META-HEURÍSTICA VNS APLICADA A PROBLEMAS DE AGRUPAMENTOS

Dalila Ribeiro Serpa
Instituto Nacional de Pesquisas Espaciais
Av. dos Astronautas, 1.758, Jd. Granja - CEP: 12227-010
São José dos Campos – SP, Brasil
Email: dalila.serpa@lac.inpe.br

Antonio Augusto Chaves
Faculdade de Engenharia de Guaratinguetá – UNESP
Av. Dr. Ariberto Pereira da Cunha, 333 – CEP: 12516-410
Guaratinguetá – SP, Brasil
Email: chaves@feg.unesp.br

Francisco de Assis Corrêa
Universidade Paulista – UNIP
CEP 12240-420 – São José dos Campos – SP, Brasil
Email: correa@unip.br

Luiz Antonio Nogueira Lorena
Instituto Nacional de Pesquisas Espaciais
Av. dos Astronautas, 1.758, Jd. Granja - CEP: 12227-010
São José dos Campos – SP, Brasil
Email: lorena@lac.inpe.br

RESUMO

Os problemas de agrupamentos surgiram da necessidade de se agrupar dados a fim de entender um objeto ou um fenômeno ainda desconhecidos. O agrupamento é feito com base na similaridade entre os objetos de um conjunto dados, onde os mais similares ficam no mesmo grupo. Este trabalho propõe uma nova abordagem para problemas de agrupamentos através da meta-heurística *Variable Neighborhood Search* (VNS). O VNS é caracterizado por realizar buscas em vizinhanças distantes. Utilizando uma modelagem para particionamento em cliques, o algoritmo gerou partições para diferentes conjuntos de dados. Essas partições foram avaliadas por dois índices de validação: Rand e CRand. Além disso, os resultados do VNS foram comparados com outros algoritmos encontrados na literatura e pode ser considerado um algoritmo promissor para a solução de problemas de agrupamentos.

PALAVRAS CHAVE. Agrupamentos. VNS. Particionamento em cliques. Meta-heurísticas.

ABSTRACT

The clustering problems arose from the need to group data in order to understand an object or a phenomenon still unknown. The clustering data is based on similarity between objects of a data set, where the most similar objects are in the same group. This paper presents a new approach to clustering problems with the *Variable Neighborhood Search* (VNS) meta-heuristic. The VNS is characterized by performing searches in a distant neighborhoods. Using a model for clique partitioning, the algorithm generated partitions to different data sets. Those partitions were evaluated with two validation indexes: Rand and CRand. Moreover, the results of VNS were compared with other algorithms in the literature and can be considered a promising algorithm for solving clustering problems.

KEYWORDS. Clustering. VNS. Clique partitioning. Metaheuristics.

1. Introdução

A todo o momento, pessoas de diferentes áreas de estudo encontram grande número de informações. Para facilitar a análise futura, essas informações são armazenadas como dados. Classificar ou agrupar esses dados é uma maneira de trabalhá-los a fim de descobrir conjuntos de categorias entre eles, assim entendendo um novo objeto ou um novo fenômeno (Xu e Wunsch, 2005). Dessa forma, surgem os problemas de agrupamentos.

Em uma definição geral, esses problemas são caracterizados pela necessidade de se agrupar objetos de um conjunto de dados de forma a manter os mais similares no mesmo grupo (*cluster*). Essa similaridade é representada por alguma característica que seja comum entre os objetos do mesmo *cluster*.

Para solucionar esses problemas, diferentes técnicas ou algoritmos de agrupamento são encontrados na literatura. Segundo Bárbara (2000) e Jain et al (1999), esses algoritmos são divididos em dois tipos principais: hierárquico e particional. A diferença entre eles está nas estruturas resultantes de cada um, onde os hierárquicos geram uma sequência aninhada de partições e os particionais uma única partição dos dados.

Existem também os algoritmos de agrupamento baseados em teoria dos grafos (Barbara, 2000; Xu e Wunsch, 2005). Entre os mais conhecidos estão os algoritmos de partição em cliques (Mehrotra, 1998; Kochenberger et al, 2005; Amorim et al, 1992). O funcionamento deles é baseado no particionamento de um grafo em cliques (subgrafos completos), onde o número de cliques é igual ao número de *clusters* encontrados.

