

META-HEURÍSTICA CLUSTERING SEARCH APLICADA AO PROBLEMA DE ROTULAÇÃO CARTOGRÁFICA DE PONTOS

Rômulo Louzada Rabello

Universidade Federal do Espírito Santo – UFES
romulolouzada@gmail.com

Geraldo Regis Mauri

Universidade Federal do Espírito Santo – UFES
mauri@cca.ufes.br

Glaydston Mattos Ribeiro

Universidade Federal do Rio de Janeiro – UFRJ
glaydston@pet.coppe.ufrj.br

Luiz Antonio Nogueira Lorena

Instituto Nacional de Pesquisas Espaciais – INPE
lorena@lac.inpe.br

RESUMO

O Problema de Rotulação Cartográfica de Pontos (PRCP) consiste em atribuir rótulos (textos) para pontos de interesse em um mapa, sem ocasionar sobreposição, a fim de proporcionar uma melhor visualização dos mesmos. Este trabalho apresenta uma aplicação da meta-heurística *Clustering Search* (CS) como uma nova alternativa para resolução do PRCP. Experimentos computacionais foram realizados utilizando conjuntos de instâncias com 25 a 13206 pontos a serem rotulados, sendo estas instâncias as mesmas consideradas em diversos trabalhos a respeito do PRCP. Os resultados obtidos demonstram o potencial da CS uma vez que as soluções ótimas (comprovadas em outros trabalhos da literatura) sempre são alcançadas e, para as instâncias cujas soluções ótimas não são conhecidas, a CS apresenta soluções melhores do que as conhecidas até então. A CS foi planejada e implementada pelo estudante sob supervisão dos professores orientadores.

PALAVRAS CHAVE. *Clustering Search*, Problema de Rotulação Cartográfica de Pontos, Meta-Heurísticas.

Área principal: MH - Meta-heurísticas.

ABSTRACT

The Point-Feature Cartographic Label Placement Problem (PFCLPP) consists of placing text labels to point features on a map avoiding overlaps to improve the map visualization. This paper presents an application of the *Clustering Search* (CS) metaheuristic as a new alternative to solve the PFCLPP. Computational experiments were performed over sets of instances with 25 up to 13206 points to be labeled. These instances are the same used in several recent and important researches about the PFCLPP. The results enhance the CS potential by finding optimal solutions (proved in previous works) and improving the best-known solutions for instances whose optimal solutions are unknown so far. The CS was developed by the student, under supervision of the advisors.

KEYWORDS. *Clustering Search*. Point-Feature Cartographic Label Placement Problem. Metaheuristics.

Main area: MH - Metaheuristics.

1. Introdução

O PRCP é classificado na literatura como *NP-Hard* (MARKS; SHIEBER, 1991) e, segundo Ribeiro e Lorena (2008b), consiste em organizar pontos em mapas de modo que não ocorra sobreposição de rótulos, proporcionando assim clareza na visualização e entendimento do mesmo.

A Figura 1 apresenta um exemplo de um mapa cartográfico com sobreposições. As regiões indicadas (setas) apontam locais com grande concentração de rótulos (sobreposições), o que inviabiliza a legibilidade dos mesmos.

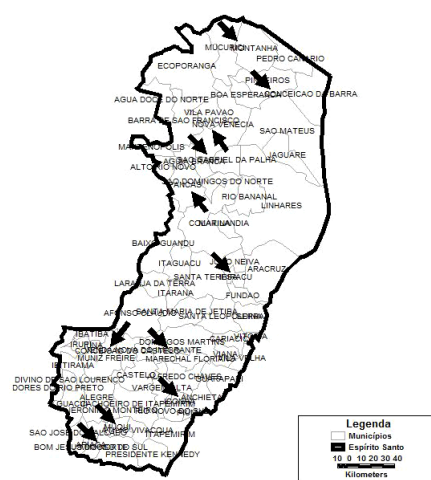


Figura 1 - Exemplo de mapa com sobreposição de rótulos (RIBEIRO; LORENA, 2008b).

