



CLUSTERING SEARCH APLICADO AO PROBLEMA DE ROTULAÇÃO CARTOGRÁFICA COM DISPERSÃO DE RÓTULOS

Eliseu Junio Araújo

Universidade Federal de São Paulo – UNIFESP, Instituto de Ciência e Tecnologia
Rua Talin, 330, CEP 12231-280, São José dos Campos, SP, Brasil
araujo.eliseu@unifesp.br

Antônio Augusto Chaves

Universidade Federal de São Paulo – UNIFESP, Instituto de Ciência e Tecnologia
Rua Talin, 330, CEP 12231-280, São José dos Campos, SP, Brasil
antonio.chaves@unifesp.br

Luiz Antônio Nogueira Lorena

Instituto Nacional de Pesquisas Especiais – INPE, Laboratório Associado de Computação e
Matemática
Av. dos Astronautas, 1758, CEP 12227-010, São José dos Campos, Brasil
luizlorena54@gmail.com

RESUMO

O Problema de localização de rótulos cartográficos se refere ao posicionamento automático de rótulos em um mapa, diagrama ou objeto gráfico. Neste trabalho é aplicado uma metaheurística híbrida, o *Clustering Search* (CS) para o problema com o intuito de encontrar soluções que possam trazer maior legibilidade ao mapa. O método tem o objetivo de maximizar o número de rótulos sem conflitos e também maximizar a distância entre rótulos em conflitos. Logo, para comprovar sua eficiência, são realizados testes com instâncias encontradas na literatura que são amplamente usadas em outros trabalhos relacionados a este problema. Os resultados apresentados mostram que o método atingiu bons resultados em medidas comparativas com modelos da literatura.

PALAVRAS-CHAVE. Problema de Localização, Rótulos Cartográficos, *Clustering Search*.

Área principal: Metaheurísticas

ABSTRACT

The point-feature cartographic label placement problem refers to the automatic placement of labels on a map, diagram, or graphic object. In this work a hybrid metaheuristic, the *Clustering Search* (CS), is applied to the problem in order to find solutions that can bring greater readability to the map. The method aims to maximize the number of labels without conflicts and also to maximize the distance between labels in conflicts. Therefore, to test its efficiency, tests are performed with instances found in the literature that are widely used in other works related to this problem. The results presented show that the method achieved good results in comparative measurements with literature models.

KEYWORDS. Placement Problem, Cartographic labels, *Clustering Search*

Main area: Metaheuristics



1. Introdução

O problema de posicionar rótulos de forma automática em um objeto gráfico é chamado de Problema de Localização de Rótulos Cartográficos. Esta é uma situação presente em Sistemas de Informações Geográficas em que normalmente abre-se a necessidade de se localizar através de identificadores três tipos de localidades: pontos (representando montes, aglomerações urbanas), linhas (representando rios, ruas, etc.) e polígonos (representando vilas, lagos). Nos dispositivos gráficos, os identificadores são conhecidos como “rótulos”. Porém, os mapas costumam apresentar diversas localidades, o que ocasiona, por vezes em quantidade considerável de rótulos em um mesmo espaço e assim ocorrem sobreposições entre eles. Tais sobreposições precisam ser evitadas objetivando uma maior legibilidade. Um exemplo sobre o estado do Espírito Santo – Brasil é mostrado na Figura 1. Observa-se a existência de sobreposições, pontuadas pelas setas em preto, que prejudicam a clareza da representação das localidades.



Figura 1 - Exemplo de posicionamento de rotulação de pontos (Ribeiro e Lorena, 2006)

O problema de posicionar rótulos em um dispositivo gráfico evitando suas sobreposições é usualmente chamado na literatura como *Point Feature Cartographic Label Placement* (PFCLP).

Christensen *et al.* (1995) fez uma formalização de como uma rotulação pode ocupar uma posição possível. Esta formalização considera quatro posições candidatas, mostradas na Figura 2.

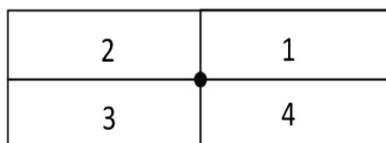


Figura 2 - Posições candidatas sugeridas em Christensen *et al.* (1995)

A Figura 3 mostra as posições candidatas e suas possíveis sobreposições que são representadas por um grafo de conflitos. As posições candidatas são representadas pelos vértices e os possíveis conflitos são representados pelas arestas. A Figura 3a representa um conjunto de quatro pontos (p_i), com quatro posições candidatas e a Figura 3b indica o grafo de possíveis conflitos com o vértice $v_{i,j}$ representando a posição candidata $j \in \{1, \dots, 4\}$ do ponto $i \in \{1, \dots, 4\}$.

