Heurística Híbrida com Mineração de Dados para o Problema de Agrupamento Capacitado com Centro Geométrico

Marcos Guerine
Instituto de Computação - Universidade Federal Fluminense
Av. General Milton Tavares de Souza, s/nº Boa Viagem - Niterói - RJ - CEP: 24210-346
mguerine@ic.uff.br
Instituto Federal de Educação, Ciência e Tecnologia do Rio de Janeiro
Rua José Breves, 550, Centro - Pinheiral - RJ CEP: 27197-000
marcos.ribeiro@ifrj.edu.br

Murilo Brugger Stockinger, Isabel Rosseti, Alexandre Plastino
Instituto de Computação - Universidade Federal Fluminense
Av. General Milton Tavares de Souza, s/nº Boa Viagem - Niterói - RJ - CEP: 24210-346
murilobs@gmail.com, rosseti@ic.uff.br, plastino@ic.uff.br

RESUMO

Neste trabalho, propõe-se a inserção de mineração de dados em uma heurística híbrida já existente na literatura, baseada em *Clustering Search e Simulated Annealing*, para solucionar o problema de agrupamento capacitado com centro geométrico (*Capacitated Centred Clustering Problem - CCCP*). Experimentos computacionais preliminares mostraram que a heurística híbrida com mineração de dados proposta foi capaz de alcançar soluções de melhor qualidade do que a heurística híbrida original, encontrando quase todos os ótimos conhecidos (já comprovados em trabalhos anteriores) e também melhorando as médias dos resultados reportados para a maioria das instâncias avaliadas, consumindo praticamente o mesmo tempo computacional. Uma análise do comportamento das heurísticas evidenciou que a utilização de padrões minerados conseguiu auxiliar na busca por soluções melhores na heurística híbrida com mineração de dados.

PALAVRAS CHAVE. Metaheurística Híbrida, Agrupamento, CCCP, Mineração de Dados

Tópicos (Metaheurística)

ABSTRACT

In this paper, a hybrid data mining heuristic is proposed to solve the capacitated centred clustering problem, based on an existing heuristic that combines Clustering Search and Simulated Annealing for the problem. Computational preliminary experiments showed that the proposed hybrid data mining heuristic was able to reach better solutions than the original strategy, finding almost all optimal solutions (already proven in previous work) and also improving the average cost of solution for several instances, with almost the same time effort. Besides, it was evidenced in a further analysis of the compared heuristics that patterns mined along the execution of the hybrid heuristic aided the search for better solutions in the proposed method.

KEYWORDS. Hybrid Metaheuristic. Clusterization. CCCP. Data Mining.

Paper topics (Metaheuristic)

1. Introdução

Uma tendência bastante comum na área de pesquisa em otimização é a combinação de uma ou mais componentes de diferentes metaheurísticas com o intuito de aproveitar as vantagens que cada uma tem a oferecer. Conceitos e processos de outras áreas de pesquisa também têm sido utilizados em conjunto com metaheurísticas, tais como técnicas de Mineração de Dados (MD) [Ribeiro et al., 2006]. De acordo com Han e Kamber [2011], Mineração de Dados consiste basicamente na extração de conhecimento de maneira automática, na forma de regras e padrões, de bases de dados de um domínio específico. Em MD, alguns padrões podem ser destacados, tais como regras de associação, padrões sequenciais, agrupamento de dados e conjuntos frequentes.

Ribeiro et al. [2006] propuseram a incorporação de técnicas de mineração de conjuntos frequentes (MCF) à metaheurística GRASP [Gendreau e Potvin, 2010], baseando-se na hipótese de que padrões extraídos de soluções subótimas poderiam conter características dessas boas soluções e, portanto, poderiam ser usados para guiar a busca por melhores soluções. Essa combinação foi aplicada ao problema do empacotamento de conjuntos (PEC), a fim de avaliar o impacto da utilização de padrões na busca realizada em uma heurística baseada em GRASP. Os resultados reportados em Ribeiro et al. [2006] foram promissores, revelando uma melhoria tanto em termos de qualidade de solução quanto em termos de tempo computacional.

A metaheurística *Data Mining* GRASP (DM-GRASP), como ficou conhecida essa combinação da metaheurística GRASP com técnicas de MCF, foi novamente aplicada com sucesso a outros problemas de otimização combinatória, tais como: o problema da maximização da diversidade e o problema de replicação de servidores para transmissão *multicast* confiável [Santos et al., 2008], o problema das *p*-medianas [Plastino et al., 2011; Martins et al., 2014], ao problema de projeto de redes a 2-caminhos [Barbalho et al., 2013] e, recentemente, ao problema do caixeiro viajante com coleta e entrega envolvendo um único tipo de produto [Guerine et al., 2016]. Experimentos computacionais comprovaram novamente o benefício da hibridização da heurística com MD, encontrando soluções melhores em um menor tempo computacional que a heurística original.

Uma característica comum dessas abordagens híbridas com mineração de dados propostas é que elas eram limitadas ao escopo de heurísticas baseadas em GRASP, com estrutura multipartida e iterações independentes, com uma fase de construção e outra de busca local. O trabalho de Guerine et al. [2015] introduziu a hibridização com MD em uma heurística baseada nas metaheurísticas *Clustering Search (CS)* [Oliveira et al., 2013] e *Simulated Annealing (SA)* [Kirkpatrick et al., 1983], cujas estruturas se diferenciam do GRASP. Essa heurística híbrida com MD foi aplicada ao problema de rotulação cartográfica de pontos (PRCP) [Guerine et al., 2015], o que possibilitou aprimorar a heurística estado-da-arte para esse problema em diversas instâncias da literatura. Experimentos computacionais adicionais evidenciaram que a inserção de movimentos que usavam os padrões minerados, substituindo movimentos simplesmente aleatórios do SA, foram capazes de acelerar a busca heurística por melhores soluções.