Uma forma de melhorar a visualização do mapa é deslocando os rótulos de modo que não haja mais sobreposições, observando que os mesmos devem estar sempre perto de seus respectivos pontos. Diante disso, é apresentado o conceito de posições candidatas de um ponto. As posições candidatas representam os possíveis locais nos quais o rótulo de determinado ponto pode ser inserido, respeitando sempre uma padronização cartográfica.

A Figura 2 apresenta a padronização proposta por Christensen et al. (1995), que é uma das mais conhecidas e utilizadas na literatura. As regiões de 1 a 8 representam as posições candidatas para o posicionamento do rótulo relativo ao ponto cartográfico, sendo que quanto menor o número da região, maior é sua preferência cartográfica (OLIVEIRA et al., 2010).

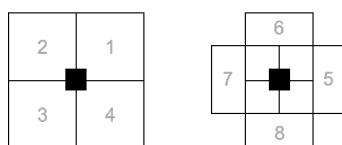


Figura 2 - Padronização proposta por Christensen et al. (1995).

No PRCP, as sobreposições de rótulos podem ser aceitas ou não. Caso as sobreposições não sejam permitidas, deve-se rotular o maior número possível de pontos, sendo que alguns pontos poderão ficar sem rótulos. Nesse caso, o PRCP pode ser visto como um Problema de Máximo Conjunto Independente (RIBEIRO et al., 2011). A abordagem escolhida para estudo neste trabalho foi a de aceitar sobreposições. Nesse caso, o objetivo é maximizar o número de rótulos “livres” (ALVIM; TAILLARD, 2009; MAURI et al., 2010). Uma formulação matemática para essa abordagem é apresentada em Mauri et al. (2010).

Este trabalho tem como objetivo apresentar uma nova alternativa para resolver o PRCP. É proposta uma aplicação da meta-heurística conhecida como *Clustering Search* (CS), utilizando um *Simulated Annealing* como parte de sua estrutura para gerar soluções ao longo do processo. A

escolha da CS é baseada no fato de a mesma vir apresentando bons resultados para diferentes tipos de problemas combinatoriais, além de ainda não ter sido utilizada para resolução do PRCP.

Os resultados da CS são comparados com os obtidos por diversos métodos apresentados em trabalhos recentes encontrados na literatura e, assim, pode-se perceber o seu desempenho, por meio de soluções iguais ou melhores para praticamente todas as instâncias consideradas.

O restante do artigo está organizado da seguinte forma: a Seção 2 apresenta uma breve revisão bibliográfica sobre o PRCP; a CS é apresentada em detalhes na Seção 3, e os experimentos computacionais são expostos na Seção 4; por fim, as conclusões são apresentadas na Seção 5.

2. Revisão Bibliográfica

Na literatura, há diversos trabalhos relacionados ao Problema de Rotulação Cartográfica de Pontos (PRCP) e, nessa seção, são apresentados brevemente alguns entre os mais recentes.

Alvim e Taillard (2009) utilizaram a meta-heurística POPMUSIC (*Partial Optimization Metaheuristic under Special Intensification Conditions*) para resolver o PRCP. A POPMUSIC utilizada divide o problema em subproblemas e aplica uma Busca Tabu. Para as instâncias com até 1000 pontos a serem rotulados, o método proposto superou os resultados conhecidos até então. Os autores ainda propuseram um conjunto de instâncias com 13206 pontos.

Mauri et al. (2010) apresentaram novas abordagens para resolução do PRCP, propondo uma nova formulação matemática e uma técnica baseada na Decomposição Lagrangiana. Os autores apresentaram melhores soluções e provaram a otimalidade, até então desconhecida, para diversas instâncias com até 1000 pontos. O modelo proposto por Mauri et al. (2010) foi utilizado por Ribeiro et al. (2009), que propuseram algumas desigualdades para fortalecer o modelo, e obtiveram melhores soluções para as instâncias com 1000 pontos.

