Uma heurística híbrida para detecção de comunidades com sobreposição

Chagas, G. O., Lorena, L. A. N., Santos, R. D. C.

Programa de Pós-Graduação em Computação Aplicada - CAP
Instituto Nacional de Pesquisas Espaciais - INPE
Avenida dos Astronautas, 1758 - Jardim da Granja, São José dos Campos - SP, Brasil - CEP 12227-010
guilherme.chagas, lorena, rafael.santos(@inpe.br)

Introdução

Problemas de agrupamento são oriundos de várias áreas da ciência como, por exemplo, bioinformática, processamento de imagens, *design* de VLSI, visão computacional, análise de dados de multimídia, localização de facilidades, compressão de dados, *marketing*, reconhecimento de padrões e aprendizado de máquina [3, 6, 1].

Uma forma bastante difundida na literatura para se obter bons agrupamentos é modelar esses problemas por meio de grafos [11]. Nesse contexto, há o *problema de edição de clusters*, também conhecido como *problema de agrupamento por correlação* [3]. Este é um problema em otimização combinatória bastante difundido na literatura e é, provavelmente, o problema de agrupamento por modificação de arestas mais estudado [6]. No problema de edição de *clusters* o objetivo é particionar os vértices de um grafo, por meio de adições e remoções de arestas, em subgrafos completos disjuntos. Encontrar o menor número de adições e remoções de arestas que particione o grafo dessa maneira é um problema NP-difícil [11]. Com isso, muitos algoritmos aproximativos e heurísticas foram propostos e muitos estudos teóricos foram realizados ao longo dos anos para esse problema.

Em algumas áreas, porém, é necessário que haja sobreposição de *clusters*, ou seja, que vértices possam pertencer a mais de um *cluster*. Por exemplo, em detecção de comunidades, indivíduos podem pertencer a vários grupos diferentes [4]. Como [6] explicam, com a definição do problema de edição de *clusters* não se consegue modelar problemas em que possa existir sobreposição de *clusters* e essa formulação tem sido criticada na literatura. Com isso, é necessário que haja uma relaxação do problema de edição de *clusters* que possibilite a sobreposição de *clusters*.

Poucos trabalhos são encontrados na literatura para o problema de edição de *clusters* com sobreposição. Em especial, no limite do conhecimento dos autores deste trabalho, não se conhece heurísticas para esse problema. Com isso, neste trabalho, é proposta uma heurística híbrida para o problema de edição de *clusters*. Nessa heurística híbrida, utilizam-se duas meta-heurísticas para se gerar *clusters* que são utilizados para se resolver um modelo por programação linear inteira mista e, com isso, gerar um agrupamento com sobreposição.

Referencial teórico

Seja um grafo $G = (V, E)$ simples, não ponderado e não direcionado, em que V é o conjunto de vértices, E é o conjunto de arestas, $n = |V|$ e $m = |E|$. Dois vértices $v, u \in V$ são adjacentes se, e somente se, $(v, u) \in E$. Um grafo G é *completo* se, e somente se, $\forall v \in V$ e $\forall u \in V$, com $v \neq u$, $(v, u) \in E$. Em um grafo completo, $m = \frac{n \cdot (n-1)}{2}$. Um *subgrafo induzido* de G por um conjunto de vértices $U \subseteq V$ é um grafo $G_U = (U, E_U)$, em que $\forall v \in U$ e $\forall u \in U$, $(v, u) \in E_U$ se, e somente se, $(v, u) \in E$. Um subconjunto de vértices $U \subseteq V$ compõem uma *clique* se o subgrafo de G induzido por U, G_U, é completo.

Dado um grafo G e um agrupamento $\mathcal{C}$, o custo de uma solução do problema de edição de *clusters* é apresentado na equação 1 [5]. Nessa equação, as variáveis x_{ij}, para $1 \leq i < j \leq n$, possuem valores $x_{ij} = 1$ se $\ell_C(i) \cap \ell_C(j) = \emptyset$ e $x_{ij} = 0$ se $\ell_C(i) \cap \ell_C(j) \neq \emptyset$.

$$K_{ce}(G, \mathcal{C}) = \sum_{i<j,\ (i,j) \in E} x_{ij} + \sum_{i<j,\ (i,j) \notin E} (1 - x_{ij}), \quad (1)$$

em que

$$x_{ij} = \begin{cases} 0, & \text{if } \ell_C(i) \cap \ell_C(j) \neq \emptyset, \\ 1, & \text{if } \ell_C(i) \cap \ell_C(j) = \emptyset. \end{cases}$$