Nesse contexto, o objetivo deste trabalho é ampliar o escopo da hibridização de técnicas de mineração de dados com as metaheurísticas CS e SA, desenvolvendo uma heurística híbrida com CS, SA e MD, aplicando-a ao problema de agrupamento capacitado com centro geométrico (CCCP, sigla em inglês). A estratégia proposta usa como base a heurística de Chaves e Lorena [2010] para o CCCP. A principal contribuição deste trabalho é investigar a hibridização de técnicas de mineração de dados com uma heurística que possui estrutura diferente daquelas que já foram amplamente exploradas na literatura. Além disso, uma vez que o CCCP possui características diferentes do PRCP, outro desafio é identificar a estrutura da base de dados que será usada para mineração, e também dos padrões a serem minerados, e em que momento esses padrões deverão ser utilizados, levando em consideração não só as particularidades das metaheurísticas CS e SA, mas também do problema de otimização abordado.

O restante do artigo está organizado da seguinte forma. Na Seção 2, o problema de agrupamento capacitado com centro geométrico é apresentado, bem como uma revisão bibliográfica. A

Seção 3 descreve a heurística de Chaves e Lorena [2010] proposto para o CCCP e a Seção 4 apresenta como a técnica de mineração foi inserida nessa heurística e também como foram utilizados os padrões minerados. Na Seção 5, os resultados computacionais preliminares obtidos são comparados com os de Chaves e Lorena [2010] e uma análise do comportamento das estratégias é realizada. Finalmente, a Seção 6 apresenta as conclusões deste trabalho, juntamente com o direcionamento de alguns trabalhos futuros.

2. Descrição do problema e revisão bibliográfica

Clusterização é tratada como um problema de otimização combinatória em [Mulvey e Crowder, 1979; Mulvey e Beck, 1984], cujo objetivo é encontrar o melhor agrupamento de um conjunto de objetos, de maneira que seja minimizada a dissimilaridade entre objetos que estejam contidos em mesmos agrupamentos. Na versão capacitada desse problema, uma quantidade máxima de elementos podem estar associados a cada agrupamento. No presente trabalho, é investigado uma variante do problema de clusterização capacitada, o problema do agrupamento capacitado com centro geométrico (*Capacitated Centred Clustering Problem* – CCCP), introduzida por Negreiros e Palhano [2006]. No CCCP, a dissimilaridade é calculada a partir da soma das distâncias de cada um dos pontos ao respectivo centro do agrupamento. O centro do agrupamento (também denominado centróide) é definido pelo centro geométrico de cada agrupamento. As principais aplicações reais para o CCCP estão no projeto de zonas de coleta de lixo e localização de plataformas de exploração de petróleo em alto mar, mas outras aplicações podem ser encontradas em [Negreiros e Palhano, 2006].

Sejam I um conjunto de objetos a serem agrupados e J um conjunto de agrupamentos. Considerando que x_{ij} é uma variável binária que assume um se o elemento i está associado ao agrupamento j, e zero caso contrário, a seguinte formulação para o CCCP foi proposta por Negreiros e Palhano [2006]:

$$min \sum_{i \in I} \sum_{j \in J} ||a_i - \overline{y_j}||^2 x_{ij} \tag{1}$$

sujeito a:

$$\sum_{j \in J} x_{ij} = 1 \qquad i \in I \tag{2}$$

$$\sum_{i \in I} x_{ij} = n_j \qquad j \in J \tag{3}$$

$$\sum_{i \in I} a_i x_{ij} = n_j \overline{y_j} \qquad j \in J \tag{4}$$

$$\sum_{i \in I} q_i x_{ij} \leq Q_j \qquad j \in J \tag{5}$$

$$\overline{y_j} \in \mathbb{R}^l, n_j \in N, x_{ij} \in \{0,1\} \qquad i \in I, j \in J \tag{6}$$

sendo $\overline{y_j}$ a posição do centróide do agrupamento j no $\mathbb{R}^l$, n_j a quantidade de elementos no agrupamento j, a_i a posição do elemento i no espaço $\mathbb{R}^l$, e $|J| = p$. Cada elemento i possui um peso q_i associado e cada agrupamento j possui uma capacidade máxima Q_j que deve ser respeitada. A função objetivo (1) minimiza a soma das dissimilaridades, isto é, as distâncias de cada ponto a_i ao respectivo centróide de seu agrupamento $\overline{y_j}$. As restrições (2) indicam que cada ponto só pode ser associado a um único agrupamento. As restrições (3) contabilizam a quantidade de elementos por agrupamento e as restrições (4) posicionam os centróides dos agrupamentos $\overline{y_j}$ em seus respectivos centros geométricos, cada um dado pela média das coordenadas dos pontos que a eles pertencem. Por fim, as restrições (5) limitam a capacidade de cada agrupamento. Em (6), são definidos os domínios das variáveis.

Na literatura, alguns trabalhos abordaram diretamente o CCCP, que foi introduzido por Negreiros e Palhano [2006]. Além da modelagem, os autores também apresentaram uma heurística em duas fases. A primeira, baseada no algoritmo de Forgy para construir uma solução inicial e a segunda baseada na metaheurística *Variable Neighborhood Search* (VNS). Instâncias do *benchmark* do problema de clusterização capacitado clássico foram usadas como problemas-teste para o CCCP, a fim de avaliar o algoritmo descrito em Negreiros e Palhano [2006].

Pereira e Senne [2008] propuseram um método de geração de colunas, adaptando uma versão originalmente proposta para o problema das *p*-medianas capacitado, para resolver o CCCP. O algoritmo faz uso de uma relaxação langrangeana/*surrogate* para estabilizar o processo de geração de colunas. Em grande parte do conjunto de instâncias, novas soluções foram reportadas.

Chaves e Lorena [2010] apresentaram uma heurística híbrida baseada nas metaheurísticas *Clustering Search* (CS) e *Simulated Annealing* (SA). O SA é empregado como uma componente que gera soluções para serem inseridas no processo de agrupamento do CS que poderão ser exploradas pelo módulo de busca local. A avaliação das soluções é feita de maneira aproximada em alguns pontos do algoritmo, visando diminuir o seu custo computacional de sempre recalcular os centróides a cada nova solução encontrada. A heurística CS-SA obteve bons resultados tanto nas instâncias de pequeno porte, com até 100 pontos e 6 agrupamentos, quanto nas instâncias maiores, reportando várias novas melhores soluções para o problema.

Em [Chaves e Lorena, 2011], os mesmos autores propuseram uma heurística híbrida baseada em CS e Algoritmo Genético (GA), e o GA é então usado como gerador de soluções para o CS. Essa nova heurística mostrou-se competitiva para o CCCP, conseguindo alcançar as melhores soluções em 18 das 25 instâncias que foram testadas, sendo 11 delas novas soluções encontradas.