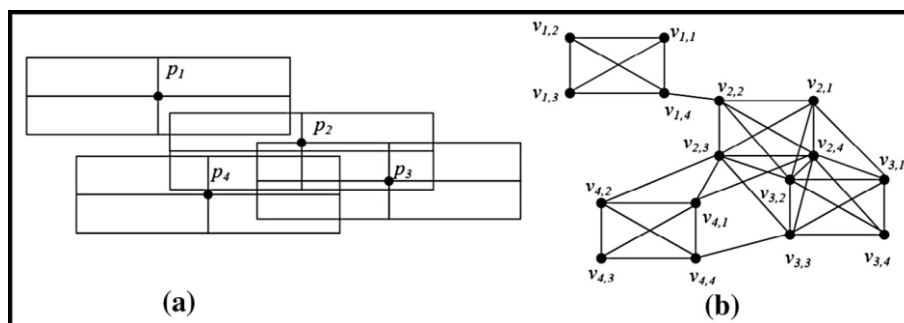


Figura 3 - Grafo de conflitos: (a) Exemplo com 4 pontos e (b) grafo de conflitos (Gomes, Ribeiro e Lorena, 2013)

Na literatura são encontradas três abordagens principais para o PFCLP: *Maximum Independent Vertex Set Problem* (MIVSP), *Maximum Number of Conflict Free Labels Problem* (MNCFLP) e *Minimum Number of Conflicts Problem* (MNCP) (Ribeiro e Lorena, 2006). O MIVSP procura rotular o maior número de localidades que não tem sobreposições, porém, devido ao problema real ser complexo, é possível deixar alguns pontos sem rótulos. No MNCFLP todos os pontos obrigatoriamente precisam ser rotulados, maximizando o número de rótulos sem sobreposições, contudo desconsidera a legibilidade do mapa. O MNCP faz com que todos os pontos devam ser rotulados e reduz o número de conflitos para prover uma solução melhor do ponto de vista da legibilidade.

A Figura 4 pontua a diferença entre MNCFLP e MNCP. A Figura 4b e 4c mostra duas soluções para a instância da Figura 4a. Ambas as soluções são viáveis e equivalentes para esta abordagem uma vez que o MNCFLP leva em conta o número de rótulos livres. Em contrapartida, se for considerado que este problema é um MNCP, a solução da Figura 4b é pior do que a da Figura 4c pois conduz para uma solução de menor legibilidade por apresentar um maior número de arestas no grafo de conflitos.

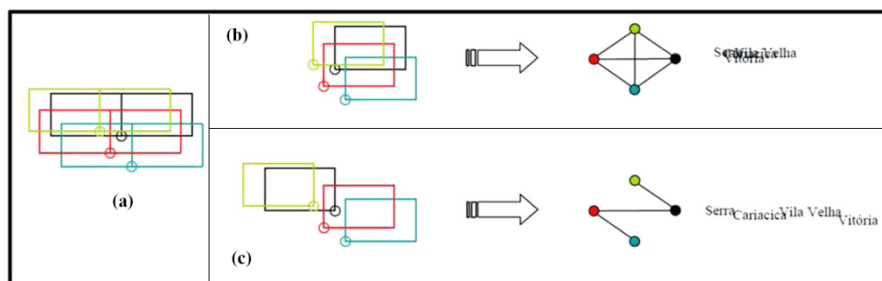


Figura 4 - Comparação entre MNCFLP e MNCP (Ribeiro, 2007)

As técnicas de solução para MNCFLP e MNCP frequentemente tendem a resolver o PFCLP atuando diretamente sobre a estrutura do problema, ou seja, pontuando cada rótulo para evitar conflitos, considerando as possíveis sobreposições. Inspirado nessas duas abordagens, ou seja, procurando rotular todos os rótulos, mas priorizando também a legibilidade do mapa, Araujo *et al* (2015) propõe uma nova formulação matemática para o problema de localização de rótulos em que objetiva uma melhor dispersão dos mesmos, o que também contribui para evitar sobreposições. Entretanto, foi observado um erro no cálculo de função objetivo das soluções obtidas pelo método de Araujo *et al* (2015). Desta forma, refez os testes computacionais sobre o modelo proposto com o intuito de corrigir esses erros. Nós também propomos a aplicação do método híbrido *Clustering Search* (CS) para maximizar a distância entre rótulos que estiverem em conflito, assim como maximizar o número de rótulos sem conflito. Testes computacionais são realizados com instâncias disponíveis na literatura para avaliar a eficiência do método.

Este trabalho é organizado como se segue. Uma revisão da literatura é apresentada na seção 2. O modelo de dispersão alvo de comparação nos testes computacionais é apresentado na seção 3. A seção 4 apresenta o método *Clustering Search*. A seção 5 apresenta os detalhes da



aplicação do método ao problema. A seção 6 explana os resultados computacionais e a seção 7 as considerações finais.

2. Revisão bibliográfica para o PFCLP

Há diversos trabalhos na literatura que abordam as três principais abordagens para o PFCLP.

O MNCP foi apresentado por Ribeiro e Lorena (2006) como uma abordagem que visa melhorar a legibilidade quando todos os pontos precisam ser rotulados e sobreposições serem evitadas. Eles introduziram algumas heurísticas de relaxações lagrangeanas e uma formulação matemática. Ribeiro e Lorena (2008) trouxeram dois modelos matemáticos para o MNCP e formalizaram a proposta de uma relaxação lagrangeana com agrupamentos que obtiveram resultados melhores do que outras técnicas na literatura. A principal diferença entre as formulações se baseia em como o grafo de conflitos é construído: um modelo matemático é baseado sobre posições candidatas e pontos enquanto a outra é um modelo baseado somente em posições candidatas.

