UMA META-HEURÍSTICA DE BUSCA EM VIZINHANÇA VARIADA PARA O PROBLEMA DE *CLUSTERING* HETEROGÊNEO

Éverton Santi
Escola de Ciências e Tecnologia
Universidade Federal do Rio Grande do Norte, Natal-RN 59072-970, Brasil
esanti@ect.ufrn.br

Daniel Aloise
Departamento de Engenharia de Computação e Automação
Universidade Federal do Rio Grande do Norte, Natal-RN 59072-970, Brasil
aloise@dca.ufrn.br

Simon J. Blanchard
McDonough School of Business
Georgetown University, Washington, DC 20057, USA
sjb247@georgetown.edu

RESUMO

Apresenta-se neste trabalho um algoritmo baseado na meta-heurística de Busca em Vizinhança Variada para a solução do Problema de *Clustering* Heterogêneo. Este novo e relativamente inexplorado problema de otimização combinatória permite que múltiplas matrizes de dissimilaridades sejam utilizadas para o particionamento de objetos. Desta forma, pode-se identificar múltiplas estruturas de categorias em relação a estes objetos, representando-se de forma apropriada a heterogeneidade presente nos dados. Esta propriedade pode ser eliminada pelo processo de agregação utilizado em algoritmos tradicionais. Os resultados mostrados neste artigo sugerem que o algoritmo proposto é eficiente, mesmo quando aplicado a instâncias de grande porte. Adicionalmente, verifica-se que este algoritmo é capaz de recuperar com precisão a informação contida nos dados. Um exemplo de aplicação é apresentado ao final deste trabalho para dados coletados a partir de clientes de uma loja de conveniências localizada nos Estados Unidos.

PALAVRAS CHAVE. ***Clustering*. Heterogeneidade. Meta-heurísticas.**

Otimização Combinatória. Meta-Heurísticas. Outras Aplicações em P.O.

ABSTRACT

We present in this work a Variable Neighborhood Search algorithm for solving the Heterogeneous Clustering Problem. It is a new and relatively unexplored combinatorial optimization problem which allows us to use multiple dissimilarities matrices when clustering objects. Thus, is possible to identify multiple category structures related to the objects and to represent properly the heterogeneous nature of data. This property might be eliminated during the aggregation step in traditional clustering. The results showed in this paper suggest that the proposed algorithm is efficient, even when applied to large instances. In addition, this algorithm recovers precisely the information contained in data. An illustrative application of the proposed algorithm is also presented, considering a convenience store from the United States.

KEYWORDS. Clustering. Heterogeneity. Metaheuristics.

Combinatorial Optimization. Metaheuristics. Other Applications in O.R.

1. Introdução

Algoritmos de *clustering* podem ser utilizados na determinação de grupos (*clusters*) de objetos. Neste contexto, objetos mais similares entre si devem ser designados a um mesmo grupo, enquanto que objetos dissimilares devem ser dispostos em *clusters* distintos [Hansen e Jaumard, 1997]. *Clustering* é onipresente, com aplicações em diversas áreas como Ciências Naturais, Psicologia, Medicina, Engenharia, Economia e *Marketing*, entre outros [e.g. Frey e Dueck, 2007; Jain et al., 1999; McLachlan e Basford, 1988].

Geralmente, os dados de entrada para um algoritmo de *clustering* clássico consistem em uma única matriz X com dimensão $n \times s$, obtida pela medição de s características relativas a n objetos. Esta matriz é utilizada para se calcular uma segunda matriz $D = (d_{ij})$ com dimensão $n \times n$, na qual cada elemento d_{ij}, para todo $i, j \in \{1, \ldots, n\}$ representa o grau de dissimilaridade entre cada par de objetos i e j.

De forma a exemplificar esta representação por uma única matriz de dissimilaridades, considera-se o conjunto de dados Iris [Fisher, 1936], um dos mais populares da literatura. Este conjunto consiste em 150 amostras oriundas de três espécies das flores Iris. Cada uma destas amostras foi mensurada em relação a quatro características. A partir destas características as flores são, geralmente, agrupadas em três espécies (*clusters*) esperadas. Para esta tarefa, algoritmos de *clustering* clássicos, como por exemplo k-*means*, *single-linkage* ou *complete-linkage* fornecem, na maioria das vezes, bons resultados.

No entanto, é possível que mais de uma matriz de dissimilaridades estejam disponíveis para problemas como este. No contexto do conjunto Iris, cada elemento da amostra poderia ser mensurado por diferentes especialistas em relação aos atributos considerados, ação que por sua vez poderia levar à medições com diferenças significativas (ou mesmo erros). Logo, ao se agregar as matrizes de dissimilaridades para se aplicar um algoritmo de *clustering* clássico, pode-se mascarar a real informação contida nos dados.

Em Ciências Sociais, por exemplo, é comum que a opinião de diversos indivíduos seja utilizada para a obtenção de medidas de similaridades par-a-par entre marcas de produtos. Uma vez que diversos indivíduos julguem o quão similar são entre si os itens de um conjunto, uma matriz de dissimilaridades será obtida para cada um destes indivíduos. Abordagens como esta são vistas em estudos sobre sobre identificação de preferências [Carpenter e Nakamoto, 1994], composição de anúncios [Schweidel et al., 2006], comparação entre marcas [Bijmolt et al., 1998], definição de arranjos de lojas [Arora, 1982] e decisões acerca de substituições [Ratneshwar e Shocker, 1991; Hamilton et al., 2014].

Considerando-se a limitação sugerida em relação aos algoritmos de *clustering* tradicionais, Santi et al. [2015] propuseram o Problema de *Clustering* Heterogêneo (PCH). Este problema consiste em identificar G grupos de indivíduos cujas matrizes de dissimilaridades sugerem uma solução similar em relação ao particionamento do conjunto de objetos considerados. Logo, para cada grupo de indivíduos identificado pelo PCH, obtém-se uma estrutura de *clusters* em relação aos n objetos. O modelo dos autores é baseado no conceito de medianas. Isto é, cada objeto é associado ao item mais representativo em seu *cluster*.