Outra abordagem interessante é trabalhar um problema de agrupamentos como um problema de otimização (Barbara, 2000). A idéia é aplicar um algoritmo baseado em um modelo matemático onde o objetivo seja maximizar ou minimizar o valor de uma função. Este algoritmo heurístico busca encontrar a melhor solução dentre muitas geradas por ele, mas não garante que a melhor solução encontrada realmente seja solução ótima, pois assim como pode chegar próximo ao ótimo também pode tê-lo encontrado. Um exemplo é o algoritmo *k*-Médias que tenta minimizar o erro quadrático entre os objetos e seus centros de *cluster* (em outras palavras, tenta minimizar a soma das distâncias entre os objetos e seus centros de *cluster*). Apesar de ser um dos algoritmos mais utilizados para este fim, quando há um grande número de objetos no conjunto de dados, o tempo computacional necessário para se chegar à melhor solução se torna bastante elevado (Chang et al, 2009). Este fato motivou os estudos de aplicações de meta-heurísticas (Chang et al, 2009; Nascimento et al, 2010; Al-Sultan, 1995; Amorim et al, 1992) a esses problemas, as quais apresentaram resultados semelhantes e até melhores que o *k*-Médias (entre outros algoritmos) em menor tempo computacional.

Dois trabalhos de destaque na área de meta-heurísticas são: Nascimento et al (2010) e Chang et al (2009). Em Nascimento et al (2010) é proposto um modelo matemático para solução de problemas de agrupamentos. O modelo é testado com a meta-heurística Procedimentos de busca gulosos, aleatórios e adaptativos (GRASP, do inglês *Greedy randomized adaptive search procedures*) (Feo e Resende, 1995). Os autores utilizam conjuntos de dados biológicos e avaliam os *clusters* gerados pelo GRASP com o índice de validação Corrected Rand (CRand) (Hubert e Arabie, 1985). Além disso, comparam os resultados com os algoritmos: *k*-Médias, *k*-Medianas (Kaufman e Rousseeuw, 1990) e Particionamento em torno de medianas (PAM, do inglês *Partitioning Around Medoids*) (Kaufman e Rousseeuw, 1990). Em Chang et al (2009), é proposto um rearranjo de genes ao Algoritmo Genético (Goldberg, 1989) chamado Algoritmo Genético com Rearranjo de Genes (GAGR, do inglês *Genetic Algorithm with gene rearrangement*), modelado para otimizar a mesma função objetivo do *k*-Médias. Os autores realizam testes em conjuntos de dados reais e também em imagens de sensoriamento remoto. Os *clusters* obtidos com o algoritmo são avaliados pelo índice de validação Rand (Rand, 1971) e comparados com os algoritmos: *k*-Médias, GA-clustering (Murthy e Chowdhury, 1996), KGA-clustering (Bandyopadhyay e Maulik, 2002).

Com base nos estudos de meta-heurísticas citados acima, este trabalho propõe uma nova abordagem para os problemas de agrupamentos através do algoritmo Busca em Vizinhança Variável (VNS, do inglês *Variable Neighborhood Search*), proposto inicialmente por Mladenovic e Hansen (1997). Neste trabalho, o VNS foi modelado para solucionar agrupamentos por meio do particionamento em cliques.

As soluções geradas pelo VNS foram avaliadas através de dois índices de validação: o Rand e o CRand. A função destes índices é mostrar, estatisticamente, quão corretos estão os agrupamentos. O cálculo dos dois índices é feito utilizando duas partições: a gerada pelo algoritmo de agrupamento e a que tem o agrupamento real. O valor de Rand e CRand varia entre [0,1], sendo que quanto mais próximo de 0, mais diferentes são as duas partições (os *clusters* encontrados em uma são diferentes dos encontrados na outra). E quanto mais próximo de 1, mais semelhantes são as partições (os *clusters* encontrados em uma são os mesmos encontrados na outra), constatando a combinação perfeita nas duas partições. O Rand não é corretamente encontrado quando as partições têm diferentes números de *clusters*, por exemplo, 2 *clusters* em uma e 5 *clusters* na outra. O CRand trata-se de uma normalização do Rand e pode ser calculado normalmente em duas partições que tenham números de *clusters* diferentes (Jain e Dubes, 1988). Para utilizá-los é necessário que os conjuntos de dados usados no experimento tragam a classificação real de cada objeto. O fato de usarem as classificações reais dos objetos é que caracteriza esses índices como validação externa.