Por fim, Oliveira et al. (2009,2010) apresentaram uma heurística de *Backtracking* e uma *Iterated Local Search* (ILS), as quais obtiveram os melhores resultados para as instâncias de 1000 e 13206 pontos.

3. Clustering Search

O *Clustering Search* (CS) é uma meta-heurística híbrida proposta por Oliveira e Lorena (2007) que emprega agrupamentos para detectar regiões promissoras do espaço de busca. Na CS, as soluções são agrupadas em *clusters* e cada nova solução é associada (direcionada) ao “melhor” *cluster* de acordo com uma métrica de distância. Um *cluster* C_i é definido por meio de uma tupla (c_i, v_i, r_i) , na qual c_i , v_i e r_i representam o centro, o volume e a ineficiência do *cluster* i , respectivamente. O centro c_i é uma solução que representa o *cluster* i , o volume v_i define o número de soluções associadas ao *cluster* i e r_i controla a estratégia de busca, indicando o número de vezes consecutivas que uma busca local foi aplicada no *cluster* i e não melhorou a principal solução do *cluster*, ou seja, o seu centro.

Neste trabalho, a CS implementada (Figura 3) foi baseada na proposta por Ribeiro et al. (2012), e pode ser descrita por meio de seis componentes principais: gerador de soluções, agrupamento iterativo, função objetivo, processo de assimilação, processo de análise e estratégia de busca. Esses componentes são apresentados a seguir:

3.1 Gerador de Soluções

Um *Simulated Annealing* (SA), baseado no utilizado por Ribeiro et al. (2012) é usado como gerador de soluções. A partir de uma solução inicial, gerada aleatoriamente, o SA executa pequenos movimentos na vizinhança da solução atual. A vizinhança N utilizada consiste em uma simples troca aleatória da posição candidata de um ponto qualquer, selecionado aleatoriamente.

3.2 Agrupamento Iterativo

Considerando o número de *clusters* γ , uma nova solução s , gerada pelo SA, é atribuída ao *cluster* i mais próximo, de acordo com a distância de Hamming H_i (HAMMING, 1950) que, neste trabalho, foi calculada de acordo com o número de pontos cuja posição candidata

selecionada é diferente entre as soluções s e c_i . Assim, nesse processo, a solução s é atribuída ao cluster i que minimiza a função $\arg \min_{i \in \{1, \dots, \gamma\}} \{H_i\}$.

3.3 Função Objetivo

A função objetivo utilizada na CS considera o número de rótulos “livres” (sem sobreposição), ou seja, $f(s)$ = número de rótulos livres. Neste trabalho, assim como nos principais trabalhos sobre o PRCP (utilizados para comparação), a preferência cartográfica não foi considerada.

3.4 Processo de Assimilação

Definido o *cluster* i para o qual uma solução s deverá ser assimilada, o centro desse *cluster* c_i e seu volume v_i são atualizados. Neste trabalho, foi utilizada uma “assimilação direta”, ou seja, s passará a ser o centro do *cluster* caso represente uma solução melhor do que o centro atual $\Rightarrow c_i = \text{melhor}(s, c_i)$.

3.5 Processo de Análise

O nível de atividade de cada *cluster* i é indicado pelo seu volume v_i e, quando esse volume atinge um limite λ , o *cluster* i deve ser investigado por uma estratégia de busca, pois o mesmo se tornou promissor.