Apesar de englobar o aspecto da heterogeneidade dos dados em um problema de *clustering*, o PCH consiste em um problema de programação não-linear inteira mista, sendo de complexa resolução mesmo para instâncias de pequeno porte, como demonstrado pelos autores. Diante desta limitação, Santi et al. [2015] buscaram formulações alternativas para o problema de forma a viabilizar sua resolução via algoritmos exatos. Para tal, apresentaram a formulação PCH1, a qual é baseada no método do grande M, e a formulação PCH2, baseada no processo de convexificação por meio das desigualdades de Fortet [Fortet, 1960].

Para as formulações PCH1 e PCH2, Santi et al. [2015] demonstraram, a partir de experimentos computacionais com o resolvedor CPLEX e dados simulados, que é viável a obtenção limites inferiores e soluções subótimas apenas para algumas das instâncias consideradas. Os au-

tores limitaram o tempo de processamento a 24h. Na ocasião, um computador com 64GB de memória e 6 processadores (12 núcleos) foi utilizado. Para estas configurações, em apenas um dos 27 casos considerados se observou a solução ótima do problema. Desta forma, considerando-se as limitações supracitadas, propõe-se neste trabalho um algoritmo baseado na meta-heurística de Busca em Vizinhança Variada (*Variable Neighborhood Search - VNS*) [Mladenović e Hansen, 1997] para a resolução do PCH. Objetiva-se a partir da implementação desta meta-heurística não só resolver o problema de forma eficiente, como também viabilizar sua aplicação a casos reais.

De forma a apresentar e validar o algoritmo proposto, além de demonstrar sua aplicabilidade, o restante deste texto está organizado como segue: a Seção 2 revisa a formulação original do PCH, apresentada por Santi et al. [2015]. As formulações PCH1 e PCH2 não serão revisadas neste trabalho, dadas as limitações impostas em relação ao tamanho do texto. A Seção 3 descreve o algoritmo *VNS* proposto. A Seção 4 apresenta os resultados computacionais obtidos a partir da meta-heurística, comparando-os aos resultados reportados em [Santi et al., 2015]. Nesta seção, ainda, avalia-se a capacidade de recuperação da informação deste algoritmo por meio de uma simulação de Monte Carlo. A Seção 5 apresenta um exemplo de aplicação do algoritmo proposto ao problema de segmentação de consumidores. Por fim, a Seção 6 apresenta algumas considerações sobre este trabalho, bem como direções futuras para esta pesquisa.

2. O Problema de *Clustering* Heterogêneo

De forma a definir o PCH [Santi et al., 2015], considera-se que m indivíduos devem julgar n objetos, obtendo-se uma matriz $D^k = (d^k_{ij})$ para todo o indivíduo $k \in \{1, \ldots, m\}$. d^k_{ij} representa o grau de dissimilaridade entre os objetos i e j de acordo com a opinião do indivíduo k, para todo $i, j \in \{1, \ldots, n\}$ e para todo $k \in \{1, \ldots, m\}$. Deve-se também obter um valor para c^k, para todo $k \in \{1, \ldots, m\}$, o qual representa o número de *clusters* esperados pelo indivíduo k em relação aos n objetos. As matrizes D^k e os escalares c^k são parâmetros do PCH, assim como o número de grupos G a ser considerado na segmentação dos indivíduos. O método utilizado para a obtenção destes valores fica a critério do usuário do modelo.

O conjunto de variáveis de decisão do PCH inclui as variáveis binárias z, as quais representam a designação dos indivíduos aos diferentes grupos. Desta forma, tem-se que $z^{kg} = 1$ caso o indivíduo k seja designado ao grupo g, e $z^{kg} = 0$ caso contrário, para todo $k \in \{1, \ldots, m\}$ e para todo $g \in \{1, \ldots, G\}$. As variáveis e, também binárias, representam a associação entre objetos e medianas, condicionada à pertinência dos indivíduos nos G grupos. Logo, $e^g_{ij} = 1$ caso o objeto i esteja associado à mediana j dentro do grupo g, e $e^g_{ij} = 0$ caso contrário, para todo $g \in \{1, \ldots, G\}$ e para todo $i, j \in \{1, \ldots, n\}$. Neste texto, o termo "grupo" é sempre utilizado como referência à segmentação dos indivíduos em *clusters*, isto é, solução armazenada nas variáveis z. O termo "estrutura de categorias", por sua vez, é sempre utilizado em referência ao particionamento dos objetos. Logo, refere-se à solução armazenada nas variáveis e.

Dados os parâmetros e variáveis de decisão descritos, formula-se o PCH como:

$$\min \sum_{k=1}^{m} \sum_{g=1}^{G} z^{kg} \left[\sum_{i=1}^{n} \sum_{j=1}^{n} d^k_{ij} e^g_{ij} \right] \tag{1}$$

sujeito a

$$\sum_{j=1}^{n} e^g_{ij} = 1, \;\; \forall g \in \{1, \ldots, G\}, \; \forall i \in \{1, \ldots, n\}, \tag{2}$$

$$e^g_{ij} \leq e^g_{jj}, \;\; \forall g \in \{1, \ldots, G\}, \; \forall i, j \in \{1, \ldots, n\}, \tag{3}$$

$$\sum_{g=1}^{G} z^{kg} = 1, \;\; \forall k \in \{1, \ldots, m\}, \tag{4}$$

$$\sum_{k=1}^{m} z^{kg} \geq 1, \ \ \forall g \in \{1, \ldots, G\}, \tag{5}$$

$$\sum_{j=1}^{n} e_{jj}^{g} = \left\lfloor \frac{\sum_{k=1}^{m} c^k z^{kg}}{\sum_{k=1}^{m} z^{kg}} \right\rfloor, \ \ \forall g \in \{1, \ldots, G\}, \tag{6}$$

$$e_{ij}^{g} \in \{0, 1\}, \ \ \forall g \in \{1, \ldots, G\}, \ \forall i, j \in \{1, \ldots, n\}, \tag{7}$$

$$z^{kg} \in \{0, 1\}, \ \ \forall g \in \{1, \ldots, G\}, \ \forall k \in \{1, \ldots, m\}, \tag{8}$$

em que (1) minimiza a soma das dissimilaridades para cada associação de objetos i e sua respectiva mediana j, condicionada à pertinência de cada indivíduo k em relação aos G grupos. O conjunto de restrições definido em (2) impõe que cada objeto i seja associado a exatamente uma mediana j dentro de um grupo g. As restrições em (3) garantem que um objeto i somente será associado ao outro objeto j em um grupo g se o objeto j for uma mediana dentro daquele grupo. As restrições em (4) impõem que um indivíduo k deve ser designado a exatamente um grupo g, enquanto que as restrições em (5) garantem que não haverá grupos vazios.

