergent Social Structures in Cultural Algorithms** Dept. of Computer Science Wayne State University, Detroit USA, 1996.

SARAMAGO, S. F. P., PRADO, J. R. 2005. **Otimização por Colônia de Partículas**. FAMAT em Revista (UFU), v. 04, p.inicial 87, p.final 103, ISSN: 1806-1958, Meio magnético. www.famat.ufu.br.

SILVA, M. P. S. **Mineração de Dados - Conceitos, Aplicações e Experimentos com Weka**. Universidade do Estado do Rio Grande do Norte (UERN) e Instituto Nacional de Pesquisas Espaciais (INPE), 2000.

ZHANG, B., HSU, M., DAYAL, U. **K-Harmonic Means -A Data Clustering Algorithm.** Research Laboratory June 28, 1999