Leasing K-Center, nova restrição e abordagem heurística

Valdomiro R. D. Neto[1], A. Ribeiro[1], G. P. P. Londe[2], W. R. da Silva[2]

[1]Pontifícia Universidade Católica de Goiás (PUC - Goiás) – Goiânia, GO – Brasil

[2]Instituto de computação
Universidade Estadual de Campinas (UNICAMP) – Capinas, SP – Brazil

{valdomiroapc,guilherme.londe2}@gmail.com, alexribeiro@pucgoias.edu.br,

welverton.silva@unicamp.br

Abstract. *This paper addresses the Leasing K-Center problem, with a new constraint in its formulation, to better reflect its original proposal. As a solution, the VNS metaheuristic is presented, with results that outperform other proposed methods (BRKGA, LS-LUBC and Gurobi solver) and with superior speed for most instances.*

Resumo. *Este trabalho aborda o problema Leasing K-Center, com uma nova restrição na sua formulação, para melhor refletir sua proposta original. Como solução é apresentada a metaheurística VNS, com resultados que superam outros métodos propostos (BRKGA, LS-LUBC e resolvedor Gurobi) e com velocidade superior para a maioria das instâncias.*

1. Introdução

O problema *Leasing K Center* (LKC) foi abordado inicialmente em [dos Santos et al. 2019] e segundo o mesmo trabalho, o LKC é um problema NP-difícil por se tratar de uma generalização do *K-Center (KC)*, que foi provado ser NP-difícil por [Kariv and Hakimi 1979].

No KC dado um conjunto V de localidades e um conjunto C de clientes, tal que $C \subseteq V$ e deve-se que escolher um conjunto F de facilidades, tal que $F \subseteq V$ e $|F| \leq K$ sendo K o número máximo de facilidades que podem ser instaladas, $K \in \mathbb{N}$. O objetivo, tanto do KC quanto do LKC, é minimizar a maior distância entre um cliente e sua facilidade mais próxima. No LKC, o conjunto de clientes pode mudar ao longo do tempo, e facilidades devem permanecer abertas por intervalos de tempo que variam para os diferentes tipos de facilidades.

Neste trabalho foi proposta uma nova restrição para a modelagem matemática do LKC e foi aplicada a metaheurística *Variable Neighborhood Search* (VNS). Foi feita uma comparação com os resultados das metaheurísticas apresentadas em [Londe 2019], que em seu modelo matemático, não contém a restrição apresentada neste trabalho, mas os resultados de suas metaheurísticas a respeitam. Também é feita uma comparação com os resultados do solver GUROBI, que resolve programas lineares com métodos determinísticos como *Branch and bound* e *Simplex*.

A metaheurística VNS foi escolhida para ser aplicada com apoio no trabalho de [dos Santos 2019], por apresentar um bom resultado em um problema muito semelhante ao LKC, o *Leasing K Median (LKM)*. A única diferença entre o LKC e o LKM está na função objetivo.

2. Formulação matemática

Esta formulação foi adaptada do trabalho de [dos Santos et al. 2019]. Sendo V, o conjunto de localidades e i uma facilidade tal que $i \in V$. Sendo T o conjunto de instantes de tempo e t um instante de tempo tal que $t \in T$. Tem-se o conjunto de clientes que devem ser atendidos em um instante t, D_t, e o cliente j tal que $j \in D_t$ e $D_t \subseteq V$. a cada facilidade i o modelo seleciona no conjunto de tipos L, um tipo l que especifica sua duração δ_l. A constante K representa o número máximo de facilidades que podem estar abertas no instante t. A função $d : (i, j) \rightarrow \mathbb{R}$, retorna a distância entre um cliente j e uma facilidade i.

O conjunto de variáveis binárias $x^t_{i,j}$ assume valor 1 se, e somente se, o cliente j se beneficia da facilidade i no instante t. O conjunto de variáveis binárias $y^t_{i,l}$ assume valor 1 se, e somente se, a facilidade i de tipo l, foi aberta no instante t. A variável z representa a maior distância entre qualquer cliente e sua facilidade mais próxima. Segue o modelo de programação linear inteira para o LKC.