As restrições em (6) impõem que o número total de medianas (*clusters* ou categorias) para cada grupo g deverá ser igual ao piso do número médio de medianas esperado pelos indivíduos alocados a este grupo. Por fim, as restrições em (7) e (8) são restrições de integralidade sobre os valores das variáveis de decisão do problema.

3. Algoritmo Proposto

O algoritmo proposto neste trabalho para resolver o problema (1–8) é composto por uma heurística construtiva, um procedimento de perturbação (*shaking*) e a busca local, elementos básicos de qualquer algoritmo baseado na meta-heurística *VNS*. Dado que o *VNS* é uma abordagem popular e consolidada na literatura, a mesma não será revisada neste texto. Maiores detalhes sobre este método podem ser encontrados em [Mladenović et al., 2007; Mladenović e Hansen, 1997]. As subseções a seguir detalham os componentes do algoritmo proposto.

3.1. Heurística Construtiva

A fim de se obter uma solução inicial para o PCH, resolve-se primeiramente o problema de se alocar m indivíduos a G grupos, considerando-se para tal o problema das p-medianas (com $p = G$). Utiliza-se neste processo uma matriz de distâncias $\mathcal{F} = (f^{ab})$ com dimensão $m \times m$, na qual cada elemento f^{ab}, para todo $a, b \in \{1, \ldots, m\}$, representa a norma de Frobenius para $D^a - D^b$. A partição obtida por meio das p-medianas representa a designação de grupos dos indivíduos, isto é, obtém-se os valores para as variáveis z. A partir dos grupos obtidos, o subproblema $M_g(z)$ é resolvido para todo $g = \{1, \ldots, G\}$ de modo a se obter uma solução inicial completa para o PCH. O problema $M_g(z)$ é definido como:

$$M_g(z) = \min \sum_{i=1}^{n} \sum_{j=1}^{n} \overline{d}_{ij}^{g} e_{ij}^{g} \tag{9}$$

sujeito a

$$\sum_{j=1}^{n} e_{ij}^{g} = 1, \ \ \forall j \in \{1, \ldots, n\}, \tag{10}$$

$$e_{ij}^{g} \leq e_{jj}^{g}, \ \ \forall i, j \in \{1, \ldots, n\}, \tag{11}$$

$$\sum_{j=1}^{n} e_{jj}^{g} = \lfloor \Omega^g \rfloor, \tag{12}$$

$$e_{ij}^g \in \{0,1\}, \ \ \forall i,j \in \{1,\ldots,n\}, \tag{13}$$

em que $\overline{d}_{ij}^g = \sum_{k=1}^m d_{ij}^k z^{kg}$ e $\Omega^g = \frac{\sum_{k=1}^m c^k z^{kg}}{\sum_{k=1}^m z^{kg}}$. O problema descrito em (9–13) corresponde também ao problema das p-medianas, mais especificamente à abordagem descrita em [Hansen e Mladenović, 1997], a qual garante que cada *cluster* contenha, no mínimo, um objeto. Como este problema é clássico na literatura, o modelo (9–13) não será descrito em detalhes. Nesta pesquisa, os problemas de otimização combinatória relacionados à heurística construtiva e demais elementos do *VNS*, como as estruturas de vizinhança da busca local, são resolvidos por meio do *solver* CPLEX de forma exata. Dependendo do tamanho das instâncias a serem utilizadas em certas aplicações, pode-se considerar métodos aproximados nestas etapas.

3.2. Perturbação (*Shaking*)

Considera-se que os indivíduos serão redistribuídos entre os G grupos de forma aleatória. O número de indivíduos a serem movidos entre os grupos a cada iteração do *VNS* é alterado em razão da melhora ou não do custo da solução obtida na iteração anterior. Desta forma, tem-se um nível de perturbação maior ou menor, dependendo-se da necessidade de se explorar vizinhanças mais distantes em relação à solução atual. Este método de perturbação, como descrito, altera os valores das variáveis z. Portanto, a primeira etapa da busca local irá resolver o problema $M_g(z)$ para cada um dos grupos alterados pela perturbação a fim de recuperar uma nova solução completa para o problema.

3.3. Busca Local

A busca local do algoritmo proposto é baseada na abordagem de Descida em Vizinhança Variada, mais conhecida do termo em Inglês *Variable Neighborhood Descent* (*VND*). Para tal, três estruturas de vizinhança são definidas: N_1, N_2 e N_3. A busca local, ou *VND*, será finalizada sempre que estas três vizinhanças forem exploradas sem que haja melhora no custo da função objetivo. Caso a exploração de uma estrutura de vizinhança leve à obtenção de uma solução de melhor custo, reinicia-se todo o processo de busca local. As próximas seções deste texto detalham cada uma destas três estruturas.

3.3.1. Estrutura de Vizinhança N_1

A vizinhança N_1 resolve o subproblema (9–13) para cada um dos grupos modificados pelo processo de perturbação. Para cada grupo, esta estrutura de vizinhança identifica a partição ótima dos objetos, considerando o número de medianas como conhecido, condicional à segmentação dos indivíduos em grupos.