Heurística Híbrida

A heurística híbrida proposta neste trabalho consiste em, basicamente, três etapas. Inicialmente, as meta-heurísticas *Biased Random-Key Genetic Algorithm* (BRKGA) [7] e *Simulated Annealing* (SA) [9] são utilizadas para se gerar um histórico de soluções do problema de edição de *clusters* sem sobreposição de um grafo de entrada. Posteriormente, todos os *clusters* que compõem as soluções desse histórico são extraídos para formar um conjunto de *clusters*. Em seguida, esse conjunto de *clusters* é utilizado, pelo CPLEX [8], para resolver a formulação por programação linear inteira mista, apresentado no modelo 2. O objetivo com o histórico de soluções oriundos das meta-heurísticas é gerar um número reduzido de bons *clusters* de entrada para o modelo por programação linear inteira mista. Com a resolução desse modelo, obtém-se uma solução do problema de edição de *clusters* com sobreposição.

$$\max \sum_{i=1}^{N} (d_i \cdot y_i - u_i) \quad (2a)$$

sujeito a

$$\sum_{j=1}^{N} \left| \frac{|C_i \cap C_j|}{|C_i \cup C_j|} - z_i \right| \cdot (y_i + y_j - 1) \leq u_i,\ i = 1, \ldots, N, \quad (2b)$$

$$\sum_{i=1}^{N} y_i = r, \quad (2c)$$

$$\sum_{i=1}^{N} a_{ji} \cdot y_i \geq b,\ j = 1, \ldots, n, \quad (2d)$$

$$y_i \in \{0, 1\}, u_i \in \mathbb{R},\ i = 1, \ldots, N. \quad (2e)$$

Na formulação 2, com as variáveis binárias y_i, para $1 \leq i \leq N$, define-se quais *clusters* C_i pertencem, ou não, à solução de agrupamento com sobreposição. Também, tem-se um custo d_i associado a cada *cluster* C_i que representa o quão bom é esse *cluster*. Esse custo é dado pela Equação 3.

$$d_i = \frac{E_{C_i}^{in}}{E_{C_i}^{\max}} - \frac{E_{C_i}^{out}}{|C_i| \cdot (|V| - |C_i|)}. \quad (3)$$

Como deve-se maximizar a função objetivo 2a, são obtidos os menores valores das variáveis reais u_i. Com essas variáveis são selecionados os *clusters* que possuam as menores diferenças entre os coeficientes de Jaccard, em relação aos outros *clusters*, e os parâmetros de controle de sobreposição z_i. Na restrição 2b, os parâmetros $z_i \in [0, 1]$ são utilizados para controlar as sobreposições entre os *clusters*. Na restrição 2c é estabelecido que exatamente r *clusters* sejam selecionados. É garantido, pela restrição 2d, que cada um dos n vértices pertençam a, ao menos, b *clusters*. Na restrição 2e as variáveis y_i são definidas com binárias e as variáveis u_i como reais.

Um pseudocódigo da heurística híbrida é apresentado no Algoritmo 1.

Algorithm 1: Heurística híbrida.

input : graph $G = (V, E)$; mixed-integer liner program $model$; BRKGA number of generations $gen_{\max}$; BRKGA population size p; BRKGA elite population size p_e; BRKGA mutant population size p_m; BRKGA elite allele inheritance probability ρ_e; SA initial temperature t_i; SA final temperature t_f; SA cooling rate α; SA Metropolis algorithm step size $sa_{\max}$.

output: overlapping cluster editing solution s;

begin
  $hist_{sol} \leftarrow brkga(G, gen_{\max}, p, p_e, p_m, \rho_e)$;
  $hist_{sol} \leftarrow hist_{sol} \cup sa(G, t_i, t_f, \alpha, sa_{\max})$;
  $clusters \leftarrow get\_clusters(hist_{sol})$;
  $s \leftarrow cplex\_solve(G, model, clusters)$;
  $s.compute\_ovlp\_clstring\_cost()$;
  return s;
end

Resultados

Todas as implementações foram realizadas na linguagem C++. Para a resolução dos modelos usou-se o IBM® ILOG® CPLEX® 12.8 [8]. Todos os testes computacionais foram realizados em um computador com processador Intel® Xeon® E5-2687W v2 CPU 3,40GHz $\times$ 8 com 25MiB de memória *cache* e com 62GiB de memória RAM.

Utilizou-se um conjunto de instâncias composto por 30 instâncias, com tamanhos entre 25 e 1000 vértices, geradas pelo algoritmo de [10]. As instâncias desse segundo conjunto possuem soluções ideais de agrupamentos com sobreposição. Com esse conjunto o objetivo foi verificar se a heurística híbrida é capaz de reproduzir o agrupamento original. Para isso utilizou-se a métrica *FBCubed* [2] para avaliar as soluções geradas pela heurística híbrida em relação às soluções ideais.

