Uma Metaheurística Híbrida com Mineração de Dados para o Problema de Rotulação Cartográfica de Pontos

Marcos Guerine, Isabel Rosseti, Alexandre Plastino
Instituto de Computação - Universidade Federal Fluminense (UFF)
Rua Passo da Pátria, 156 - Bloco E - CEP 24210-240 - Niterói/RJ - Brasil
{mguerine,rosseti,plastino}@ic.uff.br

RESUMO

Neste trabalho, propõe-se uma heurística híbrida com mineração de dados para solucionar o problema de rotulação cartográfica de pontos, partindo-se de uma heurística estado-da-arte – *Clustering Search* – para o referido problema. Experimentos computacionais mostram que a heurística híbrida com mineração de dados proposta é capaz de alcançar soluções de melhor qualidade do que a heurística original, encontrando quase todos os ótimos conhecidos (e já comprovados em trabalhos anteriores) e também melhorando os resultados reportados na literatura para instâncias com 13.206 pontos e quatro posições candidatas.

PALAVRAS CHAVE. Metaheurística Híbrida, Rotulação Cartográfica, PRCP, Mineração de Dados, Área de classificação principal (Metaheurística).

ABSTRACT

In this paper, a hybrid data mining heuristic is proposed to solve the point-feature cartographic label placement problem, based on a state-of-the-art heuristic – Clustering Search – for the problem. Computational experiments showed that the hybrid heuristic was able to reach better-costs solutions than the original strategy, finding almost all optimal solutions (already proven in previous works) and improving the best results reported in the literature for the set of instances with 13,206 points and four candidate positions.

KEYWORDS. Hybrid Metaheuristic, Map Labelling, PFCLP, Data Mining, Main area (Metaheuristic).

1. Introdução

Posicionar rótulos em um mapa de tal forma que eles claramente representem os objetos ao qual estão associados é uma importante tarefa em cartografia. De acordo com Marks e Shieber (1991), antes da automatização, a tarefa de rotulação despendia mais de 50% do tempo de produção de um mapa.

O problema de rotulação cartográfica de pontos (PRCP) é um problema de otimização combinatória (POC) amplamente estudado que consiste em atribuir rótulos de texto para cada objeto de um mapa, respeitando preferências e convenções cartográficas, tendo como objetivo evitar sobreposições entre diferentes rótulos. A ideia principal é prover clareza na visualização e entendimento do mapa a ser rotulado (Rabello *et al*., 2014).

O PRCP é uma variação do *cartographic label placement problem*, o qual normalmente abrange três diferentes tarefas: atribuir rótulos a objetos em formato de regiões (e.g.

continentes ou países), objetos em formato de linha (e.g. rios ou rodovias) e, por fim, objetos em formato de ponto (e.g. cidades ou hospitais), estudo de caso deste trabalho. Existem diversas aplicações do PRCP em cartografia automatizada, geoprocessamento e sistemas de informações geográficas (Yamamoto *et al*., 2002).

Atingir um bom nível de visualização em um mapa está diretamente relacionado com a maneira de associar esses rótulos aos pontos do mapa. Cada ponto deve possuir um conjunto de posições candidatas, cada uma com sua padronização cartográfica, indicando a posição preferencial. Em (Christensen *et al*., 1995), foi proposto um padrão cartográfico com um conjunto de oito posições cartográficas para cada rótulo.

Para a resolução de um POC de grande porte, tal como o PRCP, geralmente são empregados métodos heurísticos. As metaheurísticas são procedimentos de alto nível que coordenam heurísticas específicas para resolver esses problemas, normalmente em um tempo computacional viável. Um ramo na pesquisa sobre metaheurísticas estuda a combinação de duas ou mais componentes de diferentes metaheurísticas clássicas, a fim de obter metaheurísticas híbridas mais eficientes (Talbi, 2002).

Nos últimos anos, têm sido explorada também a combinação de conceitos e processos da área de mineração de dados com metaheurísticas. Han e Kamber (2011) definem mineração de dados como sendo a extração automática de conhecimento, expresso em forma de regras e padrões, a partir de bases de dados.

A hibridização de técnicas de mineração dados com metaheurísticas, explorada inicialmente por Ribeiro *et al*. (2004, 2006) combinando a metaheurística GRASP com técnicas de mineração de conjuntos frequentes, alcançou resultados importantes em diversos problemas de otimização (Santos *et al*., 2008; Plastino *et al*., 2011; Barbalho *et al*., 2013; Martins *et al*., 2014), conseguindo aprimorar heurísticas estado-da-arte. Estratégias semelhantes foram descritas em (Dalboni *et al*., 2003; Santos *et al*., 2006), combinando estratégias evolutivas com uma variação do algoritmo Apriori.

A ideia dessa hibridização consiste em armazenar soluções de alta qualidade obtidas por uma heurística base e usá-las como base de dados para a execução da técnica de mineração de dados. A referida técnica extrai padrões, que representam características importantes das soluções elite, para serem usados posteriormente para guiar a exploração do espaço de soluções do POC em questão.