3.6 Estratégia de Busca

Uma busca local é usada para intensificar a exploração do *cluster* mais promissor. Neste trabalho, a busca local é baseada na troca da posição candidata selecionada para um ponto qualquer. Esse processo é repetido até que nenhuma melhora seja obtida. Caso a ineficiência do *cluster* i (r_i) alcance o limite r_{max} , que indica o número de vezes que a busca local é aplicada sem melhoras, uma perturbação é aplicada em c_i por meio da vizinhança N .

```

1. Criar  $\gamma$  novas soluções (clusters) aleatoriamente
2.  $v_i \leftarrow 0$  e  $r_i \leftarrow 0 \ \forall i = 1, \dots, \gamma$     $s \leftarrow$  solução inicial    $s^* \leftarrow s$     $T \leftarrow T_0$ 
3. ENQUANTO  $T > T_c$  FAÇA
4.   iter  $\leftarrow 0$ 
5.   ENQUANTO iter  $< S_{A_{max}}$  FAÇA
6.     iter  $\leftarrow$  iter + 1    $s' \leftarrow N(s)$ 
7.     SE  $f(s') > f(s)$  ENTÃO
8.        $s \leftarrow s'$ 
9.     SENÃO
10.      Com probabilidade  $e^{-(f(s)-f(s'))/T}$   $s \leftarrow s'$ 
11.     FIM-SE
12.   FIM-ENQUANTO
13.    $T \leftarrow \alpha T$     $i \leftarrow \arg \min_{i \in \{1, \dots, \gamma\}} \{H_i\}$     $v_i \leftarrow v_i + 1$     $c_i \leftarrow \text{melhor}(s, c_i)$ 
14.   SE  $v_i = \lambda$  ENTÃO
15.      $v_i \leftarrow 0$     $s \leftarrow$  busca local( $c_i$ )
16.     SE  $f(s) = f(c_i)$  ENTÃO
17.        $r_i \leftarrow r_i + 1$ 
18.       SE  $r_i = r_{max}$  ENTÃO
19.          $r_i \leftarrow 0$     $c_i \leftarrow N(c_i)$ 
20.       FIM-SE
21.     SENÃO
22.        $r_i \leftarrow 0$ 
23.     FIM-SE
24.   FIM-SE
25.    $s^* \leftarrow \max(s^*, c_i)$ 
26. FIM-ENQUANTO
27. Retornar  $s^*$ 

```

Figura 3 - Pseudocódigo da CS implementada.

4. Experimentos Computacionais

Com o intuito de verificar a eficiência da *Clustering Search* (CS) proposta para resolução do PRCP, foram utilizados dois conjuntos de instâncias (*benchmarks*) reportados na literatura, que são considerados nos principais e mais recentes trabalhos sobre o problema.

O primeiro conjunto é formado por instâncias com 25, 100, 250, 500, 750 e 1000 pontos, todas com 4 posições candidatas, sendo 8 instâncias distintas com 25 pontos e 25 instâncias diferentes para 100, 250, 500 750 e 1000 pontos. Já o segundo conjunto é formado por 40 instâncias com 13206 pontos, sendo 20 com 4 posições candidatas e outras 20 com 8 posições candidatas. Assim, foi utilizado um total de 173 instâncias distintas e de tamanhos variados.

Para calibração dos parâmetros da CS foram selecionadas, aleatoriamente, uma instância de cada tamanho (no caso de 13206 pontos, foram consideradas duas instâncias: com 4 e 8 posições candidatas), resultando em um subconjunto com 8 instâncias. A partir desse subconjunto, foram realizados vários experimentos variando o valor dos parâmetros. A metodologia utilizada para determinação dos parâmetros é similar à adotada por Ribeiro et al. (2012). Os valores obtidos para os parâmetros foram: γ (número máximo de *clusters*) = 10; λ (volume máximo para aplicação da busca local) = 7; r_{max} (limite para aplicação da perturbação) = 4; T_0 (temperatura inicial para o SA) = 40000; T_C (temperatura de congelamento para o SA) = 0,01; α (taxa de resfriamento para o SA) = 0,975; e SA_{max} (número máximo de iterações para o SA) = 12000.