O trabalho está organizado da seguinte forma: na seção 2, apresentamos o algoritmo, descrevendo seu funcionamento e seu pseudocódigo. Na seção 3, comentamos sobre os conjuntos de dados utilizados no experimento. Na seção 4, mostramos e analisamos os resultados dos experimentos computacionais. E por último, na seção 5, deixamos algumas considerações finais.

2. VNS aplicado a problemas de agrupamento

Tratar um problema de agrupamentos como problema de otimização tornou-se uma boa opção quando se tem um conjunto extenso de dados. Utilizar métodos determinísticos (algoritmos que garantem encontrar a solução ótima) seria o ideal, mas novamente encontraríamos problemas com relação ao tempo computacional que eles demandam.

As meta-heurísticas vêm sendo recentemente investigadas na resolução de agrupamentos, pois mesmo não garantindo encontrar a solução ótima, são capazes de chegar até ela (ou próximo a ela) em tempo computacional aceitável.

Baseando nessa idéia e também nos trabalhos de Nascimento et al (2010) e Chang et al (2009), este trabalho apresenta uma nova abordagem heurística por meio do VNS que é caracterizado por fazer trocas sistemáticas de vizinhança. Combinado com uma busca local, ele só aceita a solução se e somente se ela for a melhor encontrada. Aqui, o VNS trabalhará como um algoritmo de particionamento em cliques, baseado na formulação proposta em Nascimento et al (2010).

Para melhor entender seu funcionamento, imagine um conjunto de dados representado como um grafo $G = (V, E)$, onde V é o conjunto dos N objetos (representados por N nós) e E o conjunto de arestas não direcionadas que ligam o objeto i ao j. A distância entre os dois objetos define o valor do peso associado a cada aresta. Inicialmente, cada nó está ligado com todos os outros nós, formando um grafo completo.

No VNS o conjunto de nós do grafo é representado por um vetor $s = \{s_1, s_2, ..., s_N\}$, sendo N o número de objetos. A cada posição desse vetor é atribuída uma classe, numericamente representada de acordo com a quantidade de *clusters* definida *a priori*. Assim o objeto correspondente à posição do vetor pertencerá à classe atribuída a essa posição. Esse vetor será chamado de solução. A Figura 1, ilustra um exemplo de solução, com 5 objetos e 2 classes.

1	1	2	2	2
s_1	s_2	s_3	s_4	s_5

Figura 1 – Representação de uma solução

As soluções são avaliadas por uma função objetivo que calcula a soma das distâncias entre os objetos do mesmo *cluster*. Seja *N*, o número de objetos da solução s e d_{ij} a distância entre os objetos *i* e *j*, a função objetivo $f(s)$ é dada por:

$$f(s) = \sum_{i=0}^{N-1} \sum_{j=i+1}^{N} d_{ij} \qquad \forall\, s_i = s_j$$

onde, a soma da distância d_{ij} será feita quando $s_i = s_j$, ou seja, a distância só será somada se os *clusters* dos objetos *i* e *j* forem iguais. Neste trabalho, a função objetivo será minimizada.

Na fase inicial, o algoritmo gera a solução aleatoriamente, ou seja, atribui um *cluster* aleatório a cada posição da solução. Assim são geradas 100 soluções e apenas a que obtem menor valor da função objetivo é escolhida para ser a solução inicial do VNS. O Procedimento 1 mostra essa rotina.

A partir da solução inicial, o VNS gera um vizinho s' aleatório pertencente a uma vizinhança $V_p(s)$ e aplica uma busca local nele. Neste trabalho, o processo de geração da vizinhança para o VNS contará basicamente com dois movimentos: troca dos *clusters* entre dois objetos (*swap*) e mudança de um objeto de *cluster* (*shift*). As estruturas de vizinhança são geradas a partir dos movimentos realizados. Com isso, quanto mais movimentos a realizar, mais distantes serão as vizinhanças. A Função 2 ilustra o pseudocódigo da rotina que gera os vizinhos a partir das 6 estruturas de vizinhança ($p_{\max} = 6$) definidas pelos movimentos de *shift* e *swap*.