Assim, este trabalho tem por objetivo combinar um processo de mineração de dados a uma heurística já existente para o PRCP, proposta por (Rabello *et al*., 2014) e baseada na metaheurística *Clustering Search*. Os resultados computacionais mostraram que a abordagem híbrida com mineração de dados proposta neste trabalho permite melhorar a qualidade das soluções quando comparada com a heurística original e também com os resultados da literatura.

O restante do artigo está organizado da seguinte forma. A Seção 2 apresenta o PRCP, bem como uma revisão bibliográfica. A Seção 3 descreve o *Clustering Search* de Rabello *et al*. (2014) proposto para o PRCP e a Seção 4 apresenta como a técnica de mineração foi inserida nessa heurística. Na Seção 5, os resultados computacionais obtidos são comparados com os de Rabello *et al*. (2014) e uma análise do comportamento das estratégias é realizada. Finalmente, a Seção 6 apresenta as conclusões deste trabalho, juntamente com a proposta de alguns trabalhos futuros.

2. Descrição do problema e revisão bibliográfica

O PRCP é um problema de otimização combinatória $\mathcal{NP}$-difícil (Marks e Shieber, 1991), que consiste em atribuir rótulos a pontos específicos de uma região a ser rotulada, de modo que as sobreposições entre rótulos sejam minimizadas ou evitadas. Na literatura, é possível encontrar o PRCP com três funções objetivo semelhantes, porém definidas de maneira distinta.

Na primeira versão, o PRCP pode ser modelado como o problema do conjunto independente máximo (Zoraster, 1990) e, nesse caso, deve-se obter a rotulação com o maior número de rótulos sem conflitos, considerando que pontos com sobreposição não devem ser rotulados. A segunda versão busca encontrar a rotulação com a maior quantidade de rótulos sem conflitos (livres), exigindo que todos os pontos possuam rótulos (Christensen *et al*., 1995). Na última versão, o objetivo é minimizar o número de conflitos quando todos os pontos são rotulados (Ribeiro e Lorena, 2006).

A modelagem do PRCP apresentada em (Ribeiro e Lorena, 2006) é descrita a seguir. Considera-se o grafo de conflito formado pelas posições candidatas e seus conflitos com os demais pontos. Cada posição candidata é representada por uma variável binária x_{ij}, $i \in \{1, \ldots, N\}, j \in \{1, \ldots, P_i\}$, onde P_i é a quantidade de posições candidatas do ponto i e N é o número de pontos que serão rotulados. $x_{ij} = 1$ indica que o rótulo será inserido na posição candidata j do ponto i e, caso contrário, $x_{ij} = 0$. Cada posição candidata também está associada a um custo, representado por w_{ij}.

Considera-se o conjunto S_{ij}, que contém todos os pares $\{k, t\}$ das posições candidatas x_{kt} que possuem conflito com x_{ij}. Para cada par $\{k, t\} \in S_{ij}$, em que $k \in \{1, \ldots, N\} : k > i$ e $t \in \{1, \ldots, P_k\}$, existe uma variável y_{ijkt} que representa o conflito entre x_{ij} e x_{kt}. Dadas essas definições, a formulação do PRCP é a seguinte:

$$min \sum_{i=1}^{N} \sum_{j=1}^{P_i} \left(w_{ij} x_{ij} + \sum_{(k,t) \in S_{ij}} y_{ijkt} \right) \tag{1}$$

sujeito a:

$$\sum_{j=1}^{P_i} x_{ij} = 1, \qquad \forall i = 1, \ldots, N \tag{2}$$

$$x_{ij} + x_{kt} - y_{ijkt} \leq 1, \qquad \begin{array}{r} \forall i = 1, \ldots, N, \\ \forall j = 1, \ldots, P_i \\ (k, t) \in S_{ij} \end{array} \tag{3}$$

$$x_{ij}, x_{kt} \text{ e } y_{ijkt} \in \{0, 1\}, \qquad \begin{array}{r} \forall i = 1, \ldots, N, \\ \forall j = 1, \ldots, P_i, \\ (k, t) \in S_{ij} \end{array} \tag{4}$$

A Equação 2 garante que somente uma das posições candidatas será selecionada, enquanto a Equação 3 relaciona as variáveis que representam os conflitos com as que representam as posições candidatas. As Restrições 4 indicam o domínio das variáveis do problema e a função objetivo, definida na Equação 1, busca minimizar os conflitos.

Em seu trabalho, Marks e Shieber (1991) provaram que o PRCP é $\mathcal{NP}$-difícil ao reduzir de maneira polinomial o PRCP ao problema Planar 3-SAT. Na sequência, um dos primeiros a desenvolver algoritmos heurísticos para o PRCP foi Christensen *et al*. (1995), apresentando duas estratégias: a primeira, baseada no método de gradiente descendente na forma discreta, e a segunda, baseada em *Simulated Annealing*.

Verner *et al*. (1997) apresentaram um algoritmo genético com máscaras que estimulavam a troca de posições candidatas sobrepostas durante o operador de *crossover*. Yamamoto *et al*. (2002) desenvolveram um método heurístico baseado em busca tabu, que iterativamente realizava trocas entre posições candidatas de acordo com uma lista de candidatas. As trocas respeitavam a lista tabu de proibição, com tamanho variável durante o algoritmo, e quando todos os candidatos gerados estavam na lista tabu, um critério de aspiração era adotado para escolher o candidato mais antigo da lista.