3.3.2. Estrutura de Vizinhança N_2

Para a vizinhança N_2, assume-se temporariamente que a solução do PCH é conhecida em relação ao agrupamento dos objetos (valores das variáveis de decisão e). Logo, busca-se otimizar a função objetivo pela realocação ótima dos indivíduos entre os grupos. Isto corresponde a encontrar os valores das variáveis z pela resolução do seguinte problema de otimização combinatória:

$$W(e) = \min \sum_{k=1}^{m} \sum_{g=1}^{G} z^{kg} \tilde{d}^{kg} \tag{14}$$

sujeito a

$$\frac{\sum_{k=1}^m c^k z^{kg}}{\sum_{k=1}^m z^{kg}} \geq \omega^g, \ \ \forall\, g \in \{1,\ldots,G\}, \tag{15}$$

$$\sum_{g=1}^{G} z^{kg} = 1, \ \ \forall\, k \in \{1,\ldots,m\}, \tag{16}$$

$$z^{kg} \in \{0,1\}, \ \forall\, g \in \{1,\ldots,G\}, \ k \in \{1,\ldots,n\}, \tag{17}$$

em que $\tilde{d}^{kg} = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}^{k} e_{ij}^{g}$ e $\omega^{g} = \sum_{j=1}^{n} e_{jj}^{g}$. O problema (14–17) é, na maioria das vezes, resolvido no nó raiz do algoritmo de *branch-and-bound* implementado pelo CPLEX, como observado em experimentos computacionais realizados. Desta forma, o resolvedor foi configurado para resolver apenas este nó. Caso o problema $W(e)$ não seja resolvido até a otimalidade, considerando-se esta configuração, a solução dada é descartada e a solução prévia é mantida.

3.3.3. Estrutura Vizinhança N_3

É trivial que o custo da função objetivo em problema de *clustering* seja reduzido quando se permite que um maior número de *clusters* seja utilizado. No entanto, no PCH, o número de medianas (categorias) em cada grupo g, para todo $g \in \{1,\ldots,G\}$, é limitado pelas restrições definidas em (6). Portanto, aumentar o número de medianas para um dado grupo $g*$, não necessariamente reduz custo da função objetivo. O problema como um todo pode se tornar inviável a partir desta alteração. Desta forma, a vizinhança N_3 busca verificar se o número de medianas em um determinado grupo g^* pode ser aumentado, alocando-se objetos a uma nova mediana, sem que as restrições em (6) sejam violadas. O Algoritmo 1 detalha como esta vizinhança funciona.

Algoritmo 1 Vizinhança N_3

Entrada: uma solução (z, e) para o PCH
$(z^{opt}, e^{opt}) \leftarrow (z, e)$
para $g^* = 1,\ldots,G$ **faça**
  Resolva $M_{g^*}(z)$ com Ω^{g^*} substituído por $\Omega^{g^*} + 1$;
  Resolva $W(e)$;
  se $W(e)$ é inviável **ou** o custo de (z, e) é maior que o custo de (z^{opt}, e^{opt}) **então**
    $(z, e) \leftarrow (z^{opt}, e^{opt})$;
  senão
    $(z^{opt}, e^{opt}) \leftarrow (z, e)$;
  fim se
fim para
retorna (z^{opt}, e^{opt})

Dado que a busca local proposta considera explorar três estruturas de vizinhança, fez-se necessário definir a ordem em que estas seriam aplicadas. Após análises e experimentos, optou-se por definir N_3 como a última vizinhança a ser explorada, dada que é computacionalmente mais dispendiosa em relação às demais. Adicionalmente, dado que o processo de perturbação (*shaking*) altera a solução corrente apenas em relação às variáveis z, aplicar N_2 logo após este processo faria com que seu efeito fosse revertido. Logo, N_1 foi definida como a primeira e N_2 como a segunda vizinhança a ser explorada.

4. Resultados Computacionais

Nesta seção, analisam-se os resultados obtidos por meio do algoritmo proposto em relação a dois aspectos. Primeiramente, compara-se os custos da função objetivo dados pelo *VNS* aos custos reportados por Santi et al. [2015] para as formulações PCH1 e PCH2. Avalia-se, em um segundo momento, a capacidade de recuperação da informação deste algoritmo. Para tal, os experimentos computacionais aqui reportados foram realizados a partir das mesmas instâncias utilizadas por Santi et al. [2015], geradas com base em uma simulação de Monte Carlo descrita em [Blanchard et al., 2012].

Para estas instâncias, a estrutura de grupos e categorias, informação a ser recuperada a partir dos dados, é conhecida (valores das variáveis z e e). Dentre os fatores considerados na composição das instâncias estão o número total de indivíduos ($m \in \{150, 300, 450\}$), o número

total de grupos ($G \in \{2, 6, 10\}$), o número total de objetos ($n \in \{18, 20, 30\}$), o número total de categorias (medianas) definidos para os grupos, erros aleatórios introduzidos nas matrizes de distâncias (utilizando-se $N(0, 0.5)$ ou $N(0, 0.1)$) e erros aleatórios introduzidos no número de medianas definido por cada indivíduo (utilizando-se $N(0, 0.5)$ ou $N(0, 0.1)$). Maiores detalhes sobre as características de cada uma das 27 instâncias são apresentados na Tabela 2.

O computador utilizado nos experimentos possui 6 processadores Intel(R) Xeon(R) CPU X5650 (12 núcleos de processamento) de 2.67GHz e 62GB de memória RAM. As formulações PCH1 e PCH2 foram resolvidas com o resolvedor CPLEX 12.5 (configuração padrão) com um tempo limite de 24 horas para cada instância. Para o algoritmo proposto, cada instância foi executada 10 vezes, considerando-se 10 minutos como tempo limite de processamento. Para o *VNS*, dado que suas estruturas de vizinhança e heurística construtiva fazem uso do CPLEX, este foi configurado para utilizar um único processador.