Sobre o MIVSP, Zoraster (1990) mostra um modelo matemático de programação linear, inteira e binária e introduz restrições com posições candidatas fictícias, trazendo penalizações de custo considerável na função objetivo. Strijk *et al.* (2000) trazem modelos matemáticos fazendo uso de restrições de corte com desigualdades para todas as cliques máximas (subgrafos completos) no grafo de conflitos. Para as instâncias, eles aplicaram diversas heurísticas: *Diversified Neighborhood Search*, Busca Tabu, *Simulated Annealing* e *k-opt*. Ribeiro *et al.* (2011) expuseram uma decomposição Lagrangeana para o problema e encontraram a solução ótima para quase todas as instâncias testadas da literatura, sendo o método uma boa alternativa para outros problemas modelados como MIVSP.

Sobre a abordagem com MNCFLP, Christensen *et al.* (1995) introduziram dois algoritmos baseados em uma forma discreta de gradiente de descida combinada a técnica de *Simulated Annealing*. Yamamoto *et al.* (2002) trazem um algoritmo de Busca Tabu que alcançou bons resultados em instâncias reais de teste. Alvim e Tailard (2009) resgatam a Busca Tabu proposta por Yamamoto *et al.* (2002) para testar a metaheurística POPMUSIC sobre instâncias grandes. A POPMUSIC inicialmente constrói subproblemas para uma solução do problema principal. Estes subproblemas são resolvidos e montados para fazer uma nova solução e o método é repetido para obter melhores soluções até um dado critério de parada ser alcançado.

É necessário ressaltar que os trabalhos Rabello *et al.* (2014) e Gomes *et al.* (2016) fazem abordagens interessantes sobre o PFCLP e foram usados como base para este trabalho devido aos seus resultados. Rabello *et al.* (2014) apresenta a metaheurística *Clustering Search* para explorar o PFCLP enquanto Gomes *et al.* (2016) faz uso de um Algoritmo Genético Construtivo. Esses dois trabalhos da literatura tratam de formulações similares a formulação usada como base para este trabalho, ou seja, a formulação MD. Eles trabalham com o PFCLP, mas a MD difere delas quando trabalha com a distância entre rótulos como uma penalidade dada a função objetivo. A MD procura dar uma penalidade considerável na função objetivo de acordo com a quantidade de conflitos e também quando a distância entre rótulos em conflitos é muito pequena. Outros trabalhos não fazem esse tipo de tratamento em suas formulações.

3. Modelo de dispersão para o PFCLP

Nesta seção é apresentado um modelo de dispersão para o PFCLP que é alvo de testes neste trabalho e de comparação com o CS nos testes computacionais. Esse modelo foi apresentado no trabalho de Araujo *et al.* (2015). Ele foi novamente testado e novos resultados foram encontrados, por isso está sendo explanado neste trabalho.

Considere no modelo de dispersão os seguintes parâmetros e variáveis:

- N : conjunto de pontos para serem rotulados;
- i : ponto para ser rotulado, tal que $i \in N$;
- P_i : conjunto de posições candidatas de um ponto i ;



- j : posição candidata, tal que $j \in P_i$;
- $x_{i,j}$: variável binária concernente a decisão de escolher ou não um rótulo, sendo que $x_{i,j} = 1$ indica a escolha de uma posição candidata j para o ponto rotulado i .
- $a_{i,j}$: custo referente as preferências cartográficas definidas por Christensen *et al.* (1995);
- (i, j) : par de índices representante de um ponto i e rótulo j ;
- $S_{i,j}$: conjunto de pares de índices que possuem conflitos com o par (i, j) ;
- (k, t) : par de índices pertencente ao conjunto $S_{i,j}$ com $k \in N, k > i$ e $t \in P_k$;
- $d_{i,j,k,t}$: distância entre os rótulos (i, j) e (k, t) .

O modelo, chamado de MD, é formulado como se segue.

$$\text{Max } \sum_{i=1}^n \sum_{j=1}^{P_i} (a_{i,j} \cdot x_{i,j} - w_i) \quad (1)$$

Sujeito a:

$$\sum_{j=1}^{P_i} x_{i,j} = 1 \quad \forall i = 1 \dots N \quad (2)$$

$$\sum_{(k,t) \in S_{i,j}} (1/(1 + d_{i,j,k,t})) (x_{i,j} + x_{k,t} - 1) \leq w_i \quad \forall i = 1 \dots N, \quad (3)$$

$$\forall j = 1 \dots P_i, \quad (k, t) \in S_{i,j}$$

$$x_{i,j}, x_{k,t}, y_{i,j,k,t} \in \{0,1\} \quad \forall i = 1 \dots N, \forall j = 1 \dots P_i, (k, t) \in S_{i,j} \quad (4)$$

$$w_i \geq 0 \quad \forall i = 1 \dots N \quad (5)$$

A função objetivo (1) realiza o somatório das posições candidatas escolhidas e os respectivos custos dessas preferências cartográficas. A variável w_i faz parte das restrições (3) e (5) e tem o objetivo de contribuir para que caso aconteçam conflitos inevitáveis, o conflito de maior distância entre rótulos seja escolhido para que haja maior dispersão. Como w_i também é calculada na função objetivo com valor negativo, logo é priorizado que ela tenha o menor valor possível para que a função objetivo possa alcançar o maior valor.

As restrições (2) asseguram que somente uma posição candidata será escolhida para cada ponto rotulado.