Uma nova modelagem matemática para o problema foi proposta em (Ribeiro e Lorena, 2006), assim como heurísticas baseadas em relaxações lagrangeanas dessa nova formulação. Cravo *et al*. (2008) desenvolveram uma heurística baseada em GRASP para resolver o PRCP. Nos trabalhos de Ribeiro e Lorena (2008a,b), foram apresentadas, respectivamente, uma relaxação lagrangeana com *clusters* e uma geração de colunas, conseguindo provar o ótimo e atingir melhores soluções para algumas instâncias.

No trabalho de Alvim e Taillard (2009), uma heurística baseada em POPMUSIC foi proposta. Essa heurística divide cada instância em subproblemas e utiliza uma busca tabu (baseada no trabalho de (Yamamoto *et al*., 2002)) para resolver cada um dos subproblemas. Além de melhorar as soluções dos problemas testes utilizados até então, novas instâncias com 13.206 pontos a serem rotulados foram propostas.

Mauri *et al*. (2010) trabalharam em um novo modelo matemático para o problema e também desenvolveram uma heurística baseada em decomposição lagrangeana. No referido trabalho, foram reportadas melhores soluções e valores ótimos para instâncias com, no máximo, 1000 pontos. O modelo apresentado foi aprimorado em (Ribeiro *et al*., 2009), fortalecendo a formulação por meio da inserção de inequações válidas.

Na próxima seção, será revisada com mais detalhes a heurística híbrida apresentada recentemente por Rabello *et al*. (2014), que foi escolhida como base da proposta híbrida com mineração de dados do presente trabalho por ser uma estratégia estado-da-arte para o PRCP e pelo desafio de introduzir a técnica de mineração de dados em uma metaheurística com estrutura diferente das que já foram combinadas anteriormente na literatura.

3. Metaheurística Clustering Search para o PRCP

O *Clustering Search (CS)* é uma metaheurística híbrida elaborada em (Oliveira e Lorena, 2007) que busca identificar e explorar regiões promissoras no espaço de busca, dividindo-o em *clusters*. O termo híbrido se deve ao fato de o CS requerer uma heurística de geração de soluções, que pode ser baseada em GRASP, *Simulated Annealing (SA)*, Busca Tabu ou outra metaheurística.

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, e vai sendo preenchido com soluções até que um limiar seja atingido. Nesse momento, acredita-se que esse *cluster* indica um espaço promissor de busca e, então, um procedimento de busca local é aplicado à solução central.

Os *clusters* são caracterizados por uma tupla $(\varsigma_i, \tau_i, \beta_i)$, que são, respectivamente, o 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.

A metaheurística CS para o PRCP apresentada em (Rabello *et al.*, 2014), descrita no Algoritmo 1, pode ser detalhada da seguinte maneira: inicialmente, uma solução aleatória (cada ponto recebe uma rotulação arbitrária) é construída e, em seguida, cada um dos γ *clusters* – dado de entrada – é inicializado com uma solução central também gerada randomicamente (linhas 2 e 3).

Na sequência, a solução inicial passa a ser a solução corrente x do algoritmo, que é então submetida a uma componente responsável pela geração de soluções, baseada em *Simulated Annealing (SA)*. A cada iteração entre as linhas 8 e 16, diversos movimentos da vizinhança $N(x)$ são aplicados à solução atual x, que somente serão aceitos se: (i) houver melhoria do custo da nova solução após o movimento – linha 11 – ou (ii) se um número aleatório $r < e^{\frac{-(f(x)-f(x'))}{T}}$ (critério de aceitação do SA, considerando a temperatura atual T e a diferença de valores da função objetivo entre x e x') – linha 14. Se aceito, cada movimento $N(x)$ altera uma posição candidata de um ponto escolhido aleatoriamente. Após aplicar SA_{max} movimentos, x é associada ao *cluster* i cuja solução central ς_i tem a maior similaridade com x de acordo com a distância de Hamming (linhas 17 e 18), que contabiliza quantas posições candidatas diferentes x e ς_i possuem.

Algoritmo 1 *Clustering Search* para o PRCP

1: CS (γ, τ_{max}, β_{max}, T_0, T_c, α, SA_{max})
2: Criar γ *clusters* e suas soluções centrais ς_i;
3: $x \leftarrow$ SoluçãoInicialAleatória(); $x^* \leftarrow x$;
4: **Enquanto** critério de parada não satisfeito **faça**
5: $T \leftarrow T_0$;
6: **Enquanto** $T > T_c$ **faça**
7: $iter \leftarrow 0$;
8: **Enquanto** $iter < SA_{max}$ **faça**
9: $iter \leftarrow iter + 1$;
10: $x' \leftarrow N(x)$;
11: **Se** $f(x') > f(x)$ **então**
12: $x \leftarrow x'$;
13: **senão**
14: $x \leftarrow x'$, com probabilidade $e^{\frac{-(f(x)-f(x'))}{T}}$;
15: **Fim-se**
16: **Fim-enquanto**
17: $T \leftarrow \alpha T$; $i \leftarrow arg \min_{i \in \{1,\ldots,\gamma\}} \{H_i\}$; $\tau_i \leftarrow \tau_i + 1$;
18: $\varsigma_i \leftarrow max(x, \varsigma_i)$;
19: **Se** $\tau_i = \tau_{max}$ **então**
20: $\tau_i = 0$; $x \leftarrow$Busca_Local(ς_i);
21: **Se** $f(x) = f(\varsigma_i)$ **então**
22: $\beta_i \leftarrow \beta_i + 1$;
23: **Se** $\beta_i = \beta_{max}$ **então**
24: $\beta_i \leftarrow 0$; $\varsigma_i \leftarrow N(\varsigma_i)$;
25: **Fim-se**
26: **Fim-se**
27: **Fim-se**
28: $x^* \leftarrow max(x^*, \varsigma_i)$;
29: **Fim-enquanto**
30: **Fim-enquanto**
31: **Retorne** x^*;

