Abordagens Heurísticas para o p-Cabo-Trincheira com Localização de Instalações *

Ulysses Rocha[1], Natanael Ramos[1], Lucas Melo[1], Marcelo Benedito[1], André Silva[1], Rafael Cano[1], Flávio Miyazawa[1], Eduardo Xavier[1].

[1]Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)

{ulysses.rocha, natanael.ramos, lucas.melo, marcelo.benedito}@students.ic.unicamp.br,

{andre.silva, rgcano, fkm, ecx}@ic.unicamp.br

Abstract. *We present a new variation of the Cable-Trench Problem, the p-Cable-Trench Problem with Facility Location, in which there are costs to open facilities. We propose two heuristics: one based on Relax-and-Fix and another based on BRKGA, which we evaluate through experiments.*

Resumo. *Apresentamos uma nova variação do Problema Cabo-Trincheira, o p-Cabo-Trincheira com Localização de Instalações, em que há custos de abertura de instalações. Propomos duas heurísticas: uma baseada em Relax-and-Fix e outra baseada em BRKGA, que avaliamos experimentalmente.*

1. Introdução

O Problema Cabo-Trincheira (PCT) foi definido por [Vasko et al. 2002] para modelar redes cabeadas. Nesse problema, um conjunto de clientes precisa ser conectado a um servidor através de cabos dedicados, que são acomodados em uma infraestrutura de trincheiras não capacitadas. O objetivo é minimizar a soma dos custos de construção da infraestrutura de trincheiras e cabeamento. Dado um grafo completo $G = (V, E)$, um vértice $v_r \in V$ e fatores τ e γ não negativos, o objetivo do problema é encontrar uma árvore geradora $T = (V, E_T)$ de G enraizada em v_r que representa a infraestrutura de trincheiras, minimizando $\tau(\sum_{(i,j)\in E_T} d_{ij}) + \gamma(\sum_{v\in V} d_T(v_r, v))$, onde d_{ij} é o custo entre os vértices (i, j) e $d_T(v_r, v)$ o custo entre v_r a v em T. Dentre as aplicações do PCT e suas variantes, destacamos o planejamento de redes cabeadas e sem-fio [Nielsen et al. 2008], e reconstrução de vascularização em imagens de tomografia computadorizada [Vasko et al. 2016].

Neste artigo, apresentamos uma variante do PCT, o Problema Cabo Trincheira com Localização de Instalações (p-PCTLI), em que até p instalações podem ser abertas para servir clientes e cada instalação possui um custo de abertura. Esse problema generaliza o PCT e Localização de Instalações. Além disso, também generaliza a variante apresentada por [Marianov et al. 2012], chamada p-Cabo-Trincheira (p-PCT), onde exatamente p servidores são abertos para servir clientes, mas não há custos de abertura.

Definimos formalmente o p-PCTLI e mostramos uma formulação linear inteira na Seção 2, apresentamos uma heurística baseada em Relax-and-Fix na Seção 3, e apresentamos uma heurística baseada em BRKGA na Seção 4. Concluímos o trabalho avaliando os resultados na Seção 5.

*Financiamento CNPq:306358/2014-0, 311499/2014-7, 133728/2016-1, 425340/2016-3, 155498/2016-9, 131175/2017-3. FAPESP: 2014/14375-9, 2015/11937-9, 2015/11937-9, 2016/23552-7, 2016/12006-1.

2. Problema Cabo-Trincheira com Localização de Instalações

p-**PCTLI :** Seja um grafo direcionado $G = (N, A)$, um conjunto $F \subseteq N$ de candidatos a instalações, funções $l : A \rightarrow \mathbb{R}^+$ de custo de arestas trincheiras, $d : A \rightarrow \mathbb{R}^+$ de custos de arestas cabo e $w : F \rightarrow \mathbb{R}^+$ de custo de abertura de instalações, e um parâmetro positivo p. Encontre uma floresta geradora T com componentes $T_1, T_2, ..., T_k$, onde $k \leq p$ e cada componente $T_i = (V_i, E_i)$ enraizada em um nó $r_i \in F$, tal que $\sum_{i=1}^{k} \left(\sum_{e \in E_i} l_e + \sum_{v \in V_i} d_T(r_i, v) + w_i \right)$ seja mínimo, onde $d_T(r_i, v)$ é o custo de arestas cabo no caminho de r_i a v em T_i.