Nas restrições (3) nota-se que o valor $(1/(1 + d_{i,j,k,t}))$ somente será considerado no somatório se existir conflito entre os rótulos (i, j) e (k, t) . Como busca-se o menor valor possível para w_i , as maiores distâncias entre conflitos inevitáveis acabam por ganhar prioridade. As restrições (4) e (5) indicam o domínio das variáveis.

A Figura 5 mostra um exemplo da aplicação do modelo MD, em que são apresentados 5 pontos que precisam ser rotulados. Inevitavelmente, o ponto 5 sofrerá conflito com um dos rótulos dos outros pontos. Logo, é escolhido o rótulo que possui maior distância para esse conflito, visando comprometer o mínimo possível o valor de função objetivo.

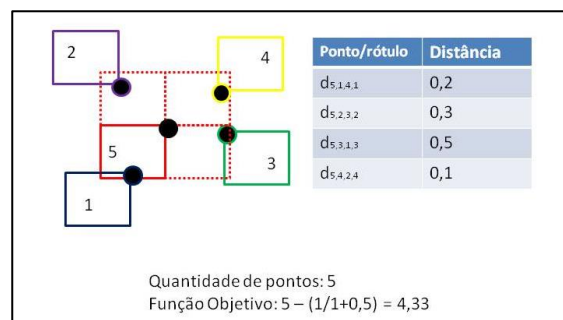


Figura 5 - Exemplo de aplicação do Modelo MD (Araujo *et al.*, 2015)



4. Clustering Search

O *Clustering Search* (Oliveira *et al*, 2013) é um método híbrido que busca combinar metaheurísticas e heurísticas de busca local, em que a busca é intensificada somente em regiões do espaço de busca que merecem atenção especial (regiões promissoras). O CS introduz uma inteligência e prioridade para a escolha de soluções para aplicar a busca local, ao invés de escolher aleatoriamente ou aplicar busca local em todas as soluções. Consequentemente, é esperado uma melhora no processo de convergência com uma diminuição no esforço computacional pois há uma aplicação mais racional das heurísticas.

O CS se atém a localizar áreas de busca promissora construindo-as em *clusters*. Um *cluster* é definido por um centro, c , que é geralmente, inicializado aleatoriamente e, posteriormente, ele tende progressivamente a pontos promissores no espaço de busca. O número de clusters, NC , deve ser fixada a priori.

O CS pode ser explicitado em quatro partes conceitualmente independentes: a metaheurística (SM), o componente de agrupamento (IC), um módulo de análise (AM) e a busca local (LS).

A Figura 6 apresenta um fluxograma conceitual do CS.

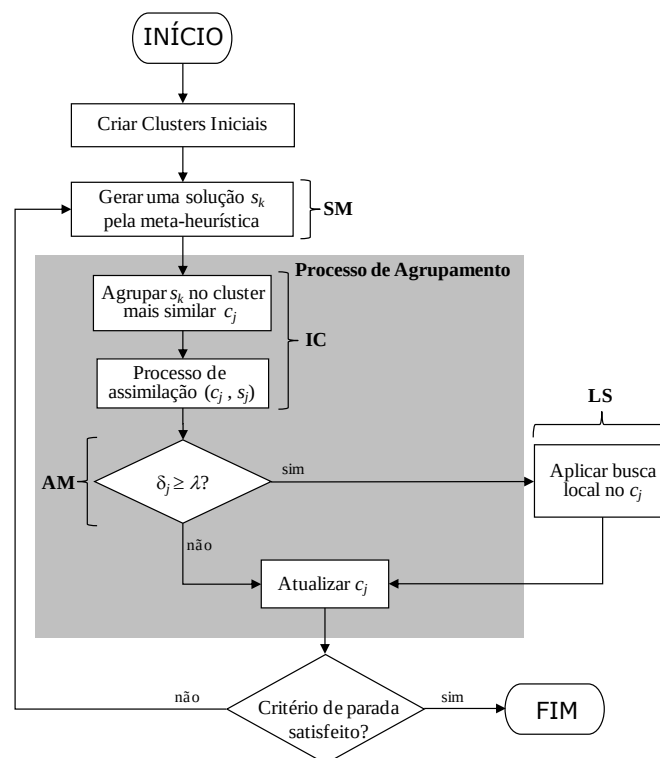


Figura 6: Componentes do CS
Fonte: Oliveira *et al* (2013)

O componente SM pode ser implementado por qualquer algoritmo de otimização que gera soluções diversificadas do espaço de busca. Ele trabalha com um gerador de soluções, explorando o espaço de busca através de uma manipulação de um conjunto de soluções, de acordo com sua estratégia de busca específica.

O componente IC procura reunir soluções similares dentro de grupos, mantendo um centro de cluster representativo delas. Uma métrica de distância, Δ , é definida, a priori, permitindo uma medida de similaridade para o processo de agrupamento. Por exemplo, em otimização combinatorial, a similaridade pode ser definida como o número de movimentos necessários para alterar uma solução até o centro do *cluster* (Oliveira *et al*, 2013). O processo de assimilação é aplicado no centro mais próximo c_i , considerando a nova solução gerada s_k . O



método *Path-relinking* (Glover *et al*, 2000) pode ser usado para gerar uma série de pontos, memorizando o melhor ponto avaliado para ser o novo centro.