Na próxima seção, serão revisados os conceitos das metaheurísticas *Simulated Annealing* e *Clustering Search*, e também será descrita a heurística híbrida apresentada por Chaves e Lorena [2010], que foi escolhida como base da proposta híbrida com mineração de dados do presente trabalho. Essa escolha deve-se ao fato de a heurística de Chaves e Lorena [2010] ser uma das estratégias competitivas para o CCCP e também pelo desafio de introduzir a técnica de mineração de dados em uma metaheurística baseada em CS e SA.

3. Heurística baseada em CS e SA para o CCCP

Simulated annealing (SA) é uma metaheurística baseada em busca local que possui um mecanismo de aceitação de soluções para escapar de ótimos locais [Gendreau e Potvin, 2010]. Ao longo da execução completa da metaheurística SA, também denominada resfriamento, o valor de temperatura T é inicializado com a temperatura inicial T_0 e vai sendo decrementado de acordo com uma taxa de resfriamento α e, para cada valor de temperatura, a solução atual é submetida a movimentos que fazem parte de uma estrutura de vizinhança. Movimentos de melhoria na solução são sempre aceitos, e movimentos de piora são aceitos de acordo com um teste de probabilidade, baseado no algoritmo de Metrópolis [Kirkpatrick et al., 1983]. Tal critério depende da temperatura corrente T e da diferença de custo das soluções sendo comparadas $\Delta(s, s')$. Um número r no intervalo $[0, 1)$ é gerado aleatoriamente e, caso satisfaça à condição $r < e^{\frac{-(\Delta(s,s'))}{T}}$, a piora na solução é aceita. Dessa maneira, movimentos de piora têm uma probabilidade de aceitação maior no início do algoritmo (i.e., temperatura alta), visando diversificar a busca, mas essa chance diminui consideravelmente ao final do algoritmo (i.e., temperatura baixa) convergindo para um ótimo local.

O *Clustering Search (CS)* é uma metaheurística híbrida proposta em Oliveira e Lorena [2007] que busca identificar e explorar regiões promissoras no espaço de busca, dividindo-o em *clusters*[1]. O termo híbrido se deve ao fato de o CS requerer uma heurística de geração de soluções,

[1] Apesar de sinônimos, os termos agrupamento e *cluster* no contexto deste artigo serão relacionados a conceitos diferentes. Para padronizar, o termo agrupamento estará relacionado a um conjunto de elementos do problema de otimização CCCP, enquanto que o termo *cluster* será empregado em referência a um conjunto de soluções que são similares de acordo com a metaheurística CS.

que pode ser baseada em GRASP, *Simulated Annealing (SA)*, Busca Tabu ou outra metaheurística. Ao longo do CS, as soluções geradas são armazenadas em *clusters* e, cada nova solução deve ser incluída no *cluster* mais relacionado de acordo com uma métrica de distância. Cada *cluster* possui uma solução central que o representa (a melhor solução associada), e vai sendo preenchido com soluções até que um limiar seja atingido. Nesse momento, considera-se que esse *cluster* indica um espaço promissor de busca e, então, um procedimento de busca local é aplicado à solução central.

Na metaheurística CS, os *clusters* são caracterizados por uma tripla $(\varsigma_i, \tau_i, \beta_i)$, cujas componentes correspondem, respectivamente, ao centro do *cluster*, seu volume e um indicador de ineficiência. O centro do *cluster* ς_i é a solução que representa o *cluster*. O volume τ_i define a quantidade de soluções que estão associadas ao *cluster* e o indicador de ineficiência β_i representa quantas iterações a busca local pode ser aplicada ao centro do *cluster* sem obter melhorias. No trabalho de Chaves e Lorena [2010], no entanto, β_i não foi considerado.

Antes de detalhar a heurística de Chaves e Lorena [2010], é importante ressaltar um detalhe na representação da solução do CCCP que influencia na descrição das componentes do seu algoritmo. Como a avaliação do custo da função objetivo no CCCP tem um alto custo computacional, uma vez que os centróides não são conhecidos a priori e devem ser recalculados para cada nova solução gerada, Chaves e Lorena [2010] optaram, com o objetivo de diminuir o custo computacional, por computar o custo aproximado das soluções, assim como foi feito em Chaves et al. [2007]. Para esse cálculo aproximado, são definidos pontos representantes para cada agrupamento, denominados medianas. Assim, para cada agrupamento do problema, um dos pontos é escolhido como mediana e o cálculo do custo da solução é feito de maneira a somar as distâncias – que já são conhecidas – dos pontos de cada agrupamento à sua respectiva mediana (em vez de calcular o centróide e só depois somar as distâncias dos pontos para seu respectivo centróide). A Figura 1 apresenta a estrutura de uma solução, onde são representados os dez pontos que caracterizam o problema e suas respectivas medianas $(3, 5, 7)$.

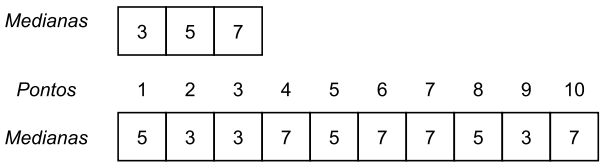


Figura 1: Representação de uma solução do CCCP.

A heurística baseada em CS e SA de Chaves e Lorena [2010] é iniciada com uma solução aleatória, escolhendo pontos como medianas de cada um dos agrupamentos de maneira arbitrária. Em seguida, para cada ponto diferente das medianas e ainda não agrupado, é feita sua associação ao agrupamento mais próximo de acordo com a distância euclidiana para as medianas, de modo que a capacidade do agrupamento não seja excedida. Em seguida, segue-se a estrutura do SA. Para cada valor de temperatura, movimentos são gerados a partir da solução corrente e são submetidos ao critério de aceitação de solução do SA. Caso a solução gerada pelo movimento seja melhor, ela é aceita. Caso contrário, existe uma probabilidade de aceite baseada na temperatura atual e no valor da diferença de custo de solução entre a solução base e a solução encontrada após o movimento.