Para a formulação em PLI do problema, inserimos um nó artificial v_0 conectado a todas as instalações do grafo, com distância de cabo igual a zero e distância de trincheira igual ao custo de abertura da instalação. Assim, restringindo o grau de v_0, podemos utilizar formulações em PLI do PCT ou p-PCT para o p-PCTLI . Utilizamos a formulação *multicommodity-flow* para o p-PCT apresentada em [Marianov et al. 2012], adaptando as restrições e valores para o p-PCTLI . Sendo l_{ij} e d_{ij} os custos de construção de trincheiras e cabos, respectivamente. Definimos $G^+ = (N^+, A^+) : (N \cup n_0, A \cup A_0)$, onde n_0 é um vértice artificial e $A_0 = \{(v_0, j) \ : \forall j \in F \ \}$, com fatores de custo $l_{0i} = w_i$ e $d_{0i} = 0$, $\forall i \in F$. Sendo f_{ij}^k a variável indicadora da passagem do cabo com destino a k no arco (i, j) e x_{ij} a variável indicadora para trincheiras no arco (i, j). Temos abaixo o PLI:

minimizar $\sum_{(i,j) \in A^+} l_{ij} x_{ij} + \sum_{k \in N} \sum_{(i,j) \in A : i \neq k} d_{ij} f_{ij}^k$

$$\text{Restrito a:} \quad \sum_{j \in F} f_{0j}^k = 1 \qquad \forall k \in N, \tag{2.1}$$

$$\sum_{i \in N^+ : i \neq k, (i,k) \in A^+} f_{ik}^k = 1 \qquad \forall k \in N, \tag{2.2}$$

$$\sum_{i \in N^+ : i \neq k; (i,j) \in A^+} f_{ij}^k - \sum_{h \in N : (j,h) \in A} f_{jh}^k = 0, \quad \forall j, k \in N : j \neq k, \tag{2.3}$$

$$\sum_{i \in N^+ : (i,j) \in A^+} x_{ij} = 1 \qquad \forall j \in N, \tag{2.4}$$

$$f_{ij}^k \leq x_{ij} \qquad \forall (i, j) \in A^+, k \in N : i \neq k, \tag{2.5}$$

$$\sum_{j \in F} x_{0j} \leq p. \tag{2.6}$$

A restrição (2.1) define v_0 como origem para todas as *commodities*. A restrição (2.2) faz com que todo vértice receba sua *commodity* correspondente. A restrição (2.4) define que todo vértice seja coberto por uma trincheira. A restrição (2.5) exige que exista uma trincheira na aresta sempre que houverem cabos, e (2.6) restringe o grau de v_0 em no máximo p.

3. Aplicando Relax-and-Fix ao p-PCTLI

Relax-and-fix é uma estratégia heurística que utiliza um modelo de Programação Linear Inteira (PLI) [Wolsey 1998]. Em nossa implementação para o p-PCTLI , utilizamos a formulação apresentada na seção 2. Iterativamente obtemos a solução da relaxação linear do PLI, utilizando-a como guia para fixar algumas variáveis com valores inteiros. Para isso, a cada iteração, executamos a heurística MOD_PRIM [Vasko et al. 2016] no grafo suporte de cada solução do PL, fixando como parte da solução final os p caminhos de menor custo de uma instalação até um novo cliente, como parte da solução final, onde p é um parâmetro dependendo do tamanho da instância. Após fixados os p caminhos, re-otimizamos o PL para fixar os próximos p caminhos.

4. Aplicando BRKGA ao p-PCTLI

O BRKGA [Toso and Resende 2015] é uma metaheurística evolutiva onde cada solução é representada por n chaves aleatórias. Em uma implementação do BRKGA para um determinado problema, é necessário um decodificador que mapeie deterministicamente o vetor de chaves em uma solução viável do problema. Em nosso decodificador, temos uma chave para cada aresta. Para obtermos uma solução viável, as arestas são ordenadas de maneira decrescente por chave e iteradas nessa ordem para construir uma árvore geradora que define as trincheiras utilizadas, de maneira similar ao algoritmo de Kruskal, onde adicionamos uma arestas se esta não formar ciclo e não resultar na abertura de mais de p instalações. Calculamos o custo de cabos para cada cliente, encontrando o caminho único entre o servidor e o cliente com uma busca em largura.