Procedimento 1 – Gerar solução inicial( )
Ler dados();
Inicializar solução inicial s;
Inicializar solução auxiliar $sAux$;
Seja *C* o número de *clusters* existentes;
Enquanto (condição de parada não satisfeita) **faça**
    Para cada posição de $sAux$ **faça**
        Atribuir randomicamente um *cluster* $c \in [1, C]$;
    fim-para;
    Avaliar s através da função objetivo $f(s)$;
    Se $f(sAux) < f(s)$ **então**
        $s \leftarrow sAux$;
    fim-se;
fim-enquanto;
fim.

A busca local realiza testes com todos os *clusters* em uma posição da solução, ou seja, testa o mesmo objeto em todos os *clusters* existentes, aceitando o *cluster* que obtiver menor valor na função objetivo. Essa posição é guardada para que não seja visitada novamente durante a mesma iteração do VNS. O Procedimento 3 mostra o pseudocódigo da busca local.

Por fim, o Procedimento 4 ilustra o pseudocódigo do VNS.

Função 2 – Gerar vizinhos (solução s , vizinhança *p*)

$s' \leftarrow s$;

Caso p **seja**:

  1: Trocar os *clusters* entre 1 par aleatório de objetos de em s';

  2: Mudar aleatoriamente 1 objeto de *cluster* em s';

  3: Trocar aleatoriamente os *clusters* entre 2 pares aleatórios de objetos em s';

  4: Mudar aleatoriamente 2 objetos de *cluster* em s';

  5: Trocar os *clusters* entre 3 pares aleatórios de objetos em s';

  6: Mudar aleatoriamente 3 objetos de *cluster* em s';

fim-caso;

Retorna s';

fim.

Procedimento 3 – Busca Local(solução s')

Atribua s' a uma solução s'';

Enquanto (condição de parada não satisfeita) **faça**

  Para cada *cluster* $c \in [1, C]$ faça

    Pegue uma posição aleatória de s'' e atribua *c* a ela;

    Guarde essa posição para que não seja visitada novamente;

    Se $f(s'') < f(s')$ **então**

      $s' \leftarrow s''$;

    Senão

      $s'' \leftarrow s'$;

  fim-para;

fim-enquanto;

fim.

Procedimento 4 – VNS(solução inicial s)

Seja $p_{\max}$ o número de estruturas diferentes de vizinhança;

Seja s^* a melhor solução encontrada;

$p \leftarrow 1$;

$s^* \leftarrow s$;

Enquanto (critério de parada não satisfeito) **faça**

  Gerar um vizinho s' da vinhança $V_p(s)$;

  Aplicar a Busca Local em s' obtendo um ótimo local s'';

  Se $f(s'') < f(s^*)$ **então**

    $s^* \leftarrow s''$;

    $p \leftarrow 1$;

  Senão

    $p \leftarrow p + 1$;

  fim-se;

  Se ($p > p_{\max}$) **então**

    $p \leftarrow 1$;

  fim-se;

fim-enquanto;

fim.

Com o algoritmo apresentado, foram realizados testes computacionais para oito conjuntos de dados encontrados na literatura.

3. Conjuntos de dados

Para testar o VNS, foram utilizados oito conjuntos de dados. Esses conjuntos foram divididos em dois grupos: A e B. Os conjuntos de dados são formados por objetos (também conhecidos como amostras, pontos ou padrões), cada objeto possui atributos (medidas ou variáveis) que caracterizam os objetos quantitativa ou qualitativamente. Além disso, os conjuntos de dados são divididos em diferentes números de *clusters*. Abaixo segue a descrição de cada grupo formado pelos conjuntos de dados:

- **Grupo A**

Este grupo é composto pelos conjuntos: Íris, Breast, Yeast e Proteínas. As três primeiras podem ser encontradas em Asuncion e Newman (2007) e a última em *http://ranger.uta.edu/~chqding/protein/*. A idéia é avaliar as soluções geradas para esse grupo com o índice CRand e assim comparar os índices resultantes com os apresentados no trabalho Nascimento et al (2010), onde podem ser encontrados maiores detalhes sobre cada conjunto de dados. A Tabela 1 ilustra a estrutura de cada conjunto, onde a primeira coluna, Conjunto de dados, mostra o nome dos conjuntos. A segunda coluna, Objetos, mostra a quantidade de objetos. A terceira coluna, Atributos, mostra a quantidade de atributos. E a última coluna mostra a quantidade de *clusters* existentes em cada conjunto de dados.