Após associar x ao *cluster* mais similar ς_i, seu volume τ_i é incrementado na linha

17. Se o volume atual atingiu o volume máximo τ_{max}, um mecanismo de busca é aplicado ao centro do *cluster* ς_i (linha 20). Caso a busca local seja realizada mais de β_{max} vezes sem melhorias, uma perturbação é realizada em ς_i com o objetivo de escapar de ótimos locais (linha 24). A perturbação efetua basicamente um movimento aleatório da vizinhança $N(\varsigma_i)$ entre posições candidatas, forçando a diversificação nesse ponto do algoritmo.

A temperatura T inicia em T_0 e é atualizada de acordo com a taxa de resfriamento α do SA na linha 17 e, caso a solução corrente x seja melhor que a melhor solução encontrada até então x^*, atualiza-se x^* com x. Finalmente, o critério de parada do CS, como descrito em (Rabello *et al*., 2014), aplica reaquecimentos sucessivos à temperatura corrente se a melhor solução tiver sido alterada nos últimos dois minutos. Esses passos se repetem até que a temperatura atinja o limiar de congelamento T_c e o critério de parada seja satisfeito.

4. Incorporando Mineração de Dados: Heurística DM-CS

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* desenvolvida por Rabello *et al*. (2014), que possui os melhores resultados da literatura para o PRCP até então, afim de aprimorá-la. Um dos principais desafios deste trabalho foi encontrar um ponto adequado para inserção do processo de MD.

Normalmente, essa incorporação ocorre da seguinte maneira. Em uma 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 a extrair subconjuntos de elementos (padrões) que representam elementos que ocorrem com uma certa frequência no conjunto elite. Por fim, a ideia é usar os padrões minerados para guiar a busca na segunda fase de execução da heurística.

Essa abordagem foi inicialmente proposta por Ribeiro *et al*. (2004, 2006), combinando mineração de conjuntos frequentes com a metaheurística GRASP para aplicar ao problema de empacotamento de conjuntos. Resultados promissores foram obtidos tanto em termos de qualidade de solução, quanto em tempo computacional. O mesmo *framework* foi também avaliado em outros problemas, tais como o problema da maximização da diversidade (Santos *et al*., 2005), o problema da replicação de servidores *multicast* confiável (Santos *et al*., 2006), o problema das p-medianas (Plastino *et al*., 2009, 2011; Martins *et al*., 2014), o problema de projeto de redes a 2-caminhos (Barbalho *et al*., 2013) e recentemente ao problema do caixeiro viajante com coleta e entrega envolvendo único tipo de produto (Guerine *et al*., 2013, 2014).

Na maioria dos trabalhos anteriores, a inserção da técnica de mineração de dados ocorreu na metaheurística GRASP, dando origem à versão híbrida, denominada DM-GRASP. No trabalho de Martins *et al*. (2014), apesar de a heurística base escolhida ter sido uma combinação de heurísticas de construção, busca local e reconexão por caminhos, a estrutura era multipartida e possuía características semelhantes à estrutura do GRASP.

Neste trabalho, busca-se inserir MD na heurística híbrida *Clustering Search* de (Rabello *et al*., 2014), que será denominada DM-CS. Como foi descrito na Seção 3, a estratégia de Rabello *et al*. (2014) pode ser dividida basicamente em duas etapas. Na primeira etapa, soluções são geradas a partir de movimentos aleatórios aplicados à solução atual, seguindo

o critério de aceitação baseado no *Simulated Annealing* a cada movimento. Na segunda, as soluções geradas são inseridas em *clusters* para posteriormente serem exploradas.

O conjunto elite de soluções é construído a cada iteração do CS contendo as duas etapas, adicionando, por iteração, a melhor solução corrente s ao conjunto elite caso: (i) s 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.

Na abordagem proposta neste trabalho, o procedimento de mineração é chamado sempre que o conjunto elite se torna estável, seguindo ideia similar a do *Multi Data Mining* GRASP explorado em (Plastino *et al*., 2011; Barbalho *et al*., 2013). Um conjunto elite é considerado estável quando permanece ϕ iterações sem modificação.