Figura 1: *Resumo dos resultados da heurística híbrida no testes realizados nas 30 instâncias geradas pelo algoritmo de Lancichinetti e Fortunato [10].*

		number of bests solutions costs					
				HH		HH avg. FBCUBED	
n	#	BRKGA	SA	$z_i = 0$	$z_i = 1$	$z_i = 0$	$z_i = 1$
25	5	2	5	2	**6**	0.46	0.67
50	5	1	4	3	**4**	0.38	0.53
100	5	1	1	3	**7**	0.31	0.54
200	5	2	1	7	**8**	0.28	0.59
500	5	1	0	3	**7**	0.22	0.27
1000	5	1	2	3	**6**	0.09	0.23

Em relação aos resultados da métrica *FBCubed*, observa-se que os melhores resultados foram obtidos com $z_i = 1$. Também, em 16 instâncias a heurística híbrida, com $z_i = 1$, obteve valores da métrica *FBCubed* maiores que $0,5$. Com esses valores da métrica *FBCubed*, os agrupamentos gerados podem ser considerados bons.

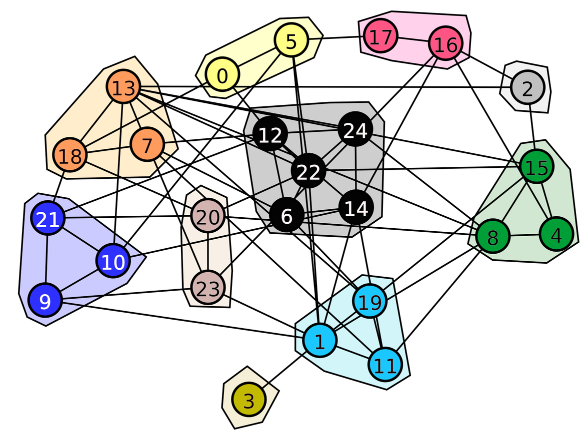

Figura 2: *Uma solução ótima do problema de edição de* clusters*, com custo 44, de uma instância com 25 vértices. Solução obtida com a resolução do modelo de [5].*

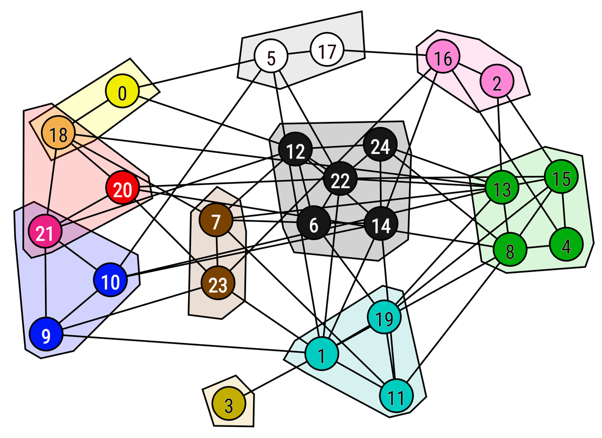

Figura 3: *Uma solução de edição de* clusters *com sobreposição, com custo de 42. Os vértices 18 e 21 pertencem, cada um, a dois* clusters.

Considerações finais

A abordagem proposta apresentou bons resultados em relação aos custos das soluções de edição de *clusters* nos testes supervisionados com as 30 instâncias geradas por meio do algoritmo de [10]. Embora melhorias ainda precisam ser realizadas, a heurística híbrida mostrou-se promissora.

Para trabalhos futuros, deve-se aperfeiçoar alguns pontos da heurística híbrida. Por exemplo, o número de *clusters* a serem utilizados em uma solução com sobreposição e aumentar a variedade do histórico de soluções.

Agradecimentos

Este trabalho foi realizado com o suporte da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), número do processo 301836/2014-0.

Referências

[1] C. C. Aggarwal. An introduction to cluster analysis. In C. C. Aggarwal and C. K. Reddy, editors, *Data Clustering Algorithms and Applications*, chapter 1, pages 1–27. CRC Press, Boca Raton, FL, USA, 2013.

[2] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486, 2009.

[3] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine Learning*, 56(1):89–113, 2004.

[4] F. Bonchi, A. Gioni, and A. Ukkonen. Overlapping correlation clustering. *Knowledge and Information Systems*, 35(1):1–32, 2013.

[5] M. Charikar, V. Guruswami, and A. Wirth. Clustering with qualitative information. *Journal of Computer and System Sciences*, 71(3):360–383, 2005.

[6] M. R. Fellows, J. Guo, C. Komusiewicz, R. Niedermeier, and J. Uhlmann. Graph-based data clustering with overlaps. *Discrete Optimization*, 8(1):2–17, 2011.

[7] J. F. Gonçalves and M. G. C. Resende. Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17(5):487–525, 2011.

[8] IBM Corporation. IBM ILOG CPLEX Optimization Studio V12.8.0 documentation, 2017.

[9] S Kirkpatrick, C D Gelatt, and M P Vecch. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[10] A. Lancichinetti and S. Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, 80(1):016118, 2009.

[11] R. Shamir, R. Sharan, and D. Tsur. Cluster graph modification problems. *Discrete Applied Mathematics*, 144(1-2):173–182, 2004.