Cinco movimentos diferentes foram definidos no trabalho de Chaves e Lorena [2010], denominados N^1, N^2, N^3, N^4 e N^5. N^1 é obtido ao trocar a alocação de dois pontos de diferentes *clusters*. N^2 troca uma mediana por um ponto que está associado a ela. N^3 move um ponto de um *cluster* para outro. N^4 é obtido ao trocar uma mediana por qualquer outro ponto. O movimento N^5 é formado pela execução de dois movimentos seguidos do movimento N^1. Cada movimento desse é selecionado aleatoriamente para ser aplicado e a solução gerada é submetida ao critério de aceitação do SA descrito anteriormente.

Para cada valor de temperatura, uma quantidade predeterminada de movimentos são aplicados à solução corrente x. A solução resultante é inserida no *cluster* mais próximo i, i.e., aquele em que o representante ς_i possui a menor distância para a solução x. Sempre que uma solução é inserida em um *cluster*, um processo de reconexão por caminhos é executado com ς_i e x. Tal procedimento se inicia encontrando o conjunto de movimentos necessários para ς_i se aproximar de x, i.e., quais medianas devem ser trocadas para que as soluções tenham o mesmo conjunto de medianas e agrupamentos. O novo centro de solução ς_i é a melhor solução encontrada nesse caminho.

Um *cluster* i é considerado promissor caso tenha atingido seu limite de volume, e assim uma busca local será ativada em ς_i. A busca local é feita da seguinte maneira: para cada agrupamento, a mediana é trocada por um dos pontos a ela associado e uma realocação dos demais pontos é feita em dois momentos. No primeiro momento, os pontos que estavam alocados à mediana trocada são realocados para a mediana mais próxima. Em um segundo momento, avalia-se se vale a pena realocar os pontos de outras medianas para essa nova mediana. Caso seja obtido uma melhoria da solução, o procedimento é repetido em todos os outros agrupamentos até que nenhum benefício seja alcançado.

A heurística CS-SA para o CCCP pode ser descrita no Algoritmo 1. O critério de parada do algoritmo é a quantidade de resfriamentos que serão aplicados. No CS-SA, originalmente, apenas um resfriamento completo é executado. No entanto, visando facilitar a incorporação do processo de mineração de dados na heurística, optou-se por realizar dois resfriamentos completos, fazendo a chamada do minerador entre eles, como será descrito na Seção 4.

Algoritmo 1 CS-SA para o CCCP

CS-SA (γ, τ_{max}, T_0, T_c, α, SA_{max})
Criar γ *clusters* e suas soluções centrais ς_i;
$x \leftarrow$ SoluçãoInicialAleatória(); $x^* \leftarrow x$;
Enquanto critério de parada não satisfeito **faça**
  $T \leftarrow T_0$;
  Enquanto $T > T_c$ **faça**
    $iter \leftarrow 0$;
    Enquanto $iter < SA_{max}$ **faça**
      $iter \leftarrow iter + 1$;
      $k \leftarrow$random$[1, 5]$;
      $x' \leftarrow N^k(x)$;
      Se $f(x') > f(x)$ **então**
        $x \leftarrow x'$;
      senão
        $x \leftarrow x'$, com probabilidade $e^{\frac{-(f(x)-f(x'))}{T}}$;
      Fim-se
    Fim-enquanto
    $T \leftarrow \alpha T$; $i \leftarrow arg \min_{i \in \{1,\ldots,\gamma\}}$ Distância(ς_i, x); $\tau_i \leftarrow \tau_i + 1$; $\varsigma_i \leftarrow min(x, \varsigma_i)$;
    Se $\tau_i = \tau_{max}$ **então**
      $\tau_i = 0$; $x \leftarrow$Busca_Local(ς_i);
    Fim-se
    $x^* \leftarrow min(x^*, \varsigma_i)$;
  Fim-enquanto
Fim-enquanto
Retorne x^*;

4. Incorporando Mineração de Dados: Heurística DM-CS-SA

Na área de Mineração de Dados (MD), existem diversas técnicas de extração de regras e padrões de base de dados. Dentre elas, está a técnica de mineração de conjuntos frequentes (MCF). Como mencionado anteriormente, a proposta deste trabalho é incorporar essa técnica de mineração de dados à heurística híbrida *Clustering Search* e *Simulated Annealing* desenvolvida por Chaves e Lorena [2010], que possui resultados competitivos na literatura para o CCCP, a fim de aprimorá-la. Um dos principais desafios deste trabalho foi identificar qual a estrutura da base de dados a ser minerada e como utilizar os padrões na heurística original.

Normalmente, a incorporação de MD ocorre em duas fases, onde, na primeira fase, a heurística original é executada e soluções de alta qualidade são coletadas e armazenadas em um conjunto elite de soluções. Em seguida, a técnica de MCF é aplicada sobre o conjunto elite, exatamente na metade da execução da heurística, a fim de extrair subconjuntos de elementos (padrões) que representam elementos que ocorrem juntos com uma certa frequência nas soluções do conjunto elite. Na segunda fase, a ideia é usar os padrões minerados para guiar a busca por novas e melhores soluções.

Essa abordagem foi inicialmente proposta por Ribeiro et al. [2006], e aplicada ao problema de empacotamento de conjuntos. Resultados promissores foram obtidos, tanto em termos de qualidade de solução, quanto em termos de tempo computacional, e o mesmo *framework* foi também avaliado para outros problemas [Santos et al., 2008; Plastino et al., 2011; Barbalho et al., 2013; Guerine et al., 2014, 2016]. Outros trabalhos desenvolveram estratégias similares, incorporando MD em outras heurísticas multipartidas [Martins et al., 2014; Maia et al., 2015]. Além disso, versões que aplicam o processo de MD mais de uma vez, denominado como *Multi Data Mining* GRASP (MDM-GRASP), também foram exploradas em [Plastino et al., 2011; Barbalho et al., 2013].

No trabalho de Guerine et al. [2015], a hibridização com MD se deu em uma heurística baseada nas metaheurísticas *Clustering Search* e *Simulated Annealing*. Essa combinação ocorreu de maneira similar ao que foi feito no MDM-GRASP: soluções elite são coletadas ao longo da execução da heurística original e o processo de mineração é aplicado sempre que o conjunto elite de soluções se estabilizar. Os padrões minerados eram utilizados dentro da componente SA, especificamente na fase de geração de soluções. Ao invés de realizar um movimento de troca que era originalmente aleatório, o novo movimento era direcionado, baseado em um dos padrões, visando aproximar a solução corrente daquele padrão. Essa mudança na heurística proporcionou uma rápida convergência para melhores soluções.