O componente AM examina cada cluster, em intervalos regulares, indicando um provável *cluster* promissor. Uma densidade de cluster, δ_j , é uma medida que indica o nível de atividade dentro do cluster j . Para simplificar, δ_j pode contar o número de soluções geradas pelo SM e agrupada em c_j . δ_j obtém um número exato *threshold*, λ , que é um valor que atingido pela contagem em δ_j , sinaliza que o cluster pode então ser melhor investigado para acelerar o processo de convergência.

Por último, o componente LS é um módulo de pesquisa interno que provém a exploração de uma região supostamente promissora, representada por um *cluster*, intensificando a busca nessa região.

5. Clustering Search aplicado ao PFCLP

Esta seção tem como objetivo explicitar como é realizada a aplicação do CS ao PFCLP.

O objetivo principal da aplicação do método para o PFCLP é tentar encontrar uma solução que procure minimizar a quantidade de conflitos entre rótulos e também busque aumentar a distância entre rótulos em conflito. Na função objetivo do método, em cada nova solução, conta-se o número de rótulos livres e as distâncias entre rótulos em conflito são usadas como penalidade na função objetivo. Assim, quanto menor a distância entre rótulos em conflito, maior a penalidade atribuída à função objetivo.

Como representação de uma solução ao PFCLP é usado um vetor de rótulos. Cada posição do vetor representa um determinado ponto de um determinado diagrama gráfico. Assim, se um diagrama possui, por exemplo, 50 pontos para serem rotulados, o tamanho do vetor que representa uma solução para seu contexto de problema é de 50. Cada posição do vetor recebe o valor do rótulo escolhido para representar aquele ponto. O valor dos rótulos entre 1 e 4, então cada posição i do vetor pode receber um valor entre 1 e 4. A Figura 7 mostra um exemplo do vetor de rótulos usado como representação de solução.

1	2	3	4	5	6	7
3	4	1	4	2	3	1

Figura 7 – Representação de solução para o PFCLP

O vetor de solução do PFCLP é usado junto ao método CS. Neste trabalho é usada a metaheurística *Simulated Annealing* (SA) como geradora de soluções para o *Clustering Search*.

São usadas três estruturas de vizinhança para o SA.

- Na primeira estrutura do SA é escolhida uma posição aleatória do vetor de rótulos e um rótulo aleatório para substituir aquele que está na posição escolhida. A Figura 8a mostra um exemplo de aplicação da estrutura. Nela, é escolhida a posição 3 do vetor e o rótulo 1 é trocado pelo rótulo 3;
- Na segunda estrutura do SA são escolhidas duas posições aleatórias do vetor de rótulos e os rótulos das posições são trocados entre si. A Figura 8b mostra um exemplo de aplicação da estrutura. Nela, são escolhidas as posições 5 e 6 do vetor e os rótulos 3 e 2 são trocados entre si;
- Na terceira estrutura do SA é escolhida uma posição aleatória do vetor de rótulos e são trocados os rótulos da posição escolhida e das posições que possuem conflito com o seu rótulo. A Figura 8c mostra um exemplo de aplicação da estrutura. Nela, é escolhida a posição 3 do vetor e então observa-se que seu rótulo atual, o de número 1 possui conflito com o rótulo da posição 1 (rótulo 2) e da posição 5 (rótulo 2). Logo, são escolhidos rótulos aleatórios para essa posição e a posição 3 passa a ter o rótulo 2, a posição 1 o rótulo 3 e a posição 5 também o rótulo 3;

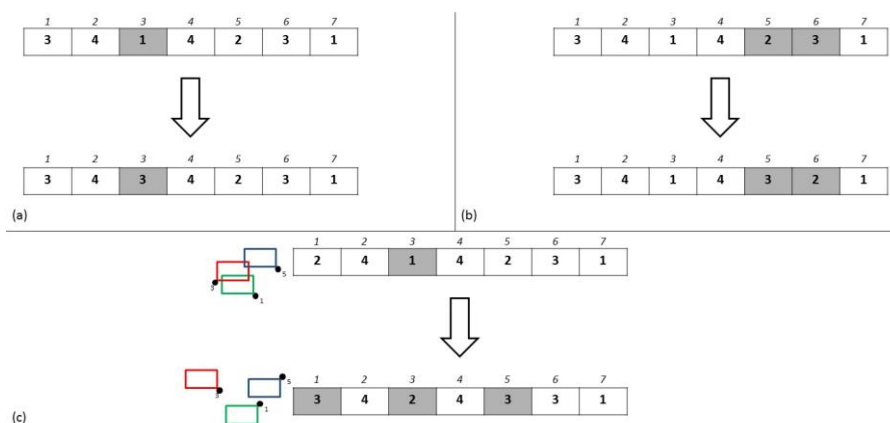


Figura 8 – Estruturas de vizinhança do SA

As soluções produzidas pelas metaheurísticas devem ser agrupadas no CS. Para tanto, o CS necessita de um critério de similaridade entre as soluções. Logo, como medida de similaridade entre soluções no PFCLP é usado o vetor de rótulos. Quanto maior o número de rótulos iguais escolhidos nos mesmos pontos do diagrama gráfico de duas soluções, menor será a distância entre elas.