Tabela 1. Estrutura dos conjuntos de dados do grupo A

Conjunto de dados	Objetos	Atributos	*Clusters*
Yeast	1484	8	10
Breast	699	9	2
Proteínas	698	125	4
Íris	150	4	3

- **Grupo B**

Este grupo é formado por cinco conjuntos de dados: Wine, Glass, Balance e Liverdisorder. A quinta base é a Íris já descrita no grupo A. Esses conjuntos podem ser encontrados em Asuncion e Newman (2007). Estão alocados neste grupo porque suas soluções serão avaliadas pelo índice Rand e os resultados serão comparados com os apresentados no trabalho Chang et al (2009). A Tabela 2 mostra a estrutura de cada conjunto, onde a primeira coluna, Conjunto de dados, mostra o nome dos conjuntos. A segunda coluna, Objetos, mostra a quantidade de objetos. A terceira coluna, Atributos, mostra a quantidade de atributos. E a última coluna mostra a quantidade de *clusters* existentes em cada conjunto de dados.

Tabela 2. Estrutura dos conjuntos de dados do grupo B

Base de dados	Objetos	Atributos	*Clusters*
Wine	178	13	3
Glass	214	9	6
Balance	625	4	3
Liverdisorder	345	6	2

Com esses dados o VNS foi testado. Na próxima seção serão mostrados os resultados e a metodologia do experimento.

4. Experimento computacional

A metodologia e os resultados para cada grupo de conjuntos de dados são diferentes e por isso serão descritos em duas subseções.

Os índices Rand e CRand foram calculados no *software* R-Project desenvolvido para cálculos estatísticos. É gratuito e pode ser obtido em *http://www.r-project.org/*.

4.1. Experimento computacional I

Neste primeiro experimento, foram agrupados os conjuntos de dados do grupo A. Como a metodologia dos testes foi inspirada em Nascimento et al (2010), também foram utilizadas diferentes métricas de distância. São elas: Distância Euclidiana, Distância City-block, Cosseno e Correlação de Pearson. Sendo a_{ik} o $k-ésimo$ atributo do objeto i e L o número de atributos, temos:

Distância Euclidiana

Comumente utilizada, calcula a distância entre dois objetos através de seus atributos. Sua formulação é dada por

$$d_{ij} = \sqrt{\sum_{k=1}^{L} \left(a_{ik} - a_{jk}\right)^2} \qquad (1)$$

Distância City-block

Essa métrica simplesmente retorna a diferença absoluta entre os atributos de dois objetos. A formulação dela é dada por

$$d_{ij} = \sum_{k=1}^{L} \left|a_{ik} - a_{jk}\right| \qquad (2)$$

Correlação de Pearson não centralizado ou Cosseno

Essa distância é uma correlação geométrica definida pelo ângulo entre dois objetos. A formulação é a seguinte:

$$D_{ij} = \frac{\sum_{k=1}^{L} a_{ik} a_{jk}}{\sum_{k=1}^{L} a_{ik}^2 \sum_{k=1}^{L} a_{jk}^2} \qquad (3)$$

O valor de D_{ij} varia entre [-1,1]. Assim, quando $D_{ij} = 1$, significa que ângulo entre os vetores é 0º, agora quando $D_{ij} = -1$, o ângulo é 90º. A distância considerada é $d_{ij} = 1 - \left|D_{ij}\right|$.

Correlação de Pearson

Essa métrica mede a relação entre dois objetos e retorna um valor entre [-1,1]. O valor 1 gera uma associação positiva (relação forte), enquanto -1 gera relação linear perfeita negativa. A formulação é dada por

$$r_{ij} = \frac{L\sum a_{ik}a_{jk} - \sum a_{ik}\sum a_{jk}}{\sqrt{L\sum a_{ik}^2 - \left(\sum a_{ik}\right)^2}\sqrt{L\sum a_{jk}^2 - \left(\sum a_{jk}\right)^2}} \qquad (4)$$

A distância considerada é dada por $d_{ij} = 1 - |r_{ij}|$.