4.1. Performance e Estabilidade do Algoritmo Proposto

Os resultados computacionais para as formulações PCH1 e PCH2, bem como para o algoritmo proposto estão dispostos na Tabela 1. Nesta tabela, a primeira coluna indica o código de cada instância. A segunda coluna corresponde aos melhores limites inferiores obtidos a partir resolução da formulação PCH2 pelo CPLEX. A terceira coluna representa o melhores limites superiores para esta mesma formulação. A quarta coluna mostra o GAP calculado entre estes dois valores. A quinta coluna reporta os melhores limites superiores obtidos por meio da resolução da formulação PCH1, resolvida via CPLEX, enquanto que a sexta coluna apresenta GAP calculado entre estes valores e os limites inferiores obtidos para a PCH2.

As demais colunas nesta tabela apresentam resultados observados a partir das 10 execuções do *VNS* para cada instância. Tem-se, respectivamente: custo da solução dada pela heurística construtiva, custo médio dos limites superiores obtidos, desvio padrão para os limites superiores, percentual de melhora relativo à diferença entre a média dos limites superiores e o custo dado pela heurística construtiva. Os valores de GAP para o *VNS* foram calculados com base na média dos limites superiores dados pelo algoritmo e os limites inferiores dados pela formulação PCH2.

Utilizaram-se estes valores como referência dado que para a formulação PCH1 o CPLEX não foi capaz de obter quaisquer resultados em relação a limites inferiores. Destaca-se que limites superiores também não foram reportados para a Instância 1 em relação à formulação PCH2. Como sugerem Santi et al. [2015], esta formulação possui elevado número de variáveis de decisão, o que dificulta sua resolução por algoritmo exatos. Para esta instância e formulação, tem-se aproximadamente 4 bilhões de variáveis de decisão.

Comparando-se os custos médios da função objetivo dados pelo algoritmo proposto em relação aos custos obtidos via CPLEX para a formulação PCH1, observa-se que o *VNS* é superior em 24 dos 27 casos. Custos idênticos são observados para as Instâncias 7 e 26, sendo o *VNS* inferior em apenas um caso, relativo à Instância 8. Quando comparado à formulação PCH2, observa-se que o *VNS* é superior em 25 casos. Para as Instâncias 7 e 26, novamente, os resultados são idênticos. Obteve-se, ainda, a solução ótima do problema para três instâncias: Instância 12, Instância 20 e Instância 26. Os resultados indicam que o algoritmo proposto é eficiente na resolução do PCH, dado o reduzido esforço computacional necessário à sua aplicação quando comparado ao CPLEX.

Sugere-se, adicionalmente, que este algoritmo é estável, dado que os valores de desvio padrão mostrados na Tabela 1 indicam variações pequenas no custo das soluções. Este indicador é afetado em maiores proporções apenas em alguns dos casos reportados, os quais consistem nas instâncias de maior complexidade de resolução. Por exemplo, para a Instância 5, uma das maiores, o número de indivíduos e grupos é elevado, bem como erros aleatórios foram introduzidos tanto no número de medianas quanto nas matrizes de dissimilares.

Por fim, ao se comparar o custo das soluções dadas pela heurística construtiva ao *VNS* como um todo, destaca-se que a heurística construtiva fornece soluções cujo custo não pode ser melhorado pelo algoritmo em 9 dos 27 casos (33% aprox.). No entanto, constata-se que as soluções

dadas por esta heurística apresentam custo melhor do que as soluções dadas pelo CPLEX para a maioria das instâncias. Isto é observado em 15 dos 27 casos, considerando-se a formulação PCH1, e em 23 dos 27 casos para a formulação PCH2.

Tabela 1: Resultados computacionais para as formulações PCH1 e PCH2 e algoritmo *VNS* proposto

Instância	PCH2			PCH1		VNS				
	Melhor Lim. Inf.	Melhor Lim. Sup.	GAP	Melhor Lim. Sup.	GAP	HC	Lim. Sup. Médio	Desv. Padrão	Melhora BL	GAP
1	2705.93	-	-	3507.73	22.86%	3257.67	3211.30	6.34	1.42%	15.74%
2	2336.58	2389.32	2.21%	2391.95	2.31%	2388.25	2388.25	0.00	0.00%	2.16%
3	4398.94	4607.07	4.52%	5104.79	13.83%	4604.21	4604.21	0.00	0.00%	4.46%
4	1823.61	1870.07	2.48%	1871.15	2.54%	1996.80	1869.92	0.00	6.35%	2.48%
5	3355.22	7353.26	54.37%	4900.88	31.54%	3936.21	3792.57	42.55	3.65%	11.53%
6	1382.72	2396.45	42.30%	1762.6	21.55%	1525.20	1525.20	0.00	0.00%	9.34%
7	2413.33	2651.01	8.97%	2651.01	8.97%	2900.01	2651.01	0.00	8.59%	8.97%
8	1470.84	1824.17	19.37%	1585.17	7.21%	1785.67	1598.42	27.76	10.49%	7.98%
9	6654	8425.28	21.02%	7784.45	14.52%	7604.90	7391.29	17.86	2.81%	9.98%
10	4995.3	7267.4	31.26%	5889.93	15.19%	6049.09	5701.86	29.07	5.74%	12.39%
11	2500.14	4235.07	40.97%	2958.48	15.49%	2835.41	2835.41	0.00	0.00%	11.82%
12	3749.99	4850	22.68%	3779.99	0.79%	3749.99	3749.99	0.00	0.00%	0.00%
13	3064.39	7330.24	58.20%	4678.68	34.50%	3562.49	3562.49	0.00	0.00%	13.98%
14	2905.84	4648.16	37.48%	3227.33	9.96%	3250.00	3075.15	4.99	5.38%	5.51%
15	5644.12	6087.38	7.28%	5775.77	2.28%	6000.37	5760.64	0.00	4.00%	2.02%
16	9670.02	10752.5	10.07%	10204.2	5.23%	10192.50	9869.40	0.00	3.17%	2.02%
17	2671.69	4127.72	35.27%	3386.65	21.11%	3441.24	3136.46	22.44	8.86%	14.82%
18	5958.8	8448.21	29.47%	7073.91	15.76%	6497.31	6497.31	0.00	0.00%	8.29%
19	1190.01	1269.01	6.23%	1264.67	5.90%	1241.67	1206.01	0.00	2.87%	1.33%
20	10935	12645	13.52%	11435.7	4.38%	10935.00	10935.00	0.00	0.00%	0.00%
21	3405.44	3726.86	8.62%	3639.28	6.43%	3709.92	3593.04	0.00	3.15%	5.22%
22	4233.5	4710.93	10.13%	4906.25	13.71%	4929.88	4610.67	0.00	6.48%	8.18%
23	5312.94	7262.72	26.85%	5784.2	8.15%	5987.09	5685.85	32.20	5.03%	6.56%
24	3212.93	4834.31	33.54%	3983.77	19.35%	3945.64	3764.60	12.40	4.59%	14.65%
25	1141.32	1460.81	21.87%	1363.56	16.30%	1389.30	1253.22	13.17	9.80%	8.93%
26	1874.99	1874.99	0.00%	1874.99	0.00%	1874.99	1874.99	0.00	0.00%	0.00%
27	8657.6	12690	31.78%	10103.4	14.31%	9207.00	9004.64	39.75	2.20%	3.85%