Após a identificação do *cluster* que receberá uma nova solução para ser agrupada, o método *Path-relinking* é usado para assimilar a nova solução. Ele gera soluções inserindo gradualmente características diferentes da solução agrupada na solução central. Nesse caso as características são os rótulos diferentes encontrados nas mesmas posições dos dois vetores de soluções. A cada característica inserida, uma nova solução é produzida. A melhor solução observada na aplicação do método passa a ser o novo centro. Após, a busca local é aplicada sobre o centro dos *cluster* se ele atingir um certo número de soluções agrupadas. Esse número limitante tem por finalidade identificar se um *cluster* é representante de uma região promissora.

Como componente de Busca Local do CS é utilizado o método VND (Hansen e Mladenovic, 1997). O VND faz uso de diferentes estruturas de busca local. Previamente, é definida uma ordem de execução dessas buscas, de modo que quando uma estrutura não apresentar uma melhora sobre a solução corrente, o método executa a próxima busca. Se uma melhora é encontrada, o método reinicia a aplicação das buscas locais sobre a nova solução retornando a primeira estrutura.

As estruturas de vizinhança do VND para o PFCLP seguem a mesma lógica das três estruturas do SA, tornando-as em buscas locais. Em cada uma delas, não são mais escolhidos pontos aleatórios do vetor de rótulos, mas todos os pontos são percorridos e todas as possibilidades de trocas são realizadas. Nesta abordagem, optou-se pela opção *First improvement* em todas as estruturas.

6. Experimentos computacionais

Esta seção apresenta os resultados dos testes computacionais do modelo MD utilizando o *solver* CPLEX 12.1 (IBM, 2009) e também do método CS. Nos testes foram utilizados conjuntos de 505, 5056 e 13188 pontos, considerando quatro posições candidatas para cada ponto, de acordo com a uniformização proposta por Christensen *et al.* (1995). Os dados usados foram obtidos de um conjunto de pontos correspondentes a cidades do mapa da Suíça encontrada em <http://mistic.heigvd.ch/taillard/problems.dir/problems.html>. Para cada conjunto de pontos há disponíveis 12 instâncias com diferentes alturas (h) e larguras (l) dos rótulos. Os testes computacionais para o modelo MD foram executados sobre um computador com processador Intel Core i7 e 4 GB de memória.

A Tabela 1 apresenta os resultados para dispersão e proporção de rótulos livres de sobreposições para as 36 instâncias testadas. A coluna *RLS(%)* apresenta a porcentagem de



rótulos livres de conflito nas soluções, sendo que 100% indica que não houve conflitos. A coluna *DMS(cm)* apresenta a menor distância encontrada entre rótulos que entram em conflito em centímetros. A coluna *Tempo(S)* apresenta o tempo computacional dos testes em segundos.

Tabela 1 – Comparação dos resultados para testes com instâncias de 505, 5046 e 13188 pontos

Instâncias	MD			CS		
	RLS (%)	DMS (cm)	Tempo (s)	RLS (%)	DMS (cm)	Tempo (s)
Instâncias de 505 pontos						
h02_l24	100,000	-	0,300	100,000	-	0,046
h02_l32	100,000	-	0,190	100,000	-	0,132
h03_l16	100,000	-	0,200	100,000	-	0,185
h02_l42	99,208	18,027	0,260	99,208	28,018	0,245
h02_l48	99,208	39,012	0,220	99,208	37,014	0,243
h03_l24	98,416	14,035	12,750	98,416	5,099	0,488
h04_l16	96,634	6,082	5500,000	97,030	2,236	2,123
h03_l28	97,030	18,027	2790,890	97,228	2,236	3,580
h04_l18	95,248	0,000	5500,000	96,040	2,236	4,980
h03_l32	95,446	13,152	5500,000	95,842	2,236	4,643
h04_l21	92,871	10,198	5500,000	94,653	2,236	7,622
h04_l24	89,703	10,049	5500,000	92,079	3,162	8,770
<i>Média</i>	96,980	14,287	2525,401	97,475	9,386	2,755
Instâncias de 5046 pontos						
h02_l24	99,445	10,049	4,450	87,872	1,414	20,000
h02_l32	99,108	17,029	13,800	83,868	1,414	20,000
h03_l16	98,098	7,281	5500,000	94,332	1,414	30,000
h02_l42	98,434	18,027	5500,000	92,152	1,414	30,000
h02_l48	97,899	13,038	5500,000	90,428	1,414	30,000
h03_l24	95,145	6,082	5500,000	87,535	1,414	40,960
h04_l16	93,480	2,828	5500,000	86,920	1,414	45,000
h03_l28	93,480	9,055	5500,000	85,117	1,414	50,000
h04_l18	91,102	3,162	5500,000	83,987	1,414	60,000
h03_l32	91,082	11,045	5500,000	82,402	1,414	56,000
h04_l21	88,327	4,123	5500,000	80,955	1,414	61,000
h04_l24	84,443	3,605	5500,000	77,091	1,414	71,000
<i>Média</i>	94,170	8,777	4584,854	86,055	1,414	42,830
Instâncias de 13188 pontos						
h02_l24	-	-	-	93,729	1,414	580,906
h02_l32	-	-	-	91,818	1,414	601,186
h03_l16	-	-	-	89,839	1,414	703,090
h02_l42	-	-	-	86,874	1,414	701,007
h02_l48	-	-	-	86,275	1,414	863,196
h03_l24	-	-	-	81,931	1,414	951,112
h04_l16	-	-	-	79,678	1,414	952,197
h03_l28	-	-	-	76,987	1,414	955,556
h04_l18	-	-	-	75,743	1,414	953,084
h03_l32	-	-	-	74,158	1,414	1221,009
h04_l21	-	-	-	71,876	1,414	1221,015
h04_l24	-	-	-	67,433	1,414	1218,359
<i>Média</i>	-	-	-	81,362	1,414	910,143