Com intuito de avaliar os resultados, o VNS foi comparado com GRASP (Nascimento et al, 2010), *k*-Médias (MacQueen, 1967), *k*-Medianas (Kaufman e Rousseeuw, 1990) e PAM (Kaufman e Rousseeuw, 1990), todos utilizando as mesmas métricas de distância.

O VNS gerou partições para os quatro conjuntos de dados. Essas partições foram avaliadas pelo índice CRand. O algoritmo foi executado 20 vezes e dentre essas execuções, somente a solução que obteve menor função objetivo foi considerada, pois deseja-se minimizar a soma das distâncias entre os objetos do mesmo *cluster*. Foram realizados testes com VNS agrupando os conjuntos de dados em *C clusters*, onde *C* varia de 2 a 12.

As Tabelas 3, 4, 5 e 6 mostram, respectivamente os valores de *C* e CRand para Distância Euclidiana, Distância City-block, Cosseno e Correlação de Pearson, encontrados pelo VNS para cada conjunto de dados juntamente com os resultados obtidos pelos algoritmos GRASP, *k*-Médias, k-Medianas e PAM em Nascimento et al (2010). Os resultados apresentados são aqueles em que o valor de *C* obteve maior valor do CRand. A coluna *Conjunto de dados* indica o nome dos conjuntos de dados, enquanto as colunas *C* e *CRand* indicam o número de *clusters* e o CRand encontrado para *C*, respectivamente. Os melhores CRands estão escritos em negrito.

Tabela 3. Resultados do CRand para Distância Euclidiana

Conjunto de dados	VNS		GRASP		*k*-Médias		*k*-Medianas		PAM	
	C	CRand	C	CRand	C	CRand	C	CRand	C	CRand
Yeast	9	0.142	9	0.150	7	0.170	6	**0.173**	8	0.143
Breast	2	**0.877**	2	**0.877**	2	0.803	2	0.782	2	0.828
Proteínas	4	0.316	4	**0.322**	7	0.322	7	0.313	6	0.250
Íris	3	**0.756**	3	**0.756**	3	0.730	3	0.744	3	0.730

Analisando os resultados, percebemos que o VNS obteve os melhores resultados para algumas bases, principalmente com a Distância City-block e de maneira geral, obteve bons resultados. Na Tabela 3, o VNS teve 2 melhores resultados: Breast e Íris, encontrando o mesmo número de *clusters* que o número real para os dois conjuntos de dados. Na Tabela 4, o VNS obteve 3 melhores resultados, menos para o conjunto Yeast. Para a distância Cosseno, Tabela 5, o VNS é o único a conseguir o melhor resultado para Breast, encontrando um número de *clusters* superior ao número real. Na Tabela 6, o VNS também encontra apenas 1 melhor resultado (Proteínas), mas encontra o mesmo número de *clusters* que o real.

Tabela 4. Resultados do CRand para Distância City-block

Conjunto de dados	VNS		GRASP		*k*-Médias		*k*-Medianas		PAM	
	C	CRand	*C*	CRand	*C*	CRand	*C*	CRand	*C*	CRand
Yeast	9	0.153	7	0.157	7	**0.181**	6	0.167	7	0.152
Breast	2	**0.877**	2	**0.877**	2	0.770	2	0.765	2	0.807
Proteínas	3	**0.293**	5	**0.293**	8	0.223	7	0.229	3	0.192
Íris	3	**0.818**	3	**0.818**	3	0.717	3	0.717	3	0.772

Tabela 5. Resultados do CRand para Cosseno

Conjunto de dados	VNS		GRASP		*k*-Médias		*k*-Medianas		PAM	
	C	CRand	*C*	CRand	*C*	CRand	*C*	CRand	*C*	CRand
Yeast	6	0.118	9	0.135	9	0.138	6	0.132	7	**0.146**
Breast	3	**0.437**	3	0.293	4	0.258	3	0.306	3	0.332
Proteínas	5	0.162	4	**0.349**	7	0.320	6	0.304	6	0.247
Íris	3	0.755	3	**0.941**	3	0.904	3	**0.941**	3	0.904