Algoritmo 2 Heurística Híbrida com Mineração de Dados para o PRCP

DM-CS (γ, τ_{max}, β_{max}, T_0, T_c, α, SA_{max}, ϕ, sup_{min}, d)
Criar γ *clusters* e suas soluções centrais ς_i;
$x \leftarrow$ SoluçãoInicialAleatória(); $x^* \leftarrow x$;
$CE \leftarrow \emptyset$; $iter_{lm} \leftarrow 0$;
Enquanto critério de parada não satisfeito **faça**
  $T \leftarrow T_0$;
  Enquanto $T > T_c$ **faça**
    $iter \leftarrow 0$;
    Enquanto $iter < SA_{max}$ **faça**
      $iter \leftarrow iter + 1$;
      Se ¬ExecutouMineração() **então**
        $x' \leftarrow N(x)$;
      senão
        $x' \leftarrow N_p(x)$;
      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
    $T \leftarrow \alpha T$; $i \leftarrow arg \min_{i \in \{1,\ldots,\gamma\}} \{H_i\}$; $\tau_i \leftarrow \tau_i + 1$;
    $\varsigma_i \leftarrow max(x, \varsigma_i)$;
    Se $\tau_i = \tau_{max}$ **então**
      $\tau_i = 0$; $x \leftarrow$Busca_Local(ς_i)
      Se $f(x) = f(\varsigma_i)$ **então**
        $\beta_i \leftarrow \beta_i + 1$;
        Se $\beta_i = \beta_{max}$ **então**
          $\beta_i \leftarrow 0$; $\varsigma_i \leftarrow N(\varsigma_i)$;
        Fim-se
      Fim-se
    Fim-se
    Se AtualizaConjuntoEliteDeSoluções(d,x,CE); **então**
      $iter_{lm} \leftarrow 0$;
    senão
      $iter_{lm} \leftarrow iter_{lm} + 1$;
    Fim-se
    $x^* \leftarrow max(x^*, \varsigma_i)$;
  Fim-enquanto
  Se $iter_{lm} \geq \phi$ **então**
    $p \leftarrow$ ExecutaMineração(CE, sup_{min});
    $iter_{lm} \leftarrow 0$;
  Fim-se
Fim-enquanto
Retorne x^*;

Após a chamada ao processo de mineração, os padrões são extraídos. Cada padrão representa um conjunto de posições candidatas que ocorreram juntas em pelo menos

sup_{min} soluções do CE, parâmetro conhecido como suporte mínimo.

Por fim, em nossa abordagem, o padrão p de maior tamanho passa a ser utilizado como guia na etapa de geração de novas soluções, utilizando-se a vizinhança modificada $N_p(x)$ ao invés de $N(x)$. A diferença está no fato de que realiza $N(x)$ movimentos aleatórios nessa etapa, enquanto que na $N_p(x)$, a cada movimento um ponto é selecionado para ser alterado e, se esse ponto ocorrer no padrão p, sua posição candidata é inserida na solução corrente, de acordo com o critério de aceitação do *Simulated Annealing*. Dessa maneira, ao invés de realizar trocas aleatórias nas posições candidatas, escolhem-se as posições candidatas que são frequentes em soluções de boa qualidade do CE, representadas por p, direcionando a busca no espaço de soluções.

O Algoritmo 2 mostra a heurística híbrida com mineração de dados. As modificações em relação ao Algoritmo 1 estão representadas nas linhas 11–15, 33–37 e 40–43. O CE é construído nas linhas 33–39, a mineração é executada nas linhas 40–43 e a utilização dos padrões acontece nas linhas 11–15. Além disso, considera-se como critério de parada para o DM-CS o tempo médio de execução do CS por instância.

5. Resultados Computacionais

A heurística *Clustering Search* (CS) de Rabello *et al*. (2014) foi implementada na linguagem C++, bem como a heurística DM-CS proposta neste trabalho. Testes computacionais foram executados em um computador equipado com processador Intel®Core^{TM} 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 consideradas as instâncias do PRCP propostas em (Yamamoto *et al*., 2002) e (Alvim e Taillard, 2009), com 25, 100, 250, 500, 750, 1.000 e 13.206 pontos e quatro posições candidatas. Vale ressaltar que o código original do CS foi disponibilizado pelos autores e utilizado como base na implementação do DM-CS.

A parametrização original da heurística CS (Rabello *et al*., 2014) foi mantida. Os valores escolhidos para γ, τ_{max}, β_{max}, T_0, T_c, α e SA_{max} são, respectivamente, 10; 7; 4; 40.000; 0.01; 0.975 e 12.000. Além desses, os parâmetros relativos à heurística híbrida com mineração de dados proposta neste trabalho, que são, o tamanho do conjunto elite d, o valor de suporte mínimo sup_{min} e a quantidade de iterações necessárias para estabilizar o conjunto elite ϕ foram estipulados, respectivamente, em 10, 8 e 5% do total de iterações realizadas a cada resfriamento do SA. Esses valores foram baseados na parametrização encontrada em (Plastino *et al*., 2011).

Ambas as heurísticas foram executadas dez vezes para cada instância 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 e pelo DM-CS. Nas instâncias de menor porte, com 25, 100, 250, 500, 750 e 1.000 pontos a serem rotulados, apenas em uma dentre as 133 instâncias o DM-CS não encontrou o melhor valor conhecido na literatura (Rabello *et al*., 2014), ficando a 0,11% do mesmo. Em relação à média de solução, a diferença percentual média, ficou em 0,01% a favor do CS em relação ao DM-CS. Tais resultados foram obtidos com o mesmo tempo computacional, com diferença percentual menor que 0,001%.

