LI Simpósio Brasileiro de Pesquisa Operacional
Limeira–SP, 02 a 06 de setembro de 2019

DESENHO DE REDES DE CONCENTRADORES USANDO MATHEURÍSTICAS

Lucas Gonzaga de Castro [1]
lucas.gonzaga.castro@gmail.com

Elisangela Martins de Sá [1]
elisangelamartins@deii.cefetmg.br

[1] Programa de Modelagem Matemática e Computacional,
Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Av. Amazonas, 7675, CEP 30510-000, Belo Horizonte (MG), Brasil

RESUMO

Neste trabalho é abordado o problema de localização de concentradores incapacitado com alocação múltipla. Para sua solução, propõe-se o uso de duas matheurísticas: Relaxation Enforced Neighborhood Search (RENS) e Variable Neighborhood Descent Branching (VNDB). Para gerar a solução inicial é aplicado o método construtivo RENS que resolve a versão relaxada do problema, fixa as variáveis que assumiram valores inteiros e resolve o subproblema com as variáveis restantes. Depois é aplicado o método de refinamento VNDB que explora o espaço de soluções por meio de trocas exaustivas de estruturas de vizinhança. Os experimentos foram aplicados a um conjunto de instâncias baseado no tráfego de um serviço postal australiano. Os resultados são comparados com as soluções encontradas por um método exato de um aplicativo comercial de propósito geral. Os experimentos computacionais mostram que as mataheurísticas encontram soluções próximas dos resultados do aplicativo comercial de propósito geral em tempo hábil.

PALAVRAS CHAVE. Matheurística, Concentradores, RENS, VNDB.

Tópicos: L&T - Logística e Transportes. MH - Metaheuristicas.

ABSTRACT

This paper has its focus on Uncapacited Multiple Allocation Hub Location problem. In order to solve it, two matheuristics was applied: Relaxation Enforced Neighborhood Search (RENS) and Variable Neighborhood Descent Branching (VNDB). To generate the initial solution, the RENS constructive method is applied, which solves the relaxed version of the problem, fixes the variables that assume integer values and solves the subproblem with the remaining variables. Then the VNDB refinement method is applied which exploits the solution space by exhaustive neighborhood structure changes. The experiments were applied to a set of instances based on traffic from an Australian postal service. The results are compared to the solutions found by an exact method of a general purpose commercial application. Computational experiments show that mataheuristics find solutions close to the results of the general purpose commercial applicationin a satisfactory time.

KEYWORDS. Matheuristic, Hub, RENS, VNDB.

Tópicos: LT – Logistics and transportation; MH – Metaheuristics

https://proceedings.science/p/106918?lang=pt-br

1. Introdução

Concentradores são nós especiais em uma rede de transporte que têm como funções coletar, consolidar, transferir e distribuir o fluxo da rede. O transporte entre nós não concentradores não é feita diretamente, mas por via de nós concentradores em que eles são alocados. A principal vantagem de redes de concentradores está no transporte de fluxo entre concentradores, usufruindo assim de uma economia de escala. Esse desconto no custo de transporte unitário contribui para diminuir o custo total. São diversas áreas que aplicam o problema de localização de concentrador, entre eles, transporte aéreo, telecomunicações, sistema de entrega.

As escolhas das localizações dos concentradores e alocações dos nós não concentradores são partes do problema de projeto da rede de concentradores que devem ser consideradas ao mesmo tempo. O problema de localização de concentradores pertence à classe de problema NP-difíceis [Garey e Johnson, 1979]. Portanto, a determinação da solução ótima para o problema por meio de algoritmos exatos, constitui em um grande desafio, pois deve utilizar métodos que exigem grande esforço computacional e podem ser impraticáveis para exemplares do problema de grandes dimensões. Por isto, diversas propostas usando heurísticas e metaheurísticas são consideradas, tais como: Busca Tabu Klincewicz1998, *Simulated Annealing* [Abdinnour-Helm e Venkataramanan, 1998], Redes Neurais Artificiais [Smith et al., 1996], Algoritmos Genéticos [Abdinnour-Helm e Venkataramanan, 1998; Topcuoglu et al., 2005; Cunha e Silva, 2007].