Tabela 6. Resultados do CRand para Correlação de Pearson

Conjunto de dados	VNS		GRASP		*k*-Médias		*k*-Medianas		PAM	
	C	CRand	*C*	CRand	*C*	CRand	*C*	CRand	*C*	CRand
Yeast	9	0.129	9	0.131	8	0.135	8	0.133	7	**0.145**
Breast	3	0.289	3	0.284	2	**0.441**	2	0.368	2	0.289
Proteínas	4	**0.344**	4	**0.344**	7	0.313	7	0.306	6	0.245
Íris	3	0.886	3	0.886	3	0.886	3	**0.941**	3	0.886

Como se pode ver, o VNS tem desempenho semelhante aos melhores algoritmos, chegando a ser o único a apresentar o melhor resultado em um dos casos. Comparando de forma geral, o GRASP apresenta os melhores resultados, seguido do VNS, o que mostra que o VNS é um algoritmo promissor na solução de problemas de agrupamentos.

4.2. Experimento computacional II

Neste segundo experimento, os conjuntos de dados utilizados foram os do grupo B. A métrica de distância, neste caso, foi a Distância Euclidiana, correspondente à formulação (1) mostrada na subseção anterior.

O VNS gerou partições para cada conjunto de dados. Para cada conjunto de dados, o algoritmo foi executado 20 vezes, gerando 20 soluções. Essas soluções foram avaliadas através do índice Rand. Para todos os conjuntos de dados foram selecionados o Rand máximo, mínimo e o médio, entre as 20 soluções geradas. Como o Rand é sensível ao número de *clusters* das duas partições utilizadas no seu cálculo, o número de *clusters* utilizado no VNS para cada conjunto de dados, corresponde ao número real.

Para verificar o desempenho do VNS, comparamos seus resultados com os de Chang et al (2009). No trabalho citado foram utilizados quatro algoritmos: *k*-Médias (Macqueen, 1967), GA-clustering (Murthy e Chowdhury, 1996), KGA-clustering (Bandyopadhyay e Maulik, 2002) e GAGR-clustering, sendo este último proposto pelos autores. Os conjuntos de dados utilizados pelos autores foram os mesmos do grupo B.

A Tabela 7 mostra os resultados dos máximos, mínimos e médios valores de Rand encontrados. Os melhores resultados estão em negrito.

Tabela 7. Resultados do máximo, médio e mínimo Rand para cada conjunto de dados

Dados	*k*-Médias	GA-clustering	KGA-clustering	GAGR-clustering	VNS
Íris					
Max	0.8623	0.8622	0.8737	0.8737	**0.8922**
Médio	0.8292	0.8518	0.8588	0.8711	**0.8922**
Min	0.7716	0.8564	0.8464	0.8623	**0.8922**
Wine					
Max	0.9265	0.9415	0.9415	0.9415	0.7191
Médio	0.8413	0.9368	0.9319	**0.9388**	0.7191
Min	0.7339	0.9246	0.9120	**0.9349**	0.7191
Glass					
Max	0.5107	0.5308	0.5367	0.5367	**0.7417**
Médio	0.3807	0.3878	0.3966	0.4047	**0.7394**
Min	0.2556	0.2759	0.2683	0.2736	**0.7253**
Balance					
Max	0.5826	0.5827	0.5834	0.6075	**0.6308**
Médio	0.5603	0.5642	0.5642	0.5646	**0.5887**
Min	0.5215	0.5345	0.5296	0.5320	**0.5581**
Liverdisorder					
Max	0.5004	0.5014	0.5043	**0.5050**	0.5016
Médio	0.4989	0.5004	0.5022	**0.5031**	0.5016
Min	0.4984	0.4993	0.5021	0.5021	0.5016

Analisando a Tabela 7, verificamos que o VNS tem desempenho superior aos outros algoritmos, pois consegue os melhores Rands em 3 dos 5 conjuntos de dados.

5. Considerações finais

Este trabalho propôs uma nova abordagem heurística aos problemas de agrupamentos por meio do algoritmo VNS. Utilizando uma abordagem para particionamento em cliques, a meta-heurística VNS gerou partições para 8 conjuntos de dados. Essas partições foram avaliadas por dois índices de validação externa: Corrected Rand (CRand) e Rand . Os resultados obtidos foram comparados com dois trabalhos publicados recentemente.