5. Resultados Numéricos

			BRKGA						Relax-and-Fix						Relaxação Lagrangeana					
class	**n**	**p**	$\overline{x}$	σ	**m**	**min**	**max**	**t**	$\overline{x}$	σ	**m**	**min**	**max**	**t**	$\overline{x}$	σ	**m**	**min**	**max**	**t**
chess	288	14	38.13	4.10	38.00	28.00	46.00	30.81	0.078	0.27	0.00	0.00	1.00	91.52	0.63	0.53	1.00	0.00	2.00	18.49
chess	288	28	65.26	11.00	68.50	45.00	84.00	32.07	0.022	0.15	0.00	0.00	1.00	75.99	0.23	0.54	0.00	0.00	3.00	14.81
chess	288	43	75.50	11.34	79.00	56.00	96.00	30.97	0.00	0.00	0.00	0.00	0.00	70.89	7.50	3.09	7.00	2.00	15.00	2.66
fp_11	266	13	50.66	4.82	51.00	41.00	63.00	37.50	0.32	0.58	0.00	0.00	2.00	139.33	0.39	0.57	0.00	0.00	3.00	19.54
fp_11	266	26	79.79	10.45	83.00	61.00	98.00	37.84	0.011	0.11	0.00	0.00	1.00	99.75	0.31	0.59	0.00	0.00	4.00	14.30
fp_11	266	39	80.50	11.26	84.00	62.00	100.00	37.29	0.011	0.11	0.00	0.00	1.00	74.49	4.50	2.52	4.00	1.00	11.00	3.11
gap-a	200	10	40.66	5.46	41.00	28.00	53.00	22.55	0.18	0.41	0.00	0.00	2.00	56.12	0.47	0.54	0.00	0.00	2.00	8.80
gap-a	200	20	65.91	10.34	69.00	47.00	86.00	23.37	0.00	0.00	0.00	0.00	0.00	41.94	0.10	0.30	0.00	0.00	1.00	6.84
gap-a	200	30	71.01	10.99	74.00	51.00	91.00	22.77	0.00	0.00	0.00	0.00	0.00	36.72	0.18	0.41	0.00	0.00	2.00	5.82
gap-b	200	10	40.83	5.29	42.00	29.00	58.00	22.75	0.22	0.54	0.00	0.00	2.00	58.56	0.46	0.50	0.00	0.00	1.00	8.86
gap-b	200	20	66.60	10.44	69.00	48.00	85.00	23.88	0.00	0.00	0.00	0.00	0.00	44.54	0.011	0.11	0.00	0.00	1.00	6.82
gap-b	200	30	70.86	11.43	73.00	50.00	89.00	23.60	0.00	0.00	0.00	0.00	0.00	38.22	0.34	0.77	0.00	0.00	4.00	6.09
gap-c	200	10	41.28	4.62	41.00	30.00	52.00	19.41	0.33	0.58	0.00	0.00	2.00	61.66	0.46	0.58	0.00	0.00	3.00	8.64
gap-c	200	20	68.48	11.10	71.50	48.00	88.00	20.03	0.00	0.00	0.00	0.00	0.00	45.06	0.044	0.21	0.00	0.00	1.00	7.05
gap-c	200	30	72.90	10.86	76.00	54.00	92.00	19.80	0.00	0.00	0.00	0.00	0.00	37.40	0.40	0.68	0.00	0.00	4.00	5.87
pc	256	12	31.01	3.49	31.00	24.00	41.00	20.33	0.21	0.51	0.00	0.00	2.00	73.09	0.53	0.52	1.00	0.00	2.00	14.05
pc	256	25	54.52	8.55	55.50	40.00	71.00	21.06	0.00	0.00	0.00	0.00	0.00	51.83	0.10	0.30	0.00	0.00	1.00	9.95
pc	256	38	69.71	11.31	73.00	50.00	90.00	21.80	0.00	0.00	0.00	0.00	0.00	46.91	3.07	2.90	2.00	0.00	12.00	5.66

Tabela 1. Resumo de Resultados

Nome	p	$\overline{x}$	σ	m	min	max
fp_17	30	11.48	18.15	17.39	-39.12	37.97
fp_17	61	10.11	16.02	12.95	-35.58	39.77
fp_17	92	12.59	17.56	17.20	-71.66	36.53