Existem vários critérios que definem as classificações do problema de localização de concentradores. O principal deles é quanto ao número de concentradores que serão instalados na rede de concentradores. Esse número pode ser passado por parâmetro, caracterizando um problema de *p*-concentradores, ou determinado pelo modelo. Outra diferenciação é a respeito sobre a forma de alocação dos nós não concentradores aos concentradores. Essa alocação pode ser feita de forma única (cada nó não concentrador é alocado à um único concentrador) ou múltipla (cada nó não concentrador pode ser alocado à mais de um concentrador). O nível de rede dos concentradores pode ser completa (existe uma conexão para cada par de concentradores da rede) ou incompleta (pares de concentradores não necessariamente são interligados). A conexão direta de nós não concentradores pode ser permitida ou não.

Devido ao grande número de desenhos possíveis para uma rede de concentradores, [O'Kelly e Miller, 1994] introduziram uma classificação topológica de redes de concentradores em oito protocolos que se diferem em três padrões de características de rede: alocação dos nós não concentradores, simples ou múltipla, conexão direta entre nós não concentradores, se é permitida ou não, e por fim, quanto a interconexão entre os concentradores, completa ou incompleta. A Figura 1 ilustra as diferenças entre os protocolos.

Além de todas essas separações, os trabalhos sobre o problema de localização de concentradores também se dividem aos aspecto de: capacidade, congestionamento e nível de serviço. A introdução da capacidade limita o fluxo que pode passar em arco ou por um concentrador, deixando o problema mais complexo [Martín e González, 2008; Ebery et al., 2000; E.Z. e Alumur, 2016; Merakl e Yaman, 2017; Correia et al., 2018]. Quando concentradores e arcos atuam próximos de sua capacidade máxima, ou mesmo se são incapacitados mas roteiam um intenso tráfico, pode ser gerado um congestionamento. O congestionamento afeta diretamente no tempo de redistribuição do fluxo. Alguns trabalhos penalizam o congestionamento na função objetivo ou o consideram usando restrições adicionais [Miranda et al., 2011; Camargo et al., 2009]. O nível de serviço se refere a qualidade do tempo de um viagem entre um par de nó da rede. Onde, o tempo de uma viagem é a soma do tempo de transporte de cada arco mais o tempo de manipulação nos concentradores. Em algumas aplicações do problema, o nível de serviço é um fator que deve ser priorizado pois é preciso

fazer o transporte em um tempo limite [Martins de Sá et al., 2017; Martins de Sá et al., 2015; Yaman et al., 2007]. A priorização do nível de serviço, deixa o problema mais complexo ainda, pois ela é conflitante com a função objetivo de minimizar os custos totais.

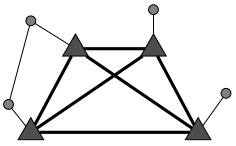
Neste trabalho será abordado o Problema de Localização de Concentradores de Múltipla Alocação Incapacitado (PLCMI) introduzida por [O'Kelly, 1992]. Este problema busca determinar a localização dos concentradores, a alocação dos nós não concentradores e as rotas de fluxo entre a rede minimizando os custos. Usando a definição de protocolos como descrito em [O'Kelly e Miller, 1994], esse problema é classificado como protocolo F, em que a alocação é múltipla, não é permitido transporte direto entre nós não concentradores e o nível de rede de concentradores é incompleta.

Este trabalho tem o papel de contribuir para a pesquisa sobre o problema de localização de concentradores mostrando que com o uso correto de heurísticas é possível encontrar boas soluções em tempo hábil. Então são utilizadas duas matheurísticas, RENS e VNDB, para resolver o PLCMI. Para gerar a solução inicial é aplicado o método construtivo RENS que resolve a versão relaxada do problema, fixa as variáveis que assumiram valores inteiros e resolve o subproblema com as variáveis

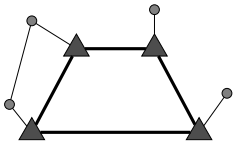


(a) Protocolo A

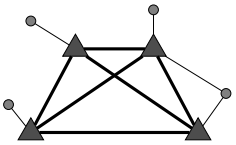
(b) Protocolo B



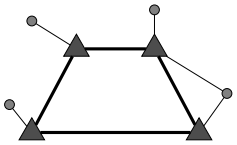
(c) Protocolo C



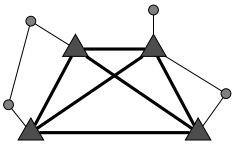
(d) Protocolo D



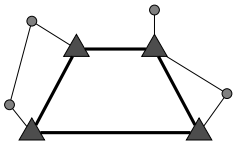
(e) Protocolo E



(f) Protocolo F



(g) Protocolo G



(h) Protocolo H

Figura 1: Exemplo de redes do protocolo A-H

restantes. Depois é aplicado o método de refinamento VNDB que explora o espaço de soluções por meio de trocas exaustivas de estruturas de vizinhança. Para comparar os resultados, foi resolvido o modelo matemático do problema, através de uma programação inteira mista, para obter a solução ótima das instâncias.