Após a calibração dos parâmetros, a CS foi aplicada dez vezes para cada instância. Para todas as instâncias com 25, 100, 250, 500 e 750 pontos, a CS encontrou as soluções ótimas (provadas por Mauri et al., 2010) em todas as 10 execuções (desvio = 0%), e o tempo médio de processamento foi de 15,73, 0,00, 3,44, 15,61 e 27,98 segundos, respectivamente.

Para as instâncias com 1000 pontos, o desvio ($((\text{Melhor} - \text{Média})/\text{Melhor}) * 100$) médio entre as 10 execuções para cada instância foi de 0,03%, e o tempo médio de execução foi de 94,18 segundos. Os resultados para essas instâncias são apresentados na Tabela 1. Nessa tabela, é apresentado, para cada instância, o número de rótulos livres (Rot. Livres = $f(s)$) para a melhor solução, o percentual de rótulos livres (%RL = $(\text{Rot. Livres}/n) * 100$ – sendo n o número de pontos da instância, nesse caso $n = 1000$) para a melhor solução, o desvio e o tempo médio de execução (entre as 10 execuções).

Tabela 1 - Resultados para as instâncias com 1000 pontos.

Inst	Melhor Solução		Desvio (%)	Tempo (s)	Inst	Melhor Solução		Desvio (%)	Tempo (s)
	Rot. Livres	%RL				Rot. Livres	%RL		
1	939	93,90	0,00	52,58	14	933	93,30	0,01	70,40
2	934	93,40	0,00	86,44	15	934	93,40	0,12	79,85
3	934	93,40	0,01	127,50	16	932	93,20	0,02	101,16
4	934	93,40	0,00	129,94	17	937	93,70	0,00	40,75
5	961	96,10	0,00	84,39	18	946	94,60	0,12	120,94
6	932	93,20	0,00	86,17	19	950	95,00	0,11	114,15
7	929	92,90	0,04	147,85	20	934	93,40	0,01	75,79
8	940	94,00	0,00	79,76	21	930	93,00	0,09	119,05
9	927	92,70	0,01	94,56	22	952	95,20	0,04	97,31
10	944	94,40	0,00	82,37	23	934	93,40	0,00	71,31
11	947	94,70	0,00	50,35	24	932	93,20	0,03	123,78
12	935	93,50	0,04	141,03	25	945	94,50	0,00	92,55
13	955	95,50	0,04	84,62					
Médias									
Rot. Livres: 938,80		%RL: 93,88		Desvio: 0,03		Tempo (s): 94,18			

Para as instâncias com 1000 pontos, as melhores soluções encontradas na literatura, até então, são aquelas obtidas pela heurística de *Backtracking* proposta por Oliveira et al. (2010), que por sua vez só apresentam a média do percentual de rótulos livres (%RL), ou seja, os autores não apresentam as soluções individuais para cada instância.

A Tabela 2 apresenta uma comparação com a literatura dos resultados obtidos pela CS proposta para as instâncias com 250, 500, 750 e 1000 pontos. Os resultados apresentados nessa tabela foram obtidos por meio das médias dos percentuais de rótulos livres (%RL) para cada “tamanho” de instância. Os melhores resultados são destacados em negrito. A partir de uma análise dessa tabela, pode-se perceber que, para as instâncias com 250, 500 e 750 pontos, os resultados obtidos pela CS foram idênticos aos melhores resultados apresentados na literatura (soluções ótimas - já comprovadas). Já para as instâncias com 1000 pontos, a CS apresentou uma melhora de 0,01% de rótulos livres em relação ao melhor resultado da literatura.

Tabela 2 - Comparação dos resultados obtidos com a literatura (250, 500, 750 e 1000 pontos).