$$\text{Minimizar } z \tag{1}$$

Sujeito a:

$$\sum_{j \in D_t} d(i,j) x^t_{i,j} \leq z, \forall i \in V, \forall t \in T \tag{2}$$

$$\sum_{i \in V} x^t_{i,j} = 1, \forall j \in D_t, \forall t \in T \tag{3}$$

$$\sum_{l \in L} \sum_{t'=\max(t-\delta_l+1,1)}^{t} y^{t'}_{i,l} \geq x^t_{i,j}, \forall i \in V, \forall t \in T \tag{4}$$

$$\sum_{i \in V} \sum_{l \in L} \sum_{t'=\max(t-\delta_l+1,1)}^{t} y^{t'}_{i,l} \leq K, \forall t \in T \tag{5}$$

$$\sum_{l \in L} \sum_{t'=\max(t-\delta_l+1,1)}^{t} y^{t'}_{i,l} \leq 1, \forall i \in V, \forall t \in T \tag{6}$$

A restrição (2) garante que z seja a maior distância entre um cliente e a sua facilidade mais próxima. A restrição (3) limita que um cliente seja atendido por apenas uma facilidade. A restrição (4) faz com que um cliente seja atendido apenas por facilidades disponíveis no instante t. A restrição (5) garante que o número de facilidades abertas simultâneamente não supere K. (6) garante que uma facilidade de tipo l aberta em uma localidade i seja única nessa localidade durante todo seu período de duração δ_l.

3. Nova restrição

Uma das contribuições deste trabalho é a proposta da restrição (6). Foi observado que a modelagem do LKC proposta em [dos Santos et al. 2019], permite que uma facilidade seja reaberta estando a mesma, já aberta. Com esta restrição, o modelo reflete melhor a proposta do problema apresentada em [Lintzmayer and Mota 2017].

4. Metaheurística VNS para o LKC

O VNS é uma metaheurística baseada em buscas locais em que a cada iteração, é feita uma alteração na melhor solução encontrada até o momento e depois é aplicada uma busca local. Para mais detalhes sobre o VNS, conferir o trabalho de [Hansen and Mladenović 2003].

Uma solução para o LKC pode ser representada por um conjunto de tuplas S, em que $s = (i, t, l)$ e $s \in S$, i é uma localidade onde foi instalada a facilidade, t é o instante em que a facilidade foi aberta e l é o tipo da facilidade. A função $F : S \rightarrow \mathbb{R}$ retorna o valor objetivo de uma solução S.

Como primeiro passo, gera-se uma solução inicial, executando o algoritmo de Gon descrito em [Garcia-Diaz et al. 2019] para cada instante e em seguida, eles são unidos em intervalos usando a técnica de memorização descrita em [dos Santos 2019]. Assim, tem-se uma melhor solução S até o momento. Como forma de explorar a vizinhança de S, ordena-se as tuplas de S em ordem decrescente do número de clientes que elas atendem. Então, seleciona-se as $|S| - r$ melhores tuplas, onde r é o indice da vizinhança que está sendo explorada ($r \in [1, |S| - 1]$), as restantes são descartadas e novas são geradas aleatoriamente. Tendo uma solução S', gerada na etapa anterior, é aplicada uma busca local. Itera-se entre todas as tuplas $s \in S'$, substituindo a localidade i por cada localidade de V. A alteração feita em S' gera a solução S'', se $F(S'') < F(S')$, substituí-se S' por S''. Após passar por todas as tuplas de S', caso $F(S') < F(S)$, substituí-se S por S'.

5. Experimentos computacionais e conclusões

Os experimentos foram feitos em ambiente Linux, processador Ryzen 3 2200G 3.5 Ghz, quatro núcleos, 8GB de memória RAM. O VNS foi implementado em linguagem C++. A programação linear inteira foi implementada em Python 3.8.1 usando o Gurobi (9.1.2). Para cada teste foi estabelecido o limite de 25 minutos (1500s), foi considerada a melhor solução obtida neste período.

A Tabela 1 contém os resultados das instâncias. As instâncias e os resultados de LS-LUBC e BRKGA são do trabalho de [Londe 2019]. Os resultados do Gurobi foram obtidos com a modelagem exposta neste trabalho. São 20 instâncias divididas em dois grupos de 10. No primeiro grupo, o Gurobi consegue gerar limitantes inferiores para as instâncias. No segundo, o Gurobi não pôde executar as instâncias por serem muito grandes, retornando resultados heurísticos e sem limitantes. Soluções ótimas estão destacadas na Tabela 1. Neste trabalho, considera-se uma solução superior á outra, se o valor objetivo é inferior, ou se o valor objetivo é igual e o tempo para achar a solução inferior.