Existem várias aplicações para o PLCMI, como por exemplo, companhias aéreas que querem reduzir os custos de transporte agregando um fluxo maior entre concentradores para usufruir da economia de escala. Empresas de transporte de cargas em que é preciso escolher centros de distribuição para rotear o fluxo minimizando os custos.

O restante deste trabalho está estruturado como segue. Na Seção 2 o problema em questão é caracterizado. A Seção 3 apresenta a metodologia, resultados computacionais são apresentados na Seção 4. Finalmente, na Seção 5, conclui-se este trabalho, bem como apresentam-se possíveis propostas a serem exploradas em trabalhos futuros.

2. Caracterização do Problema

Este trabalho usa a formulação baseada de [Ernst e Krishnamoorthy, 1996] e [Campbell, 1994], em que múltipla alocação é permitida, nó não concentrador pode ligar a mais de um concentrador, os concentradores são incapacitados, não possuem limite de fluxo, e há um custo fixo de instalação para cada concentrador. Sendo assim, este é um problema de alocação múltipla de localização de concentrador incapacitado com custo fixo.

Algumas suposições são feitas: (i) O nível de rede do concentrador é uma rede conectada, ou seja, existe um caminho entre qualquer par de nós. (ii) Cada nó não concentrador pode ser alocado a mais de um concentrador. (iii) Nós não concentradores não podem ser ligados diretamente. (iv) Apenas nós concentradores podem funcionar como transbordo. (v) Um fator de desconto α ($0 \leq \alpha \leq 1$) representa uma economia de escala para cada arco entre concentradores.

Parâmetros do problema:

- n : número de nós na rede
- H_k : custo de instalação do concentrador k
- I_{km} : custo de instalação do arco entre concentradores (k, m)
- w_{ij} : demanda que deve ser roteado da origem i até o destino j
- c_{km} : custo de roteamento de uma unidade de fluxo no arco (k, m)
- α : economia de escala
- $o_i = \sum_{j=1}^{n} w_{ij}$: demanda total originada no nó i
- $d_j = \sum_{i=1}^{n} w_{ij}$: demanda total destinada ao nó j

Variáveis do problema:

- z_k : variável binária que indica se um concentrador foi instalado no nó (k).
- y_{km} : variável binária que indica se um arco entre concentrador foi instalado entre os concentradores k e m.
- f_{ikm} : quantidade de fluxo originado no nó i que é roteado no arco entre concentradores (k, m)

- a_{ik} : quantidade de fluxo originado no nó i que é transportada usando o arco de acesso (i, k)
- b_{ijk} : quantidade de fluxo originado no nó i e destinado ao nó j que é transportada usando o arco de acesso (k, j)

A formulação para o problema pode ser dado por:

$$\text{Min} \sum_k H_k z_k + \sum_k \sum_{m:m\neq k} I_{km} y_{km} + \sum_i \left[\sum_k c_{ik} a_{ik} + \sum_k \sum_{m:m\neq k} \alpha c_{km} f_{ikm} + \sum_j \sum_m c_{mj} b_{ijm} \right] \quad (1)$$

$$\text{subject to } y_{km} \leq z_k \quad \forall k, m : k \neq m \quad (2)$$

$$y_{km} \leq z_m \quad \forall k, m : k \neq m \quad (3)$$

$$\sum_k a_{ik} = o_i \quad \forall i \quad (4)$$

$$\sum_k b_{ijk} = w_{ij} \quad \forall i, j : i \neq j \quad (5)$$

$$a_{ik} + \sum_{\substack{m \\ m\neq k}} f_{imk} = \sum_j b_{ijk} + \sum_{\substack{m \\ m\neq k}} f_{ikm} \quad \forall i, k \quad (6)$$

$$a_{ik} \leq o_i z_k \quad \forall i, k \quad (7)$$

$$\sum_i b_{ijk} \leq d_j z_k \quad \forall j, k \quad (8)$$

$$\sum_i f_{ikm} \leq (\sum_i o_i) y_{km} \quad \forall k, m : k \neq m \quad (9)$$