Nas instâncias previamente analisadas, o módulo de mineração de dados não obteve resultados expressivos, uma vez que na grande maioria das instâncias a heurística CS ori-

ginal já alcançava as melhores soluções da literatura muito antes do módulo de mineração ser ativado. Isso acontece devido à facilidade de resolução desses problemas de pequeno porte, o que não acontece com as instâncias de 13.206 pontos, que serão analisadas a seguir em separado.

Como o código do CS foi disponibilizado pelos autores, foi possível realizar testes com o código original, porém com sementes distintas das que foram empregadas no artigo original, uma vez que não foi possível saber quais foram as sementes usadas. Embora os resultados encontrados com a execução do CS tenham sido próximos aos reportados pelo artigo, as melhores soluções não foram exatamente as mesmas. Para facilitar a comparação, na Tabela 1, está reportado, para cada instância, o melhor valor conhecido (BKS), levando em consideração tanto o trabalho de (Rabello *et al*., 2014), quanto os melhores resultados da nossa execução do CS.

Na Tabela 1, estão os resultados computacionais obtidos por ambas as heurísticas para as instâncias maiores, com 13.206 pontos. Essa tabela reporta, para cada instância e cada algoritmo, a melhor solução obtida, o valor médio de solução relativo às dez execuções e o tempo computacional médio das heurísticas CS e DM-CS. Vale ressaltar que o tempo de execução do DM-CS foi estipulado de acordo com o tempo médio despendido pela execução do CS. Além disso, apresenta a diferença percentual ($\Delta_{\%} = \frac{Valor - BKS}{BKS}$, onde $Valor$ é o custo da melhor solução do respectivo algoritmo) dessas heurísticas em relação aos BKS. Na comparação entre os algoritmos, os valores em negrito representam os melhores resultados obtidos e, ao final da tabela, encontra-se a média geral das diferenças percentuais.

Tabela 1. Resultados computacionais do CS e DM-CS para instâncias com 13.206 pontos

		CS (Rabello *et al*., 2014)				DM-CS				
Instâncias	**BKS**	**Melhor**	**$\Delta_{\%}$Melhor**	**Média**	**$\Delta_{\%}$Média**	**Melhor**	**$\Delta_{\%}$Melhor**	**Média**	**$\Delta_{\%}$Média**	**Tempo[a]**
13206_1	12479	12478	-0.01	12472.2	-0.05	**12493**	0.11	**12487.1**	0.06	477.45
13206_2	12113	12113	0.00	12102.0	-0.09	**12141**	0.23	**12133.7**	0.17	607.09
13206_3	11895	11895	0.00	11882.4	-0.11	**11926**	0.26	**11914.2**	0.16	624.59
13206_4	11702	11698	-0.03	11692.3	-0.08	**11733**	0.26	**11726.4**	0.21	680.64
13206_5	11758	11758	0.00	11746.0	-0.10	**11808**	0.43	**11790.9**	0.28	712.92
13206_6	10889	10889	0.00	10875.4	-0.12	**10945**	0.51	**10931.1**	0.39	716.79
13206_7	10491	10487	-0.04	10476.1	-0.14	**10547**	0.53	**10530.6**	0.38	543.92
13206_8	10821	10821	0.00	10802.9	-0.17	**10871**	0.46	**10859.2**	0.35	663.53
13206_9	10806	10805	-0.01	10795.6	-0.10	**10859**	0.49	**10844.9**	0.36	628.49
13206_10	10397	10391	-0.06	10378.5	-0.18	**10454**	0.55	**10432.4**	0.34	602.95
13206_11	10068	10068	0.00	10051.2	-0.17	**10131**	0.63	**10110.4**	0.42	652.42
13206_12	9955	9941	-0.14	9928.5	-0.27	**9992**	0.37	**9976.0**	0.21	529.51
13206_13	10787	10782	-0.05	10776.3	-0.10	**10871**	0.78	**10849.8**	0.58	807.44
13206_14	10121	10121	0.00	10102.5	-0.18	**10181**	0.59	**10168.7**	0.47	618.45
13206_15	9699	9699	0.00	9686.6	-0.13	**9768**	0.71	**9755.2**	0.58	723.61
13206_16	9306	9290	-0.17	9280.5	-0.27	**9368**	0.67	**9352.6**	0.50	781.07
13206_17	10255	10250	-0.05	10236.0	-0.19	**10302**	0.46	**10277.6**	0.22	472.71
13206_18	9509	9501	-0.08	9483.3	-0.27	**9578**	0.73	**9564.7**	0.59	644.29
13206_19	9074	9063	-0.12	9054.3	-0.22	**9159**	0.94	**9135.9**	0.68	829.29
13206_20	8598	8598	0.00	8585.0	-0.15	**8701**	1.20	**8685.6**	1.02	902.03
Média			-0.04		-0.15		**0.55**		**0.40**	