A análise dos resultados pode comprovar que o VNS tem desempenho competitivo aos algoritmos utilizados na comparação. No primeiro experimento, ele obteve resultados próximos ou iguais aos melhores algoritmos e em um caso foi o melhor algoritmo. Já no segundo experimento, o VNS obteve melhor desempenho, conseguindo os melhores resultados para três dos cinco conjuntos de dados.

Diante desses resultados, o VNS pode ser considerado um algoritmo com potencial aplicação a problemas de agrupamentos.

Como trabalho futuro, pretendemos usar a meta-heurística híbrida Busca por Agrupamentos (CS, do inglês *Clustering Search*) (Chaves, 2009) na solução de problemas de agrupamentos. Esta meta-heurística é caracterizada por realizar buscas apenas em regiões promissoras à melhoria da solução. O intuito do CS é melhorar o processo de convergência e diminuir o esforço computacional. Acreditamos que o CS seja um algoritmo promissor na solução de problemas de agrupamentos.

Referências

Al-Sultan, K.S. (1995), A Tabu search approach to the clustering problem. *Pattern Recognition*, 28, 1443-1451.
Amorim, S.G.; Barthélemy, J.P.; Ribeiro, C.C.(1992), Clustering and clique partitioning: Simulated annealing and tabu search approaches. *Journal of Classification*, 9, 17-41.
Asuncion, A. e Newman, D. J. (2007), UCI Machine Learning Repository. http://www.ics.uci.edu/~mlearn/MLRepository.html. Irvine, CA: University of California, School of Information and Computer Science.
Bandyopadhyay, S. e Maulik, U.(2002), An evolutionary technique based on K-Means algorithm for optimal clustering in R^N, *Inf. Sci. Appl.*, 146, 221-237.
Barbara, D.(2000), An introduction to cluster analysis for data mining. http://www-users.cs.umn.edu/~han/dmclass/cluster_survey_10_02_00.pdf [acessado em 26 de abril de 2010].
Chang, Dong-Xia; Zhang, Xian-Da e Zheng, Chang-Wen. (2009), A genetic algorithm with gene rearrangement for K-means clustering, *Pattern Recognition*, 42, 1210-1222.
Feo, T. e Resende, M. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization.*6, 109–133.
Hubert, L. e Arabie, P.(1985). Comparing partitions. *Journal of Classification*, 2, 193-218.
Jain, A.K. e Dubes, R.C. , *Algorithms for Clustering*, Prentice-Hall, 1988.
Jain, A.K.; Murty, M.N.; Flynn, P.J.(1999), Data clustering: A review. *ACM Computing Surveys* 31(3), 264-323.
Kaufman, L.; Rousseeuw, P.J., *Finding groups in data: an introduction to cluster analysis*. Wiley, New York, 1990.
Kochenberger, G.; Glover, Fred; Alidaee, B.; Wang, Haibo. (2005), Clustering of Microarray data via Clique Partitioning. *Journal of Combinatorial Optimization*, 10, 77-92.
MacQueen, J. B. (1967) Some methods for classification and analysis of multivariate observations. P*roceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley, 281 – 297.
Mehrotra, A.;Trick, M.(1998), Cliques and clustering: A combinatorial approach. *Operations Research Letters*, 22, 1-12.
Mladenovic, N. e Hansen, P.(1997), Variable neighborhood search. *Computers and Operations Research*, 24, 1097-1100.
Murthy, C.A.; Chowdhury, N.(1996), In search of optimal clusters using genetic algorithms. *Pattern Recognition Letters*, 17, 825-832.
Nascimento, M.C.V; Toledo, F.M.B. e Carvalho, A.C.P.L.F.(2010), Investigation of a new GRASP-based clustering algorithm applied to biological data. *Computers and Operations Research*, 37 (8), 1381-1388.
Rand, W.M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association.*60, 846 – 850.
Xu, R. e Wunsch, D. (2005), Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3), 645-678.
Chaves, A. A. (2009). Uma meta-heurística híbrida com Busca por Agrupamentos aplicada a problemas de otimização combinatória. Tese de Doutorado em Computação Aplicada - Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, SP.