Na estratégia aqui proposta, o processo de MD é novamente inserido em uma heurística baseada em CS e SA, pertecente a Chaves e Lorena [2010], que possui estrutura semelhante à heurística de Rabello et al. [2014]. Como foi descrito na Seção 3, a heurística base de Chaves e Lorena [2010] pode ser dividida em duas etapas. Na primeira etapa, soluções são geradas a partir de movimentos aleatórios aplicados à solução corrente, seguindo o critério de aceitação baseado no SA. Na segunda, as soluções geradas são inseridas em *clusters* para posteriormente serem exploradas.

O conjunto contendo as melhores soluções (conjunto elite) é construído a cada iteração do CS contendo as duas etapas, adicionando, por iteração, a melhor solução corrente x ao conjunto elite caso: (i) x seja melhor que a pior solução do conjunto elite, ou (ii) o conjunto elite não esteja completamente cheio (com d soluções). Em ambos os casos, somente são admitidas soluções distintas das que já estão presentes no conjunto elite.

Diferentemente de Guerine et al. [2015], neste trabalho um mapeamento deve ser realizado com as soluções do conjunto elite, a fim de constituir as transações da base de dados que serão mineradas. Esse mapeamento, que é uma proposta do presente trabalho, é necessário devido à estrutura de solução do CCCP, que é formada por um conjunto de p subconjuntos de elementos, onde p é o número de agrupamentos. A Figura 2 explica melhor a necessidade do mapeamento ao ilustrar a estrutura de solução do CCCP para uma instância com dez pontos e três agrupamentos.

É possível observar que cada uma das soluções é um conjunto de três subconjuntos de elementos, e o que diferencia as três soluções do CCCP são as composições dos agrupamentos. Dessa maneira, as soluções que compõem o conjunto elite serão consideradas, para fins da aplicação do algoritmo de mineração, como um conjunto de transações. Cada agrupamento, de cada solução elite, representará uma transação. Assim, a base de dados a ser minerada será composta por $d * p$ transações (ou agrupamentos), sendo d o número de soluções elite e p o número de agrupamentos. Um padrão minerado será representado por um conjunto de elementos frequentes que ocorreram juntos em um mesmo agrupamento.

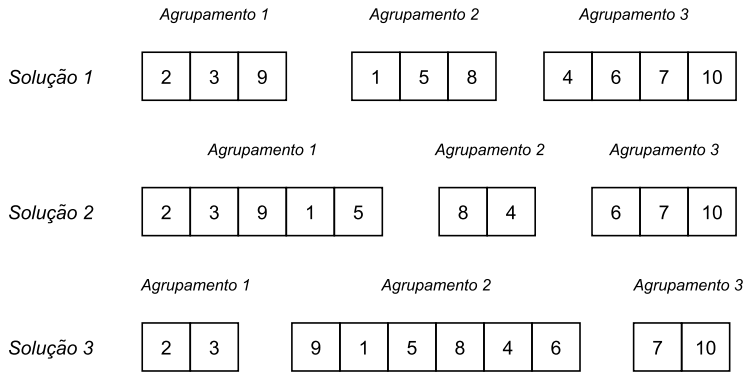


Figura 2: Representação de três soluções distintas do CCCP, divididas por agrupamentos.

Como mencionado na Seção 3, o algoritmo CS-SA realiza dois resfriamentos completos. O processo de mineração é realizado exatamente entre esses dois resfriamentos, e cada padrão extraído representa um conjunto de pontos que ocorreram juntos no mesmo agrupamento em pelo menos sup_{min} soluções do conjunto elite, parâmetro do algoritmo de mineração conhecido como suporte mínimo. Em nossa abordagem, antes de utilizar os padrões minerados de fato, cada ponto é associado ao maior padrão que o contenha. No caso de empates, o desempate é determinado pelo maior valor de suporte do padrão. Essa associação é feita visando facilitar a utilização dos padrões, como será detalhado a seguir.

Para cada movimento de N^1 até N^5, foram criados movimentos adaptados ${N_p}^1$, ${N_p}^2$, ${N_p}^3$, ${N_p}^4$ e ${N_p}^5$, todos levando em consideração os padrões minerados. Para o movimento ${N_p}^1$, seleciona-se um ponto k qualquer e, baseando-se em seu padrão associado, escolhe-se: i) um ponto t que esteja nesse padrão mas que não pertença ao mesmo agrupamento; e ii) um ponto u que esteja no mesmo agrupamento mas não faça parte do padrão. Troca-se t com u, fazendo com que k e t passem a ficar juntos, seguindo o padrão.

Para o movimento ${N_p}^2$, seleciona-se aleatoriamente a mediana m que será trocada. Dentre os pontos alocados a essa mediana, escolhe-se um que esteja no mesmo padrão de m, e a troca é realizada. Para o movimento ${N_p}^3$, um ponto k é sorteado e, baseando-se em seu padrão associado, escolhe-se um ponto p que pertença ao padrão mas que não esteja no mesmo agrupamento. O ponto p é então movido para o mesmo agrupamento de k. Para o movimento ${N_p}^4$, seleciona-se aleatoriamente a mediana m que será trocada. Dentre todos os pontos, escolhe-se um que esteja no mesmo padrão de m, e a troca é realizada. Finalmente, o ${N_p}^5$ é formado pela execução de dois movimentos ${N_p}^1$.

O Algoritmo 2 mostra a heurística híbrida com mineração de dados, que será denominado DM-CS-SA. As modificações em relação ao Algoritmo 1 estão representadas nas linhas 12–16, 27 e 30. O CE é construído na linha 27, a mineração é executada na linha 30 e a utilização dos padrões acontece nas linhas 12–16.