Método de solução	Número de pontos - <i>n</i>			
	250	500	750	1000
Clustering Search (CS) proposta	100,00	99,68	97,96	93,88
<i>Backtracking</i> (OLIVEIRA et al., 2010)	100,00	99,68	97,96	93,87
ILS (OLIVEIRA et al., 2009)	100,00	99,68	97,95	93,76
Modelo C2 (RIBEIRO et al., 2009)	100,00*	99,68*	97,96	93,86
Modelo C1 (RIBEIRO et al., 2009)	100,00*	99,68*	97,96	93,84
Cplex com formulação forte (MAURI et al., 2010)	100,00*	99,68*	97,96	93,84
Decomposição Lagrangiana (MAURI et al., 2010)	100,00*	99,68*	97,96*	93,74
Cplex com formulação simples (MAURI et al., 2010)	100,00*	99,68*	97,91	90,13
Pop(asc) (ALVIN; TAILLARD, 2009)	100,00	99,67	97,72	92,68
Pop(10) (ALVIN; TAILLARD, 2009)	100,00	99,67	97,46	91,94
Pop(30) (ALVIN; TAILLARD, 2009)	100,00	99,67	97,72	92,54
Pop(70) (ALVIN; TAILLARD, 2009)	100,00	99,67	97,73	92,58
Pop(asc) (ALVIN; TAILLARD, 2009)	100,00	99,67	97,72	92,68
Tabu(50n) (ALVIN; TAILLARD, 2009)	100,00	99,57	97,53	91,54
Tabu(100n) (ALVIN; TAILLARD, 2009)	100,00	99,57	97,54	91,54
Tabu(500n) (ALVIN; TAILLARD, 2009)	100,00	99,57	97,55	91,59
Geração de Colunas (RIBEIRO; LORENA, 2008a)	100,00*	99,67	97,67	92,40
LagClus (RIBEIRO; LORENA, 2008b)	100,00*	99,67	97,65	91,42

*Soluções obtidas por métodos que **comprovaram** a otimalidade para todas as instâncias.

Tabela 3 - Resultados para as instâncias com 13206 pontos.

Inst.	4 posições candidatas				8 posições candidatas			
	Melhor Solução		Desvio (%)	Tempo (s)	Melhor Solução		Desvio (%)	Tempo (s)
	Rot. Livres	%RL			Rot. Livres	%RL		
1	12479	94,49	0,08	745,49	12911	97,77	0,09	689,21
2	12104	91,66	0,09	622,20	12462	94,37	0,11	1310,75
3	11893	90,06	0,10	590,51	12496	94,62	0,12	1079,20
4	11702	88,61	0,08	1168,21	12345	93,48	0,13	1260,45
5	11754	89,00	0,10	787,37	12451	94,28	0,13	995,25
6	10887	82,44	0,18	944,27	11688	88,51	0,24	1535,81
7	10491	79,44	0,10	1234,69	11330	85,79	0,10	1661,03
8	10817	81,91	0,12	1109,20	11641	88,15	0,13	1784,80
9	10806	81,83	0,10	1404,79	11318	85,70	0,20	1343,55
10	10397	78,73	0,13	1787,64	11257	85,24	0,20	1587,48
11	10062	76,19	0,09	1692,21	10948	82,90	0,17	1690,55
12	9955	75,38	0,25	1383,71	10717	81,15	0,16	2038,04
13	10787	81,68	0,12	1070,82	11694	88,55	0,23	1305,68
14	10120	76,63	0,16	1918,69	10669	80,79	0,19	2064,86
15	9695	73,41	0,14	1511,97	10576	80,08	0,25	2077,44
16	9306	70,47	0,21	1987,13	10231	77,47	0,23	1796,65
17	10255	77,65	0,17	1391,97	11246	85,16	0,28	2199,06
18	9509	72,01	0,20	1115,01	10082	76,34	0,23	2032,38
19	9074	68,71	0,32	1301,86	10031	75,96	0,24	1797,77
20	8595	65,08	0,13	1984,39	9599	72,69	0,42	1761,00
Média	10534,40	79,77	0,14	1287,61	11284,60	85,45	0,19	1600,55

Na Tabela 3 são apresentados os resultados para as instâncias com 13206 pontos, considerando 4 e 8 posições candidatas. O desvio médio e o tempo médio de execução entre as 10 execuções foi de 0,14% e 0,19% e 1287,61 e 1600,55 segundos, respectivamente.