4.2. Análise da Capacidade de Recuperação da Informação do Algoritmo Proposto

As 27 instâncias utilizadas neste trabalho foram geradas com base em uma simulação de Monte Carlo. Logo, o particionamento dos indivíduos em grupos, bem como as estruturas de categorias a serem recuperadas para os objetos são conhecidos. De forma a verificar o quão eficaz o algoritmo proposto é ao recuperar estas informações, calculou-se o Índice de Rand Ajustado (*Adjusted Rand Index - ARI*) [Hubert e Arabie, 1985]. Este índice indica a similaridade entre duas estruturas de *clusters*. Pode-se, portanto, comparar a estrutura de *clusters* recuperada para a variáveis z e e em relação ao resultado esperado para cada instância.

Os resultados apresentados nas duas últimas colunas da Tabela 2 listam os valores calculados para o ARI médio em relação à recuperação das estruturas de categorias (coluna $\overline{e}$) e os grupos de indivíduos (coluna $\overline{z}$). Para calcular estes valores médios, considerou-se o ARI calculado para cada um dos indivíduos separadamente.

Os valores mostrados na Tabela 2 sugerem que o algoritmo proposto é eficiente em recuperar a informação original contida nos dados, pois o ARI médio em relação aos grupos de indivíduos é 0.889. Com relação à recuperação das estruturas de categorias o ARI médio é 0.952. Para 9 das 27 instâncias o ARI médio calculado sugere uma recuperação perfeita da informação. De acordo com os dados mostrados na Tabela 2, percebe-se que o algoritmo proposto é menos eficiente em recuperar a informação para as instâncias em que o número de grupos (G) é elevado ($G = 10$), bem como para aquelas em que há um maior nível de ruído adicionado ao número de medianas.

5. Exemplo de Aplicação

De forma a ilustrar a aplicação do PCH, considerou-se a empresa *Vital Vittles*, inaugurada em 1973. Está é uma das pioneiras no segmento de loja de conveniências a atender os estudantes

Tabela 2: Simulação de Monte Carlo: Recuperação da Informação

					Perturbações		ARI	
Instância	Indivíduos	Grupos	Objetos	Medianas	Dissimilaridades	Medianas	$\overline{z}$	$\overline{e}$
1	150	10	30	50 % 3, 50 % 6	N(0, 0.1)	N(0, 0.5)	0.886	0.903
2	300	2	18	All 6	N(0, 0.1)	0	1.000	1.000
3	450	2	18	50 % 3, 50 % 6	N(0, 0.05)	0	1.000	1.000
4	150	2	18	All 3	N(0, 0.05)	N(0, 0.5)	0.947	0.988
5	450	10	18	All 6	N(0, 0.05)	N(0, 1)	0.928	0.911
6	150	10	18	50 % 3, 50 % 6	N(0, 0.05)	0	1.000	1.000
7	300	2	18	All 6	0	N(0, 0.5)	0.987	0.881
8	150	10	18	50 % 3, 50 % 6	0	N(0, 1)	0.166	0.979
9	300	10	30	All 3	N(0, 0.05)	N(0, 0.5)	0.700	0.787
10	450	6	18	All 3	N(0, 0.1)	N(0, 1)	0.771	0.844
11	150	6	30	All 6	N(0, 0.1)	0	1.000	1.000
12	300	10	18	All 3	0	0	1.000	1.000
13	450	10	18	All 6	N(0, 0.1)	0	1.000	1.000
14	300	6	18	50 % 3, 50 % 6	0	N(0, 1)	0.960	0.985
15	300	2	30	All 6	N(0, 0.05)	N(0, 1)	0.934	0.985
16	450	2	30	50 % 3, 50 % 6	0	N(0, 1)	0.879	0.969
17	300	6	18	50 % 3, 50 % 6	N(0, 0.1)	N(0, 0.5)	0.900	0.958
18	300	6	30	50 % 3, 50 % 6	N(0, 0.05)	0	1.000	1.000
19	150	6	18	All 6	0	N(0, 0.5)	0.968	0.987
20	450	6	30	All 3	0	0	1.000	1.000
21	150	2	30	All 3	N(0, 0.1)	N(0, 1)	0.821	0.955
22	450	2	18	50 % 3, 50 % 6	N(0, 0.1)	N(0, 0.5)	0.956	0.990
23	450	6	18	All 3	N(0, 0.05)	N(0, 0.5)	0.783	0.882
24	300	10	18	All 3	N(0, 0.1)	N(0, 1)	0.751	0.862
25	150	6	18	All 6	N(0, 0.05)	N(0, 1)	0.890	0.913
26	150	2	18	All 3	0	0	1.000	1.000
27	450	10	30	All 6	0	N(0, 0.5)	0.889	0.926

da Georgetown University, nos Estados Unidos. A empresa vende diversos itens, como comida congelada, lanches, salgadinhos e doces, além de suprimentos diversos para o lar. Para este estudo, vinte marcas de chocolates foram selecionadas: Almond Joy, Baby Ruth, Butterfinger, Hershey (Almond), Hershey (Plain), Junior Mints, Kit Kat, M & M (Peanut), M & M (Plain), Mars Bar, Milky Way, Mounds Bar, Crunch (Nestlé), Oh Henry!, Payday, Reece's Cups, Snickers, Three musketeers, Twix e York Mint.