Para o primeiro grupo de instâncias, o VNS obteve resultados melhores do que o LS-LUBC e BRKGA em todas as instâncias, e melhores do que o Gurobi em 60% das instâncias, com 3 resultados ótimos. O VNS foi também mais rápido com um tempo médio de 1.2s contra 722s, 32.4s, 134.4s de Gurobi, LS-LUBC, BRKGA, respectivamente. Nas instâncias em que teve resultado inferior ao Gurobi, a diferença percentual média entre os dois métodos foi de 4.53%.

No segundo grupo, o VNS superou Gurobi e BRKGA em todas as instâncias e superou LS-LUBC em 60% das instâncias. VNS foi o mais rápido com um tempo médio de 339.3s contra 1500s, 631.3s, 663.9s de Gurobi, LS-LUBC, BRKGA, respectivamente.

Nas instâncias em que teve resultados inferiores ao LS-LUBC, a diferença percentual média entre os dois métodos foi de 2.42%.

Com base nos dados apresentados, observa-se que o VNS tem os melhores resultados entre os métodos apresentados e com velocidade superior, para as instâncias em que apresenta resultados inferiores, possui valores muito próximos dos melhores encontrados.

Tabela 1. Resultados das instâncias.

	Grupo de instâncias 1								Grupo de instâncias 2							
	VNS		Gurobi		LS-LUBC		BRKGA		VNS		Gurobi		LS-LUBC		BRKGA	
id	obj	seg	obj	seg	obj	seg	obj	seg	obj	seg	obj	seg	obj	seg	obj	seg
1	**125**	0	**125**	619	127	34	127	27	224	643	394	1500	226	288	236	1367
2	127	1	209	1500	128	28	133	12	204	4	418	1500	198	866	212	1417
3	**112**	1	**112**	5	123	1	**112**	333	236	1021	400	1500	242	558	240	843
4	**112**	0	**112**	12	116	2	116	1	248	219	388	1500	250	1274	256	562
5	**115**	0	**115**	39	125	4	**115**	403	228	543	458	1500	230	540	234	392
6	102	2	114	1500	102	77	108	41	248	197	302	1500	240	938	256	242
7	100	1	**96**	217	103	34	102	105	224	332	N/A	N/A	220	401	234	949
8	97	2	**95**	66	101	11	100	47	236	68	302	1500	236	137	240	298
9	102	0	170	1500	106	85	111	315	68	238	107	1500	67	845	78	206
10	102	5	**98**	1046	128	48	109	110	68	128	86	1500	68	466	69	363

Referências

dos Santos, J., Londe, G., Ribeiro, A., and da Silva, W. (2019). Formulações e heurısticas para os problemas leasing k-median e leasing k-center. In *Anais do IV Encontro de Teoria da Computação*, Porto Alegre, RS, Brasil. SBC.

dos Santos, J. M. (2019). Heurísticas para o problema Leasing K-Median. `https://repositorio.pucgoias.edu.br/jspui/handle/123456789/3736`. Monografia (Bacharel em Ciência da computação), Pontifícia Universidade Católica de Goiás, Brazil.

Garcia-Diaz, J., Menchaca-Mendez, R., Menchaca-Mendez, R., Pomares Hernández, S., Pérez-Sansalvador, J. C., and Lakouari, N. (2019). Approximation algorithms for the vertex k-center problem: Survey and experimental evaluation. *IEEE Access*, 7:109228–109245.

Hansen, P. and Mladenović, N. (2003). A tutorial on variable neighborhood search. Technical report, Les Cahiers Du GERAD, Hec Montreal and GERAD.

Kariv, O. and Hakimi, S. L. (1979). An algorithmic approach to network location problems. i: The p-centers. *SIAM Journal on Applied Mathematics*, 37(3):513–538.

Lintzmayer, C. N. and Mota, G. O. (2017). Caderno de problemas. In 1o Workshop Paulista em Otimização, Combinatória e Algoritmos. Não publicado.

Londe, G. (2019). Heurísticas para o problema Leasing K-Center. `https://repositorio.pucgoias.edu.br/jspui/handle/123456789/3742`. Monografia (Bacharel em Ciência da computação), Pontifícia Universidade Católica de Goiás, Brazil.