Tabela 2. Gap (em %) entre o BRKGA e a Relaxação Lagrangeana

O conjunto de instancias utilizadas foi baseado em [Beasley 1990], com adaptações análogas às de [Marianov et al. 2012], adicionando o custo de abertura das instalações que foi escolhido aleatoriamente entre o custo mínimo e máximo das arestas da instância. O *Relax-and-Fix* foi implementado utilizando GUROBI OPTIMIZER, o BRKGA foi implementado utilizando a biblioteca disponibilizada em [Toso and Resende 2015]. O ambiente computacional utilizado foi um Intel(R) Xeon(R) CPU E5-2420, 12 núcleos, 1.90 GHz, 32 GB memória RAM. O *Relax-and-Fix* foi executado com 100 iterações. Os parâmetros utilizados para o BRKGA foram, população 50, elite 20%, mutação 15% , herança de característica 70%, com tempo máximo de 30 minutos e com 10 repetições.

Usamos 180 instancias nas quais todos os métodos foram executados, variando o valor de γ/τ em 0.25, 0.5 e 0.75. p foi definido como 10%, 20% e 30% do número de vértices de cada instância, totalizando 1620 repetições. Para comparar os resultados do BRKGA para instancias onde o PLI não pode ser executado devido a limitações de recursos computacionais, adaptamos a heurística utilizando Relaxação Lagrangeana apresentada em [Marianov et al. 2012]. O resumo dos resultados pode ser observado na Tabela 1. A primeira linha apresenta os métodos de solução. Cada linha subsequente possui resultados médios para as classes de instâncias. A coluna **Nome** informa a classe de instancias que foi executada, n o número de vértices da respectiva classe, $\overline{x}$, σ, **m**, **min** e **max** apresentam a média, desvio padrão, mediana, valores mínimo e máximo para o gap (em %) de função objetivo entre o método de solução e o ótimo. As colunas **t** apresentam o tempo de execução médio para cada classe.

Podemos destacar a heurística Relax-and-Fix, que obteve o custo ótimo para 76.97% das instâncias analisadas, e soluções no máximo 2% acima do ótimo. O BRKGA encontrou em média, resultados 60.2% acima do ótimo, mas com tempo de execução em média 2.45x menor que o Relax-and-Fix. A Relaxação Lagrangeana foi executada em todas as instancias testadas, como apresenta a Tabela 1 obteve bons resultados, mantendo um gap para a solução ótima de em média 1.09%, sendo também em média 9x mais rápida do que a heurística Relax-and-Fix. A Tabela 2, apresenta as instâncias nas quais somente os métodos BRKGA e Relaxação Lagrangeana foram executados, o grupo *fp_17* com instancias de 614 vértices, das 30 instâncias e 270 replicações, o BRKGA apresentou resultado melhor que a Relaxação Lagrangeana em apenas 54 instâncias.

Em trabalhos futuros pretendemos explorar novas formulações e heurísticas envolvendo Relaxação Lagrangeana. Para o Relax-and-Fix, explorar novas formas de fixação de arestas combinando com PLI. Para o BRKGA, pretendemos variar os parâmetros adotados e buscar por novos decodificadores para melhorar os resultados obtidos.

Referências

Beasley, J. E. (1990). OR-Library: distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41(11):1069–1072.

Marianov, V., Gutiérrez-Jarpa, G., Obreque, C., and Cornejo, O. (2012). Lagrangean relaxation heuristics for the p-cable-trench problem. *Comp. Oper. Res.*, 39(3):620–628.

Nielsen, R. H., Riaz, M. T., Pedersen, J. M., and Madsen, O. B. (2008). On the potential of using the cable trench problem in planning of ict access networks. In *ELMAR, 2008. 50th International Symposium*, volume 2, pages 585–588.

Toso, R. F. and Resende, M. G. (2015). A c++ application programming interface for biased random-key genetic algorithms. *Optimization Methods and Soft.*, 30(1):81–93.

Vasko, F. J., Barbieri, R. S., Rieksts, B. Q., Reitmeyer, K. L., and Jr., K. L. S. (2002). The cable trench problem: combining the shortest path and minimum spanning tree problems. *Computers & Operations Research*, 29(5):441 – 458.

Vasko, F. J., Landquist, E., Kresge, G., Tal, A., Jiang, Y., and Papademetris, X. (2016). A simple and efficient strategy for solving very large-scale generalized cable-trench problems. *Networks*, 67(3):199–208.

Wolsey, L. A. (1998). *Integer Programming*. Wiley-Interscience publication.