Para a coleta de dados, utilizou-se uma tarefa de triagem (*card sorting*), dado que esta abordagem é simples, de fácil entendimento e elimina a necessidade de que itens sejam julgados par-a-par em relação à similaridade. Este método, de forma natural, fornece ainda valores para os parâmetros c^k, necessários à aplicação do PCH. A partir dos dados coletados, as matrizes de dissimilaridades D^k, para todo $k \in \{1, \ldots, m\}$ foram calculadas por meio do método descrito em [Takane, 1980]. Maiores detalhes sobre tarefas de triagem podem ser encontradas em [Coxon, 1999].

Para os objetos (marcas) selecionados, 189 estudantes foram submetidos à tarefa de triagem, devendo separar estes itens em pilhas (categorias), classificando-os de acordo com o critério de similaridade que julgassem adequado. O número de pilhas a ser considerado para a classificação dos objetos também não foi imposto. Isto é, cada estudante poderia estabelecer o número de categorias que achasse conveniente. De posse dos dados gerados a partir da tarefa de triagem, executou-se o algoritmo *VNS* de forma a resolver o PCH para valores de G (número de grupos) variando de 1 a 8. Para cada um destes valores, 10 execuções de 10 minutos foram realizadas. A Tabela 3 apresenta os melhores custos observados para função objetivo para cada um dos valores de G. Utilizou-se o computador já descrito neste texto para a realização destes experimentos, bem como as mesmas configurações previamente descritas para o algoritmo.

De forma a escolher a solução a ser analisada, considerou-se a "regra do cotovelo", abordagem comum na literatura para a determinação do número de *clusters*. De acordo com esta regra,

os dados apresentados na Tabela 3 sugerem que os indivíduos devem ser divididos em dois grupos distintos ($G = 2$), dado que para valores maiores que este o custo da função objetivo não decresce de forma significativa em relação ao ponto anterior ($G - 1$). A Tabela 4 apresenta as estruturas de categorias recuperadas pelo algoritmo para cada um dos grupos de indivíduos. Optou-se por nomear estes dois grupos de indivíduos como "Novatos" e "Especialistas", em razão das estruturas de categorias obtidas. Destaca-se que esta é uma análise subjetiva dos resultados. Portanto, diferentes indivíduos podem chegar a conclusões distintas.

Para o grupo Novatos, os produtos foram particionados em 5 categorias. Observa-se que a classificação obtida em relação aos produtos se baseia fortemente em três aspectos: (a) ingrediente principal (quando este ingrediente é evidente no nome do produto), (b) estrutura física e (c) popularidade. Por exemplo, a Categoria 1 engloba produtos mentolados: Junior Mints e York Mint. O agrupamento destes itens parece ser bem óbvia, dado seus nomes. A Categoria 2 inclui chocolates em barras, tanto pequenas quanto grandes: Hershey plain (mediana), Kit Kat, Hershey (Almond) e Nestle Crunch. A Categoria 3 inclui chocolates pequenos, não em barras, como M&M Peanuts (mediana) M&M Regular e Reese's Cups.

As categorias 4 e 5 representam doces que diferem em relação à popularidade das marcas, não havendo características comuns que os descrevem em relação a ingredientes. A categoria 4 apresenta os chocolates mais populares, visto que os indivíduos declararam que conheciam todos estes itens. Pode-se ver que o chocolate Snickers é a mediana desta categoria. Este produto é, de fato, um dos chocolate mais vendido. Os outros elementos desta categoria incluem Butterfinger, Milky Way, 3 Musketeers e Twix. Por fim, a categoria 5 engloba os chocolates com menor popularidade: Mounds (mediana), Almond Joy, Baby Ruth, Mars, Oh Henry! e Payday.

Tabela 3: Exemplo de aplicação: seleção de modelo (G)

G	Custo	% Melhora
1	2493.59	-
2	2317.92	7.05%
3	2300.35	0.76%
4	2283.16	0.75%
5	2276.37	0.30%
6	2266.53	0.43%
7	2262.32	0.19%
8	2256.51	0.26%

Para o grupo Especialistas, observa-se novamente a categoria dos chocolates mentolados, incluindo Junior Mints e York Mint. No entanto, a classificação de itens para este grupo é mais complexa, pois além de apresentar duas categorias adicionais, percebe-se que estas são baseadas em um conhecimento aprimorado sobre os diferentes ingredientes dos doces e demais características. Para estes indivíduos, a Categoria 2 é composta por chocolates nogados: Milky Way (mediana), Mars, Snickers e Three Musketeers. A Categoria 3 contém chocolates com base em amêndoas e côco: Almond Joy (mediana), Hershey (Almond) e Mounds Bar. A Categoria 4 inclui chocolates crocantes, como Kit Kat (mediana), Nestle's Crunch e Twix, sendo que estes chocolates são conhecidos por esta característica. A Categoria 5 contém os chocolates Reese's cups (mediana) e Butterfinger, dois chocolates conhecidos por serem feitos com manteiga de amendoim. A categoria 6 possui os chocolates Payday (mediana), Baby Ruth e Oh Henry, todos caracterizados pela mistura de caramelo e amendoim. Por fim, a Categoria 7 contém os chocolates de pequeno porte, como M&M, Hershey Plain e M&M Peanuts.

De posse das informações recuperadas pelo algoritmo proposto, os tomadores de decisão podem refletir sobre diversas questões em relação a seus produtos e empresa, como por exemplo, identificar produtos que melhor representam determinada categoria, quais produtos são concorrentes e qual o melhor arranjo para os produtos em gôndolas. O diferencial no uso do PCH para estas

reflexões está no fato de que é possível relacionar as categorias definidas para os produtos a grupos distintos de indivíduos, para os quais se pode traçar perfis. Desta forma, campanhas de *Marketing* podem ser direcionadas de forma a maximizar receitas.