Uma comparação entre o percentual de rótulos livres (%RL) das melhores soluções obtidas pela CS com as melhores soluções apresentadas na literatura é apresentada na Tabela 4. Para as instâncias de 13206 pontos com 4 posições candidatas, a CS encontrou 19 soluções melhores que as apresentadas por Oliveira et al. (2009), ou seja, para apenas uma instância a CS não apresentou a melhor solução. Já em relação aos métodos propostos por Alvim e Taillard (2009), todas as soluções apresentadas pela CS foram melhores. Assim, pode-se notar que, em relação às melhores soluções conhecidas até então, a CS apresentou um aumento médio (por instância) de aproximadamente 344 pontos rotulados sem sobreposição. Já para as instâncias de 13206 pontos com 8 posições candidatas, a CS apresentou soluções melhores para todas as instâncias e, em relação às melhores soluções conhecidas até então, o aumento médio (por instância) foi de aproximadamente 474 pontos rotulados sem sobreposição. Oliveira et al. (2009) não apresentam resultados para essas instâncias.

Tabela 4 - Comparação dos resultados obtidos com a literatura (13206 pontos).

Inst.	4 posições candidatas					8 posições candidatas			
	Alvin e Taillard (2009)			Oliveira et al. (2009)	CS proposto	Alvin e Taillard (2009)			CS proposto
	Tabu(50n)	Pop(asc)	Pop(10)			Tabu(50n)	Pop(asc)	Pop(10)	
1	92,20	93,10	92,55	94,65	94,49	97,13	97,61	97,22	97,77
2	88,29	89,18	88,73	91,05	91,66	92,54	93,43	92,70	94,37
3	86,12	87,12	86,69	88,91	90,06	92,87	94,09	93,06	94,62
4	83,80	84,87	84,42	86,43	88,61	91,40	92,38	91,54	93,48
5	84,25	85,26	84,73	88,47	89,00	92,44	93,47	92,51	94,28
6	74,31	75,92	75,49	79,89	82,44	84,56	85,85	84,96	88,51
7	69,23	71,76	71,16	76,34	79,44	80,54	82,23	81,35	85,79
8	73,21	75,15	74,94	79,68	81,91	83,95	85,39	84,26	88,15
9	73,00	74,81	74,28	79,47	81,83	80,57	81,84	81,07	85,70
10	67,47	70,33	69,96	75,97	78,73	79,84	81,29	80,61	85,24
11	63,56	67,11	66,66	72,85	76,19	76,58	78,45	77,46	82,90
12	62,01	65,62	65,10	72,17	75,38	74,36	76,00	75,17	81,15
13	73,19	74,54	74,15	79,48	81,68	84,55	85,70	84,48	88,55
14	64,44	67,34	67,12	73,51	76,63	73,63	75,09	74,58	80,79
15	61,95	63,03	62,86	70,22	73,41	72,57	74,20	73,29	80,08
16	58,23	59,21	58,87	66,27	70,47	68,35	70,82	70,04	77,47
17	67,07	68,82	68,17	74,75	77,65	79,22	80,70	79,52	85,16
18	60,02	60,87	60,44	68,22	72,01	66,96	69,07	68,13	76,34
19	56,10	56,75	56,55	64,32	68,71	65,53	68,89	67,86	75,96
20	51,89	52,23	51,98	60,66	65,08	58,98	64,52	63,66	72,69
Média	70,52	72,15	71,74	77,17	79,77	79,83	81,55	80,67	85,45