5. Resultados Computacionais

A heurística híbrida *Clustering Search* (CS) e *Simulated Annealing* (SA) de Chaves e Lorena [2010] foi implementada na linguagem C++, bem como a heurística DM-CS-SA proposta neste trabalho. Testes computacionais foram executados em um computador equipado com processador Intel Core i5 CPU 650 @ 3.20GHz, com 8GB de memória RAM e Sistema Operacional Linux Ubuntu versão 14.04. Todos os experimentos foram executados em uma única *thread* e foram considerados três conjuntos de instâncias do CCCP utilizados em Chaves e Lorena [2010]: o primeiro denominado *ta* contendo sete instâncias, o segundo denominado *SJC* contendo seis instâncias, e o

Algoritmo 2 Heurística Híbrida com Mineração de Dados para o CCCP

DM-CS-SA (γ, τ_{max}, T_0, T_c, α, SA_{max}, sup_{min}, d,u)
Criar γ *clusters* e suas soluções centrais ς_i;
$x \leftarrow$ SoluçãoInicialAleatória(); $x^* \leftarrow x$;
$CE \leftarrow \emptyset$;
Enquanto critério de parada não satisfeito **faça**
  $T \leftarrow T_0$;
  Enquanto $T > T_c$ **faça**
    $iter \leftarrow 0$;
    Enquanto $iter < SA_{max}$ **faça**
      $iter \leftarrow iter + 1$;
      $k \leftarrow$random$[1, 5]$;
      Se $\neg$ ExecutouMineração() **então**
        $x' \leftarrow N^k(x)$;
      senão
        $x' \leftarrow N_p^k(x)$;
      Fim-se
      Se $f(x') > f(x)$ **então**
        $x \leftarrow x'$;
      senão
        $x \leftarrow x'$, com probabilidade $e^{\frac{-(f(x)-f(x'))}{T}}$;
      Fim-se
    Fim-enquanto
    $T \leftarrow \alpha T$; $i \leftarrow arg \min_{i \in \{1,\ldots,\gamma\}} \{H_i\}$; $\tau_i \leftarrow \tau_i + 1$; $\varsigma_i \leftarrow min(x, \varsigma_i)$;
    Se $\tau_i = \tau_{max}$ **então**
      $\tau_i = 0$; $x \leftarrow$Busca_Local(ς_i);
    Fim-se
    AtualizaConjuntoEliteDeSoluções(d,x,CE);
    $x^* \leftarrow min(x^*, \varsigma_i)$;
  Fim-enquanto
  ExecutaMineração(CE, sup_{min});
Fim-enquanto
Retorne x^*;

terceiro conjunto denominado *doni* contendo sete intâncias. Vale ressaltar que o código original do CS-SA foi disponibilizado pelos autores e utilizado como base na implementação do DM-CS-SA.

A parametrização da heurística CS-SA em [Chaves e Lorena, 2010] foi obtida em testes preliminares. Os valores escolhidos para T_0, T_c, α e SA_{max} são, respectivamente: 100000, 0,0001, 0,95 e 5000. Além desses, os parâmetros relativos à heurística híbrida DM-CS-SA proposta neste trabalho são: o tamanho do conjunto elite d, a quantidade de padrões minerados u e o valor de suporte mínimo sup_{min}. Os valores desses parâmetros foram escolhidos com base nos trabalhos de Plastino et al. [2011] e Guerine et al. [2015], sendo usados, respectivamente, 10, 100 e 8.

As heurísticas foram executadas dez vezes para cada instância citada da literatura, com sementes diferentes. Foram reportados os custos das melhores soluções, o custo médio das dez soluções e também o tempo médio de execução encontrado pelo CS-SA e pelo DM-CS-SA. Como o código do CS-SA foi disponibilizado pelos autores, foi possível realizar testes com o código original, porém com sementes distintas das que foram empregadas em [Chaves e Lorena, 2010], uma vez que não foi possível saber quais foram as sementes usadas. Embora os resultados encontrados com a execução do CS-SA tenham sido ligeiramente diferentes dos reportados em [Chaves e Lorena, 2010], optou-se por apresentar apenas os resultados alcançados nestas execuções, a fim de realizar uma comparação mais justa com a heurística proposta.

Na Tabela 1, estão os resultados computacionais obtidos por ambas as heurísticas para as instâncias *ta*, *SJC* e *doni*. Essa tabela reporta, inicialmente, o melhor valor conhecido na literatura para cada uma das instâncias (coluna BKS). Em seguida, para cada algoritmo, a melhor solução obtida, o valor médio de solução e o tempo computacional médio das heurísticas CS-SA e DM-CS-SA, relativos às dez execuções, são reportados. Além disso, apresenta-se a diferença percentual (Δ %) dos respectivos valores encontrados pela heurística híbrida com mineração de dados DM-CS-SA em relação à heurística original CS-SA, para cada um dos critérios (melhor custo de solução,

melhor média de custo de solução e melhor tempo médio). Na comparação entre os algoritmos, os valores em negrito representam os melhores resultados obtidos em cada critério e, ao final da tabela, encontra-se a média geral das diferenças percentuais. Resultados iguais, considerados empates, não foram destacados na tabela.