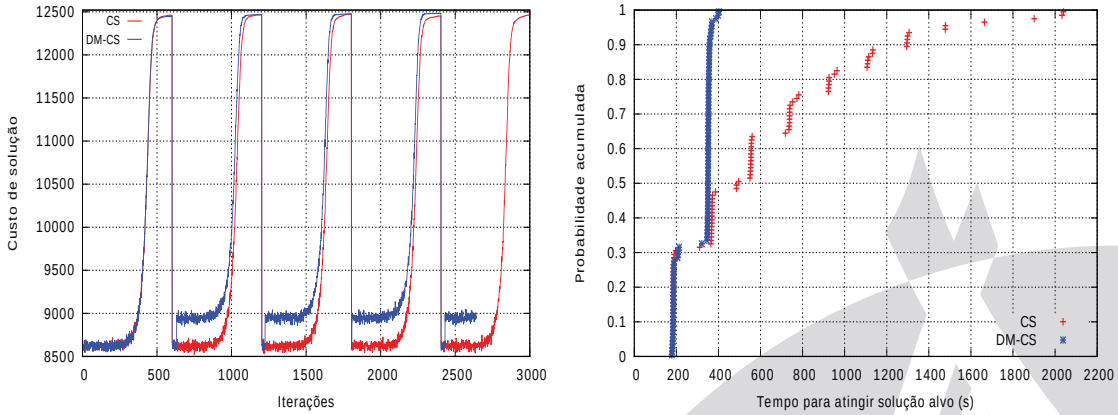
[a] - Tempos computacionais médios, em segundos, para ambos algoritmos

Observando a Tabela 1, é possível notar que, para todas as instâncias, a heurística DM-CS apresenta melhores soluções e melhores valores médios de solução, sendo que o ganho percentual geral relativo à melhor solução do DM-CS foi em média igual a 0,55%, e, relativo ao valor médio de solução, de 0,40%.

Alguns experimentos adicionais foram realizados a fim de ilustrar e comparar o

comportamento dos dois algoritmos analisados neste trabalho. A Figura 1(a) mostra como se comportam ambos os algoritmos, reportando por iteração os valores de solução obtidos por cada uma das estratégias. Esse teste foi realizado para a instância 13206_1 durante uma execução de uma semente.

É possível notar que os dois algoritmos CS e DM-CS possuem um comportamento similar durante toda sua execução. Os resfriamentos totais e os reaquecimentos acontecem a cada 600 iterações. À medida que a temperatura vai decrescendo, ambas estratégias atingem soluções melhores. Na Figura 1(a), as minerações do DM-CS ocorreram nas iterações 138, 631, 1224, 1806 e 2431. Com exceção da primeira, em todas as outras o custo de solução do DM-CS melhorou bastante em relação ao CS logo após a utilização do padrão. Esse comportamento corrobora a hipótese de que os padrões extraídos estão auxiliando na busca por melhores soluções, atuando como uma componente de intensificação.



(a) Análise de Comportamento - Instância 13206_1

(b) TTTPlot - Instância 13206_12

Figura 1. Análise de comportamento e gráfico *time-to-target* em instâncias de grande porte.

A Figura 1(b) representa outra comparação entre os dois algoritmos abordados, baseada nos gráficos *Time-to-Target Plots* (TTT-plots) (Aiex *et al.*, 2007), que são usados para analisar o comportamento de algoritmos com componentes aleatórias. Esses gráficos mostram a probabilidade acumulada, no eixo das ordenadas, de um algoritmo encontrar uma solução melhor ou igual a uma solução alvo prefixada, em um tempo de execução definido no eixo das abscissas. Nesse experimento, foram feitas 100 execuções de ambas estratégias, adotando-se a solução alvo de valor 9928.

É possível notar que o comportamento da versão híbrida supera o CS puro. A probabilidade do DM-CS encontrar, por exemplo, a solução alvo em 400 segundos é de quase 100% enquanto que para o CS essa mesma probabilidade está em torno de 47%.

6. Conclusões e Trabalhos Futuros

Neste trabalho, foi proposta a introdução de uma técnica de mineração de dados em uma heurística previamente proposta e estado-da-arte para o problema de rotulação cartográfica de pontos. Foram realizados experimentos computacionais em diversas instâncias da literatura, com o número de pontos variando entre 25 e 13.206, cada um com quatro posições candidatas.

Os resultados indicaram o benefício da introdução de mineração de dados na heurística original, alcançando melhores soluções, se comparadas com as da literatura. O

ganho médio em relação às melhores soluções foi 0.55% e para os valores médios de solução esse ganho foi de 0.40%. Além disso, experimentos complementares comprovaram a superioridade da versão híbrida com mineração no que diz respeito à convergência do método. Essas avaliações demonstram o bom desempenho do método proposto.

Como trabalhos futuros, pretende-se ampliar o estudo sobre os padrões minerados, a fim encontrar novas utilidades para eles no algoritmo como, por exemplo, inserindo-os na busca local. Além disso, deve-se estender os testes para as demais instâncias da literatura, que possuem oito posições candidatadas, com o objetivo de confirmar o benefício da estratégia híbrida com mineração em instâncias mais difíceis.

Referências

Aiex, R. M., Resende, M. G. C., e Ribeiro, C. C. (2007). TTT plots: A perl program to create time-to-target plots. *Optimization Letters*, 1:355–366.

Alvim, A. C. F. e Taillard, É. D. (2009). POPMUSIC for the point feature label placement problem. *European Journal of Operational Research*, 192:396–413.