A CS foi implementada em C++ e todos os experimentos foram realizados em um PC com processador Intel Core i5 de 2.66GHz com 4GB de memória RAM. Oliveira et al. (2009,2010) utilizaram um PC com processador Intel Pentium IV de 3.0GHz com 1GB de memória RAM, com um tempo limite de execução de 120 segundos por instância. Os autores não apresentam o desvio médio obtido nas soluções, e consideram soluções iniciais já “otimizadas” por uma heurística construtiva, cujo tempo de execução não é apresentado. Já Alvin e Taillard (2009) utilizaram um PC com processador Intel Xeon de 3.2GHz com 6GB de memória RAM. Os métodos Tabu(50n), Pop(asc) e Pop(10) apresentaram, respectivamente, um tempo médio de 1466,21, 437,74 e 7,63 segundos para as instâncias de 13206 pontos com 4 posições candidatas, e 1655,90, 838,39 e 21,14 segundos para as instâncias com 8 posições candidatas.

5. Conclusões

Pode-se observar neste trabalho que a aplicação da meta-heurística híbrida *Clustering Search* (CS) para resolução do Problema de Rotulação Cartográfica de Pontos (PRCP) foi

eficiente, apresentando soluções idênticas às obtidas pelos melhores métodos encontrados na literatura (todas as soluções ótimas, em todos os experimentos) para as instâncias com 25, 100, 250, 500 e 750 pontos. Já para as instâncias com 1000 e 13206 pontos, os resultados obtidos pela CS foram superiores aos conhecidos até então na literatura.

Com a aplicação da CS, foi possível encontrar uma melhora significativa nas soluções para as instâncias com 1000 e 13206 pontos. Isto é, com a aplicação da CS, pode-se obter um maior número de rótulos sem sobreposições (rótulos livres), o que resulta em um mapa com uma visualização mais clara dos pontos e seus respectivos rótulos.

Assim, a CS demonstrou ser uma nova alternativa para resolução do PRCP, apresentado um bom desempenho quando comparado a métodos propostos nos principais e mais recentes trabalhos sobre o problema.

Agradecimentos

Os autores agradecem ao CNPq (processos 300692/2009-9, 300747/2010-1, 477148/2011-5, 307002/2011-0 e 476862/2012-4) e à FAPES (processos 54689473/2011 e 59830042/2012) pelo apoio financeiro.

Referências

- Alvim, A. C. F. e Taillard, E. D.** (2009), Popmusic for the point-feature label placement problem, *European Journal of Operational Research*, 192, 396-413.
- 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.
- Hamming, R. W.** (1950), Error detecting and error correcting codes, *Bell System Technical Journal*, 26, 147-160.
- Marks, J. e Shieber, S.**, The computational complexity of cartographic label placement, *Relatório Técnico*, Harvard University, 1991 (<http://www.eecs.harvard.edu/~Shieber/papers/label.ps.gz>), 2012.
- 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 & Operations Research*, 37, 2164-2172.
- Oliveira, A. C. M. e Lorena, L. A. N.** (2007), Hybrid evolutionary algorithms and clustering search, *Hybrid Evolutionary Algorithms, Studies in Computational Intelligence*, 75, 77-99.
- Oliveira, C., Urrutia, S. A. e Noronha, T. F.** (2010), Heurística de backtracking para o problema da rotulação cartográfica de pontos, *Anais do XLII SBPO - Simpósio Brasileiro de Pesquisa Operacional*.
- Oliveira, C., Urrutia, S. A. e Noronha, T. F.** (2009), Heurística ILS para o problema da rotulação cartográfica de pontos, *Anais do XII SPOLM - Simpósio de Pesquisa Operacional e Logística da Marinha*.
- 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, *Anais do XLI SBPO - Simpósio Brasileiro de Pesquisa Operacional*.
- 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, G. M., Laporte, G. e Mauri, G. R.** (2012), A comparison of three metaheuristics for the workover rig routing problem, *European Journal of Operational Research*, 220, 28-36.
- Ribeiro, G. M., Mauri, G. R. e Lorena, L. A. N.** (2011), A lagrangean decomposition for the maximum independent set problem applied to map labeling, *Operational Research: An International Journal*, 11, 229-243.