Tabela 1: Resultados computacionais do CS-SA e DM-CS-SA para instâncias do CCCP

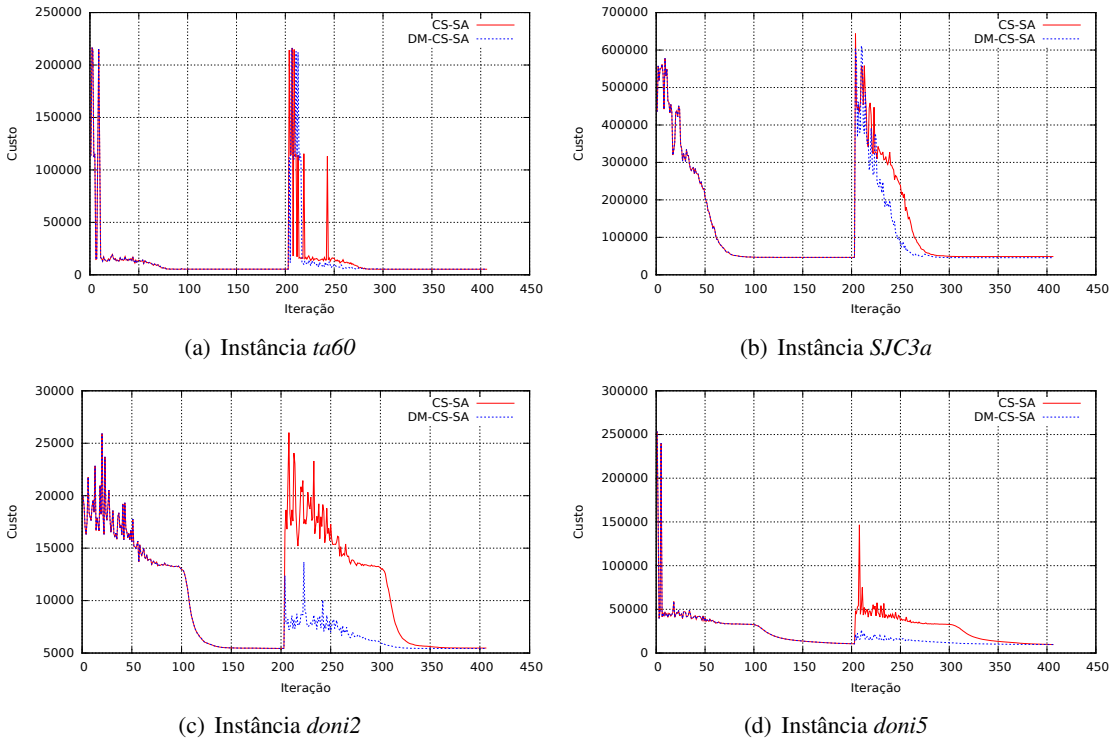
				CS-SA			DM-CS-SA					
Instância	n	p	BKS	MelhorSol	MédiaSol	Tempo	MelhorSol	Δ %	MédiaSol	Δ %	Tempo	Δ %
ta25	25	5	1251,45	1251,45	1251,45	1,24	1251,45	0,00	1251,45	0,00	1,31	4,90
ta50	50	5	4474,52	4474,52	**4478,14**	1,98	4474,52	0,00	4478,87	0,02	2,05	3,48
ta60	60	5	5356,58	5356,58	**5361,97**	2,17	5356,58	0,00	5363,32	0,03	2,32	6,87
ta70	70	5	6240,67	6240,67	6240,76	2,30	6240,67	0,00	6240,76	0,00	2,86	24,36
ta80	80	7	5515,46	5730,28	5739,83	3,36	5730,28	0,00	5739,83	0,00	3,40	1,34
ta90	90	4	8899,05	9069,85	**9069,85**	3,18	9069,85	0,00	9073,18	0,04	3,53	11,17
ta100	100	6	8102,04	8102,04	**8122,70**	3,97	8102,04	0,00	8127,86	0,06	4,24	6,69
SJC1	100	10	17359,75	17359,75	17391,20	4,30	17359,75	0,00	**17387,76**	-0,02	4,54	5,63
SJC2	200	15	33181,65	**33181,65**	33226,73	10,82	33184,75	0,01	**33221,39**	-0,02	10,66	-1,51
SJC3a	300	25	45358,23	45610,18	45799,03	23,53	**45563,07**	-0,10	**45789,61**	-0,02	24,11	2,48
SJC3b	300	30	40661,94	**40975,54**	**41137,65**	27,09	40985,81	0,03	41146,95	0,02	28,07	3,58
SJC4a	402	30	61931,60	62409,77	62737,01	37,46	**62320,15**	-0,14	**62733,65**	-0,01	40,18	7,25
SJC4b	402	40	52214,55	52746,81	53011,44	47,40	**52715,52**	-0,06	**52869,56**	-0,27	49,75	4,97
doni1	1000	6	3021,41	3024,50	3036,78	57,09	3024,50	0,00	3036,78	0,00	54,81	-3,98
doni2	2000	6	6080,70	6373,56	6376,14	149,62	6373,56	0,00	6376,14	0,00	143,32	-4,21
doni3	3000	8	8438,96	8499,58	8568,88	307,62	8499,67	0,00	**8564,96**	-0,05	311,75	1,34
doni4	4000	10	10854,48	**10884,96**	11019,68	545,07	10906,93	0,20	**11011,63**	-0,07	546,06	0,18
doni5	5000	12	11134,94	11208,07	11266,76	868,55	**11181,28**	-0,24	**11263,29**	-0,03	868,44	-0,01
doni6	10000	23	15722,67	16065,01	**16347,72**	4369,24	16065,01	0,00	16388,56	0,25	4263,91	-2,41
doni7	13221	30	18596,74	19779,49	20164,57	8213,34	**19740,41**	-0,20	**20132,76**	-0,16	8171,90	-0,50
								-0,025		-0,011		3,582

Observando a Tabela 1, é possível perceber que, para o grupo de instâncias pequenas (*ta*), ambas as heurísticas chegaram exatamente nas mesmas melhores soluções, com uma vantagem para o CS-SA em relação à media de custo de solução. Em relação ao grupo de instâncias *SJC*, é possível verificar que o desempenho do DM-CS-SA foi superior ao CS-SA, uma vez que encontrou cinco melhores médias de solução em um total de seis instâncias, além de conseguir reportar três melhores soluções, contra duas melhores soluções e apenas uma melhor média de solução para a abordagem original. Para o último grupo, novamente a heurística DM-CS-SA apresentou os melhores resultados, reportando quatro melhores médias de solução em um total de sete instâncias, ao passo que conseguiu apresentar duas melhores soluções, contra uma melhor solução e apenas uma melhor média de solução do CS-SA. Os ganhos percentuais gerais relativos à melhor solução e em relação à média foram ambos negativos, i.e., que foram melhores em relação ao CS-SA, sendo, respectivamente, -0,025% e -0,011%. Todos esses resultados foram obtidos com um tempo bem similar, sendo que o DM-CS-SA é ainda, em alguns casos, mais rápido que o CS-SA.

Para avaliar melhor o comportamento das heurísticas CS-SA e DM-CS-SA, um experimento foi realizado reportando o custo de solução encontrado por iteração de cada uma das duas heurísticas sendo comparadas. Essa avaliação foi feita para uma única execução das heurísticas e, na Figura 3, são apresentados os resultados para as instâncias *ta60*, *SJC3a*, *doni2* e *doni5*. É possível perceber que o comportamento das duas heurísticas é exatamente igual no primeiro resfriamento. No segundo resfriamento, quando os padrões minerados começam a ser utilizados, fica evidente a rápida melhoria no custo de solução após a utilização dos padrões nos movimentos adaptados, se comparado com o CS-SA.