Tabela 4: Exemplo de aplicação: estrutura de categorias

Segmento 1 - Novatos

Nome da Categoria	Mediana	Membros				
Mentolados	Junior Mints	York Mint				
Tabletes	Hershey (Plain)	Kit Kat	Hershey (Almond)	Nestle's Crunch		
Doces Pequenos	M & M (Peanut)	M & M (Plain)	Reese's Cups			
Populares	Snickers	Butterfinger	Milky Way	Three Musketeers	Twix	
Pouco Populares	Mounds Bar	Almond Joy	Baby Ruth	Mars	Oh Henry!	Payday

Segmento 2 - Especialistas

Nome da Categoria	Mediana	Membros		
Mentolados	Junior Mints	York Mint		
Nogados	Milky Way	Mars	Snickers	Three Musketeers
Amêndoas/Côco	Almond Joy	Hershey (Almond)	Mounds Bar	
Crocantes	Kit Kat	Nestle's Crunch	Twix	
Manteiga de Amendoim	Reese's Cups	Butterfinger		
Amendoim e Caramelo	Payday	Baby Ruth	Oh Henry!	
Doces Pequenos	M & M (Plain)	Hershey (Plain)	M & M (Peanut)	

6. Considerações Finais

Apresentou-se neste trabalho um algoritmo baseado na meta-heurística *VNS* para a solução do PCH. Diferentemente dos modelos de *clustering* tradicionais, o PCH permite que se considerem múltiplas matrizes de dissimilaridade como dados entrada sem que haja o processo de agregação. Desta forma, pode-se representar de forma mais apropriada a natureza heterogênea dos dados. A maior contribuição deste trabalho se relaciona ao fato de que o algoritmo apresentado viabiliza a aplicação do PCH a casos reais. Os resultados computacionais apresentados sugerem que o algoritmo proposto é capaz de recuperar a real informação contida nos dados a um custo computacional relativamente baixo, uma vez que comparado ao resolvedor CPLEX para as formulações alternativas propostas por Santi et al. [2015].

Destaca-se que os resultados apresentados neste trabalho não se limitam somente à área acadêmica. Tomadores de decisão atuantes nos mais diversos segmentos, principalmente aqueles relacionados a áreas como Gestão e Marketing, poderão se beneficiar a partir de sua aplicação. Pode-se compreender melhor como os clientes percebem as relações entre diferentes produtos, bem como identificar caraterísticas inerentes a estes consumidores. Trabalhos futuros englobam um estudo mais aprofundados sobre formulações e algoritmos alternativos para o PCH, buscando-se sua resolução de forma exata.

Agradecimentos

Esta pesquisa foi parcialmente financiada pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico do Brasil (Edital Universal), prêmio concedido ao segundo autor e pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, na modalidade de bolsa de doutorado concedida ao primeiro autor.

Referências

Arora, R. (1982). Consumer involvement in retail store positioning. *Journal of the Academy of Marketing Science*, 10(1-2):109–124.

Bijmolt, T. H., Wedel, M., Pieters, R. G., e DeSarbo, W. S. (1998). Judgments of brand similarity. *International Journal of Research in Marketing*, 15(3):249–268.

Blanchard, S. J., Aloise, D., e DeSarbo, W. S. (2012). The heterogeneous p-median problem for categorization based clustering. *Psychometrika*, 77(4):741–762.

Carpenter, G. S. e Nakamoto, K. (1994). Reflections on"consumer preference formation and pioneering advantage". *Journal of Marketing Research*, 31(4):570–573.

Coxon, A. P. M. (1999). *Sorting data: collection and analysis*. Thousand Oaks: Sage.

Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, VII:179–188.

Fortet, R. (1960). Applications de l'algèbre de boole en recherche opérationelle. *Revue Française de Recherche Opérationelle*, 4(14):17–26.

Frey, B. J. e Dueck, D. (2007). Clustering by passing messages between data points. *Science*, 315 (5814):972–976.

Hamilton, R. W., Thompson, D. V., Arens, Z. G., Blanchard, S. J., Haubl, G., Kannan, P., Khan, U., Lehmann, D. R., Meloy, M. G., Roese, N. J., e Thomas, M. (2014). Consumer substitution decisions: An integrative framework. *Marketing Letters*, 25(3):305–317.

Hansen, P. e Jaumard, B. (1997). Cluster analysis and mathematical programming. *Mathematical Programming*, 79(1):191–215.

Hansen, P. e Mladenović, N. (1997). Variable neighborhood search for the p-median. *Location Science*, 5(4):207–226.

Hansen, P., Mladenović, N., e Pérez, J. (2010). Variable neighborhood search: Methods and applications. *Annals of Operations Research*, 175(1):367–407.

Hubert, L. e Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2(1):193–218.

Jain, A. K., Murty, M. N., e Flynn, P. J. (1999). Data clustering: A review. *ACM computing surveys (CSUR)*, 31(3):264–323.

McLachlan, G. J. e Basford, K. E. (1988). Mixture models: Inference and applications to clustering. *Applied Statistics*.

Mladenović, N., Brimberg, J., Hansen, P., e Moreno-Pérez, J. A. (2007). The p-median problem: A survey of metaheuristic approaches. *European Journal of Operational Research*, 179(3):927–939.

Mladenović, N. e Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100.

Ratneshwar, S. e Shocker, A. D. (1991). Substitution in use and the role of usage context in product category structures. *Journal of Marketing Research*, 28(3):281–295.

Santi, E., Aloise, D., e Blanchard, S. J. (2015). O problema de clustering heterogêneo. In *Anais do XLVII SBPO*, p. 2858–2869, Rio de Janeiro. SOBRAPO.

Schweidel, D. A., Bradlow, E. T., e Williams, P. (2006). A feature-based approach to assessing advertisement similarity. *Journal of Marketing Research*, 43(2):237–243.

Takane, Y. (1980). Analysis of categorizing behavior using a quantification method. *Behaviormetrika*, 8(7):75–86.