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

Christensen, J., Marks, J., e Shieber, S. (1995). An Empirical Study of Algorithms for Point-feature Label Placement. *ACM Transactions on Graphics*, 14:203–232.

Cravo, G. L., Ribeiro, G. M., e Lorena, L. A. N. (2008). A greedy randomized adaptive search procedure for the point-feature cartographic label placement. *Computers & Geosciences*, 34:373–386.

Dalboni, F. L., Ochi, L. S., e Drummond, L. M. A. (2003). On improving evolutionary algorithms by using data mining for the oil collector vehicle routing problem. Em *International Network Optimization Conference*, páginas 182–188, Évry, França.

Guerine, M., Rosseti, I., e Plastino, A. (2013). Incorporando Mineração de Dados a uma Heurística GRASP/VND para o Problema do Caixeiro Viajante com Coleta e Entrega Envolvendo único Tipo de Produto. Em *Anais do XLV Simpósio Brasileiro de Pesquisa Operacional (XLV SBPO)*, páginas 1970–1981, Natal, RN, Brasil.

Guerine, M., Rosseti, I., e Plastino, A. (2014). Extending the Hybridization of Metaheuristics with Data Mining to a Broader Domain. Em *Proceedings of the 16^{th} International Conference on Enterprise Systems*, páginas 395–406, Lisboa, Portugal.

Han, J. e Kamber, M. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 3^a edição.

Marks, J. e Shieber, S. (1991). The computational complexity of cartographic label placement. Technical report. Harvard University, USA.

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

Mauri, G. R., Ribeiro, G. M., e Lorena, L. A. N. (2010). A New Mathematical Model and a Lagrangean Decomposition for the Point-feature Cartographic Label Placement Problem. *Computers & Operation Research*, 37:2164–2172.

Oliveira, A. e Lorena, L. (2007). Hybrid Evolutionary Algorithms and Clustering Search. Em Abraham, A., Grosan, C., e Ishibuchi, H., editores, *Hybrid Evolutionary Algorithms*, volume 75 of *Studies in Computational Intelligence*, páginas 77–99. Springer Berlin Heidelberg.

Plastino, A., Fonseca, E. R., Fuchshuber, R., Martins, S. L., Freitas, A. A., Luis, M.,

e Salhi, S. (2009). A hybrid data mining metaheuristic for the p-median problem. Em *Proceedings of the SIAM International Conference on Data Mining*, páginas 305–316.

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

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*, 234:802 – 808.

Ribeiro, G. M., Constantino, M. F., e Lorena, L. A. N. (2009). Um estudo sobre desigualdades válidas para o problema de maximização de rótulos livres. Em *Anais do XLI Simpósio Brasileiro de Pesquisa Operacional (XLI SBPO)*, páginas 2807–2818, Porto Seguro, Brasil.

Ribeiro, G. M. e Lorena, L. A. N. (2006). Heuristics for cartographic label placement problems. *Computers & Geosciences*, 32:229–243.

Ribeiro, G. M. e Lorena, L. A. N. (2008a). Column generation approach for the point-feature cartographic label placement problem. *Journal of Combinatorial Optimization*, 15:147–164.

Ribeiro, G. M. e Lorena, L. A. N. (2008b). Lagrangean Relaxation with Clusters for Point-feature Cartographic Label Placement Problems. *Computers & Operations Research*, 35:2129–2140.

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.

Ribeiro, M. H., Trindade, V. F., Plastino, A., e Martins, S. L. (2004). Hybridization of GRASP Metaheuristics with Data Mining Techniques. Em *Proceedings of the ECAI workshop on hybrid metaheuristics*, páginas 69–78.

Santos, H. G., Ochi, L. S., Marinho, E. H., e Drummond, L. M. (2006a). Combining an Evolutionary Algorithm with Data Mining to Solve a Single-Vehicle Routing Problem. *Neurocomputing*, 70:70–77.

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.

Santos, L. F., Milagres, R., Albuquerque, C. V., Martins, S. L., e Plastino, A. (2006b). A Hybrid GRASP with Data Mining for Efficient Server Replication for Reliable Multicast. Em *Proceedings of the IEEE GLOBECOM conference*, páginas 1–6.

Santos, L. F., Ribeiro, M. H., Plastino, A., e Martins, S. L. (2005). A Hybrid GRASP with Data Mining for the Maximum Diversity Problem. Em *Proceedings of the International Workshop on Hybrid Metaheuristics*, volume 3636 of *Lecture Notes in Computer Science*, páginas 116–127, Barcelona, Spain.

Talbi, E.-G. (2002). A Taxonomy of Hybrid Metaheuristics. *Journal of Heuristics*, 8:541–564.

Verner, O. V., Wainwright, R. L., e Schoenefeld, D. A. (1997). Placing Text Labels on Maps and Diagrams using Genetic Algorithms with Masking. *INFORMS Journal on Computing*, 9:266–275.

Yamamoto, M., Camara, G., e Lorena, L. A. N. (2002). Tabu Search Heuristic for Point-Feature Cartographic Label Placement. *Geoinformatica*, 6:77–90.

Zoraster, S. (1990). The solution of large 0-1 integer programming problems encountered in automated cartography. *Operations Research*, 5:752–759.