6. Conclusões e Trabalhos Futuros

Neste trabalho, foi proposta a introdução de uma técnica de mineração de dados em uma heurística já conhecida na literatura para resolver o problema de agrupamento capacitado com centro geométrico. Foram realizados experimentos computacionais preliminares em três grupos de instâncias da literatura, e os resultados indicaram o benefício dessa hibridização com mineração de dados, alcançando grande parte das melhores soluções encontradas pela heurística original e também conseguindo melhorar, em vários casos, o custo médio de solução. Nos dois grupos de



(a) Instância *ta60*

(b) Instância *SJC3a*

(c) Instância *doni2*

(d) Instância *doni5*

Figura 3: Gráficos de Custo x Iteração

instâncias maiores, o algoritmo proposto conseguiu cinco melhores soluções contra três da estratégia em comparação. Com relação à média de custo de solução, a heurística híbrida com mineração conseguiu nove melhores médias contra apenas duas da heurística original, em um total de 13 instâncias. Tais resultados foram obtidos em um tempo de execução muito próximo, que em alguns casos foi até mesmo reduzido em relação à heurística base original. Além disso, experimentos complementares analisando o comportamento das duas heurísticas comprovaram a redução do custo de solução logo após a introdução dos padrões, durante a busca por melhores soluções.

Como trabalho futuro, pretende-se ampliar o estudo sobre os padrões minerados, a fim de encontrar formas de utilizá-los no algoritmo DM-CS-SA como, por exemplo, na etapa de busca local. Além disso, deve-se estender os testes para as demais instâncias da literatura, com o objetivo de confirmar o benefício do uso dos padrões.

Referências

Barbalho, H., Rosseti, I., Martins, S. L., e Plastino, A. (2013). A hybrid data mining GRASP with path-relinking. *Computers & Operations Research*, 40(12):3159–3173.

Chaves, A. A., de Assis Correa, F., e Lorena, L. A. N. (2007). *Clustering Search Heuristic for the Capacitated p-Median Problem*, p. 136–143. Springer Berlin Heidelberg, Berlin, Heidelberg.

Chaves, A. A. e Lorena, L. A. N. (2010). Clustering search algorithm for the capacitated centered clustering problem. *Computers & Operations Research*, 37(3):552 – 558.

Chaves, A. A. e Lorena, L. A. N. (2011). Hybrid evolutionary algorithm for the Capacitated Centered Clustering Problem. *Expert Systems with Applications*, 38(5):5013–5018.

Gendreau, M. e Potvin, J.-Y. (2010). *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science*. Springer, $2nd$ edition.

Guerine, M., Rosseti, I., e Plastino, A. (2014). Extending the hybridization of metaheuristics with data mining to a broader domain. In *Proceedings of the 16th International Conference on Enterprise Systems*, p. 395–406, Lisboa, Portugal.

Guerine, M., Rosseti, I., e Plastino, A. (2015). Uma Metaheurística Híbrida com Mineração de Dados para o Problema de Rotulação cartográfica de pontos. In *Anais do Simpósio Brasileiro de Pesquisa Operacional*, p. 2150–2161, Porto de Galinhas, PE, Brasil.

Guerine, M., Rosseti, I., e Plastino, A. (2016). Extending the hybridization of metaheuristics with data mining: Dealing with sequences. *Intelligent Data Analysis*, 5:1–24.

Han, J. e Kamber, M. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, $3rd$ edition.

Kirkpatrick, S., Gelatt, C. D., e Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Maia, M., Plastino, A., e Penna, P. (2015). Incorporando Mineração de Dados a uma Heurística Multi-Start ILS para o Problema de Roteamento de Veículos com Frota Heterogênea. In *Anais do Simpósio Brasileiro de Pesquisa Operacional*, p. 1862–1873, Porto de Galinhas, PE, Brasil.

Martins, D., Vianna, G. M., Rosseti, I., Martins, S. L., e Plastino, A. (2014). Making a state-of-the-art heuristic faster with data mining. *Annals of Operations Research*, p. 1–22. doi: 10.1007/s10479-014-1693-4.

Mulvey, J. M. e Beck, M. P. (1984). Solving capacitated clustering problems. *European Journal of Operational Research*, 18(3):339–348.

Mulvey, J. M. e Crowder, H. P. (1979). Cluster analysis: An application of lagrangian relaxation. *Management Science*, 25(4):329–340.

Negreiros, M. e Palhano, A. (2006). The capacitated centred clustering problem. *Computers & Operations Research*, 33(6):1639–1663.

Oliveira, A. e Lorena, L. (2007). Hybrid evolutionary algorithms and clustering search. In Abraham, A., Grosan, C., e Ishibuchi, H., editors, *Hybrid Evolutionary Algorithms*, volume 75 of *Studies in Computational Intelligence*, p. 77–99. Springer Berlin Heidelberg.

Oliveira, A., Chaves, A. A., e Lorena, L. A. N. (2013). Clustering Search. *Pesquisa Operacional*, 33:105–121.

Pereira, M. e Senne, E. (2008). A column generation method for the capacitated centred clustering problem. *VI ALIO/EURO workshop on applied combinatorial optimization*, p. 1–6.

Plastino, A., Fuchshuber, R., Martins, S. d. L., Freitas, A. A., e Salhi, S. (2011). A hybrid data mining metaheuristic for the p-median problem. *Statistical Analysis and Data Mining*, 4(3): 313–335.

Rabello, R. L., Mauri, G. R., Ribeiro, G. M., e Lorena, L. A. N. (2014). A clustering search metaheuristic for the point-feature cartographic label placement problem. *European Journal of Operational Research*, 234(3):802–808.

Ribeiro, M. H., Plastino, A., e Martins, S. L. (2006). Hybridization of GRASP metaheuristic with data mining techniques. *Journal of Mathematical Modelling Algorithms*, 5:23–41.

Santos, L. F., Martins, S. L., e Plastino, A. (2008). Applications of the DM-GRASP heuristic: a survey. *International Transactions in Operational Research*, 15:387–416.