No contexto das instâncias de 505 pontos, nota-se que formulação MD e o CS apresentam resultados próximos em termos de porcentagem de rótulos livres. Contudo, quando as instâncias passam a apresentar problemas de soluções mais difíceis de se encontrar, como por exemplo a partir da instância h04_l16, o modelo passa a ter dificuldades em encontrar sua solução e seu tempo computacional se torna inviável. Assim, o tempo foi limitado a 5500 segundos. A partir dessa instância, há a necessidade de uso de uma metaheurística que encontre boas soluções para o problema. Logo foi aplicado o CS. O método prioriza otimizar os rótulos livres e como critério de desempate, usa a distância para estabelecer qual solução é superior a outra. Assim, a partir da instância h04_l16, o CS passa a superar os resultados do modelo MD em termos de rótulos livres. Sobre a instância h03_l28, ela encerra seu processo antes de 5500 segundos, contudo esperava-se que a formulação não encontrasse o número ótimo de rótulos livres, pois sua função objetivo procura otimizar rótulos livres com a distância, o que faz com que haja a possibilidade de não encontrar o número ótimo de rótulos livres, mas sim um balanceamento da dispersão de rótulos.

Sobre o contexto das instâncias de 5046 pontos, percebe-se que o modelo MD passa então a apresentar soluções melhores que o CS. Contudo, novamente o tempo computacional se tornou inviável em problemas de maior complexidade, e necessitou novamente ser limitado. O CS não supera os resultados em termos de rótulos livres e distância, mas apresenta resultados próximos em tempo mais aceitável que o modelo. Por exemplo, na instância h02_l42, o modelo apresentou 98,434% de rótulos livres em 5500 segundos enquanto o CS apresentou 92,152% de rótulos livres, número próximo ao modelo, em 30 segundos. Portanto, o CS ainda se mostra competitivo, podendo até mesmo superar os resultados do modelo com alguns ajustes, como mudança em seu método de agrupamento, para consumir menos tempo computacional e aumentar o uso de busca local ou também uso de outra metaheurística geradora de soluções. Uma observação é que o valor 1,414 se repete nos valores de distância. Esses resultados foram conferidos e apesar de se repetirem, estão corretos. Isso se deve ao fato de o CS priorizar o número de rótulos livres sobre a distância, o que provoca com que enquanto ele ainda possa continuar a encontrar bons valores de quantidade de rótulos sem conflitos, a distância pode não ser otimizada e um valor pequeno ser apresentado no resultado final.

As instâncias de 13188 pontos apresentam limitações para o modelo, que não consegue encontrar soluções para esse contexto em tempo e uso de memória adequados. Portanto, somente os resultados do CS são apresentados. Nota-se que o CS encontrou bons resultados nas instâncias de problemas mais simples, porém a medida que se cresce a complexidade do contexto, há maior dificuldades em se encontrar boas soluções. Por exemplo, enquanto na instância h02_l24 a porcentagem de rótulos livres chega a 93,729%, na instância h04_l24, a porcentagem chega a 67,433%. Outro problema é o tempo computacional que aumenta junto a dificuldade do problema, fazendo com que o CS necessite de um certo gasto de tempo considerável para se encontrar boas soluções. Contudo, ainda consegue entregar boas soluções em um problema que em menores instâncias, com menos pontos, o modelo obteve dificuldades em encontra-las.

7. Conclusão

Este trabalho aplicou o método híbrido *Clustering Search* (CS) e fez nova aplicação do modelo matemático MD ao Problema de Localização de Rótulos Cartográficos (PFCLP, do inglês *Point Feature Cartographic Label Placement*) visando melhorar a legibilidade dos mapas. Esse método utiliza o conceito de regiões promissora, buscando identifica-las para que a aplicação de busca local seja feita de forma mais inteligente sobre elas. O CS e o modelo MD então tem o objetivo de maximizar o número de rótulos sem conflitos e também maximizar a distância entre rótulos em conflitos.

O método heurístico encontrou bons resultados em instâncias encontradas na literatura. Foram feitos testes em instâncias de diferentes tamanhos e contextos de problemas, com 505, 5046 e 13188 pontos. Nas instâncias de 505 pontos, o método CS encontrou bons resultados, chegando mesmo a encontrar soluções melhores que o modelo MD, enquanto que nas de 5046 pontos, o CS apresentou soluções inferiores ao MD, porém com tempo computacional mais competitivo. No contexto de 13188 pontos, não foi possível aplicar o modelo devido a limitações



de tempo e memória computacional, contudo o método CS conseguiu encontrar boas soluções em tempo razoável. Houve uma repetição de resultados nos valores de distância entre as instâncias de 5046 e 13188 pontos devido ao CS priorizar o número de rótulos livres sobre a distância, o que faz o método se atentar na otimização de um objetivo, deixando o outro como um critério de desempate entre soluções com mesmo valor de rótulos livres.

Como trabalhos futuros pretende-se fazer novos estudos sobre as mudanças no método de agrupamento do CS, para que ele melhore sua eficiência. Pretende-se também explorar alternativas para que ele venha a se tornar autoconfigurável para evitar problemas na escolha de seus parâmetros. Sugere-se também que o PFCLP seja tratado em contexto de problema multiobjetivo, tendo os rótulos livres e distância mínima como objetivos a serem otimizados ao mesmo tempo.

Agradecimentos

Este trabalho foi financiado pela Fundação de Amparo à Pesquisa do Estado de São Paulo - FAPESP (Processo 2014/00580-0 e 2016/07135-7) e pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (Processo nº 482170/2013-1 e 301836/2014-0).

Referências bibliográficas

- AGARWAL, P. K., KREVELD, M. V., SURI, S. (1998). **Label placement by maximum independent set in rectangles**. *Computational Geometry: Theory and Applications*, 11, 209–218.
- ALVIM, A. C. F., TAILLARD, E. D. (2009). **POPMUSIC for the point feature label placement problem**. *European Journal of Operational Research*, 192(2), 396–413.
- ARAÚJO, E. J.; CHAVES, A. A.; LORENA, L. A. N. (2015) **Novo modelo de dispersão para o problema de localização de rótulos cartográficos**. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL (SBPO), Porto de Galinhas, PE. *Anais...*2015.
- CHRISTENSEN, J., MARKS, J., SHIEBER, S. (1995). **An empirical study of algorithms point-features label placement**. *ACM Transactions on Graphics*, 14(3), 203–232.
- CRAVO, G. L., RIBEIRO, G. M., LORENA, L. A. N. (2008). **A greedy randomized adaptive search procedure for the point-feature cartographic label placement**. *Computers & Geosciences*, 34(4), 373–386.
- GLOVER, F., LAGUNA, M., MARTÍ, R. (2000). **Fundamentals of scatter search and path relinking**. *Control and cybernetics*, 29(3), 653-684.
- GOMES, S. P., RIBEIRO, G. M., LORENA, L. A. N. (2013). **Dispersion for the point-feature cartographic label placement problem**. *Expert Systems with Applications*, 40(15), 5878-5883.
- GOMES, S. P., LORENA, L. A. N., RIBERO, G. M. (2016). **A constructive genetic algorithm for discrete dispersion on point feature cartographic label placement problems**. *Geographical Analysis*, 48(1), 43-58.
- HANSEN, P., MLADENOVIC, N. (1997). **Variable neighborhood search**. *Computers & operations research*, 24(11), 1097-1100.
- IBM (2009). IBM ILOG CPLEX Optimizer 12.1, New York, <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/> [Accessed June 20, 2012].
- OLIVEIRA, A. C. M.; CHAVES, A. A.; LORENA, L.A.N. (2013), **Clustering Search**. *Pesquisa operacional*, v.33 (1), 105-121.



- RABELLO, R. L., MAURI, G. R., RIBEIRO, G. M., 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, G.M. (2007). **Relaxação Lagrangeana com divisão em clusters para alguns problemas de otimização modelados em grafos de conflitos.** Ph.D. Dissertation Instituto Nacional de Pesquisas Espaciais. 63, 191.
- RIBEIRO, G. M., LORENA, L. A. N. (2006). **Heuristics for cartographic label placement problems.** *Computers & Geosciences*, 32(6), 739–748.
- RIBEIRO, G. M., LORENA, L. A. N. (2008). **Lagrangean relaxation with clusters for point-feature cartographic label placement problems.** *Computers & Operations Research*, 35(7), 2129–2140.
- RIBEIRO, G. M., MAURI, G. R., LORENA, L. A. N. (2011). **A Lagrangean decomposition for the maximum independent set problem applied to map labeling.** *Operational Research*, 11(3), 229–243.
- STRIJK, T., VERWEIJ, B., AARDAL, K. (2000). **Algorithms for maximum independent set applied to map labelling.** *Department of Computer Science, Utrecht University.*
- VERWEIJ, A. M., AARDAL, K. I. (1999). **An optimization algorithm for maximum independent set with applications in map labelling.** In: *Proceedings 7th annual European symposium on algorithms* (pp. 426–437). Prague, Czech Republic.
- WAGNER, F., WOLFF, A., KAPOOR, V., STRIJK, T. (2001). **Three rules suffice for good label placement.** *Algorithmica*, 30, 334–349.
- YAMAMOTO, M., CAMARA, G., LORENA, L. A. N. (2002). **Tabu search heuristic for pointfeature cartographic label placement.** *Geoinformatica*, 6(1), 77–90.
- YAMAMOTO, M., LORENA, L. A. N. (2005). **A constructive genetic approach to pointfeature cartographic label placement.** In T. Ibaraki, K. Nonobe, & M. Yagiura (Eds.), *Metaheuristics: Progress as real problem solvers* (pp. 285–300). Kluwer Academic Publishers.
- ZORASTER, S. (1990). **The solution of large 0–1 integer programming problems encountered in automated cartography.** *Operations Research*, 38(5), 752–759.