$$f_{ikm}, a_{ik}, b_{ijk} \geq 0 \quad \forall i, j, k, m \quad (10)$$

$$y_{km}, z_k \in \{0, 1\} \quad \forall k, m : k \neq m. \quad (11)$$

A função objetivo 1 minimiza a soma dos custos de instalação de concentradores, ativação de arcos entre concentradores e transporte de fluxo entre cada par de origem e destino. As restrições 2 e 3 garantem que só exista um arco entre concentradores, se os concentradores estiverem instalados. As restrições 4 - 6 são restrições de conservação de fluxo que garantem que a demanda w_{ij} saia do nó i, e chegue no nó j, mas seja devidamente contabilizada em um concentrador k. As restrições 7 - 9 são para garantir que todo o fluxo de demanda passará apenas por arcos e concentradores que foram instalados. Finalmente, as restrições 10 e 11 correspondem às restrições de não negatividade e integralidade respectivamente.

3. Matheurísticas

[Puchinger e Raidl, 2005] definem a matheurística como método para obtenção de soluções incumbentes e limitantes no geral. Já heurísticas e metaheurísticas são frequentemente usadas para determinar limitantes e soluções incumbentes em aproximações de branch and bound.

Segundo [Hansen et al., 2009], há duas razões para dar ênfase a este tipo de combinação. Primeiro, a literatura tem demonstrado a possibilidade de utilização extremamente eficaz das meta-heurísticas para resolver problemas de otimização. No entanto, as meta-heurísticas atuais não fazem

uso de ferramentas matemáticas explícitas. Em segundo lugar, ferramentas matemáticas inovadoras têm sido propostas e se mostrado bastante eficazes em problemas de otimização combinatória, estocástica ou contínua. Entretanto, essas ferramentas visam a solução exata desses problemas.

Para [Hanafi e Todosijević, 2017] a vantagem de uma heurística bem projetada e cuidadosamente escolhida não é apenas pela sua capacidade de produzir soluções de alta qualidade em um curto intervalo de tempo, mas também o fato de poderem facilmente combinar com métodos exatos para acelerá-los.

Neste trabalho foram usadas as matheurísticas RENS, para gerar uma solução inicial, e VNDB, que aplica o refinamento na solução.

3.1. Solução Inicial - RENS

O método RENS foi proposto por [Berthold, 2007] como uma nova heurística em que a motivação é trabalhar na vizinhança da solução do relaxado. O RENS começa resolvendo a versão relaxada do problema. Depois disso, todas as variáveis inteiras que receberam valores inteiros na solução relaxada do problema linear são fixadas, e a integralidade das variáveis é restabelecida gerando um subproblema de programação inteira mista, denominado como sub-MIP. Então uma busca de vizinhança extensa é executada nas variáveis restantes. Se o sub-MIP é resolvido por otimalidade, a solução obtida é o melhor arredondamento de uma solução fracionária LP que qualquer heurística de arredondamento puro pode gerar.

Algoritmo 1: RENS

Entrada: min {cx: x ∈ X, x_j∈ {0,1} para j ∈ I}
Saída: s'
1 **início**
2 $\bar{x} \leftarrow$ ResolvaProblemaRelaxado ;
3 $J \leftarrow \{j \in I : \bar{x_j} \in \{0,1\}\}$;
4 $s' \leftarrow$ arg min {cx: x ∈ $X, x_j = \bar{x_j}$ para j ∈ J} ; `// Resolve o subproblema MIP`
5 **fim**

O Algoritmo 1 ilustra o método RENS. Ele recebe como entrada o modelo de otimazação e tem como saída uma solução inicial. O Algoritmo inicia resolvendo a versão relaxada do problema. Após a otimização, é definido um conjunto J que corresponde às variáveis que assumiram valores inteiros. Por fim, o subproblema MIP é construído fixando as variáveis do conjunto J. A solução inicial s' é obtida após a otimização do

3.2. Refinamento - VNDB

Variable Neighborhood Descent Branching é um método de busca local que explora o espaço de soluções por meio de trocas de estruturas de vizinhança aceitando somente soluções de melhora da solução corrente e retornando à primeira estrutura quando uma solução melhor é encontrada [Hanafi e Todosijević, 2017]. Caso não seja encontrada uma solução de melhora a busca continuará na estrutura de vizinhança seguinte. Caso não exista uma estrutura de vizinhança seguinte a execução é interrompida e a solução corrente é retornada como a solução ótima local. Podem ser fixados valores de tempo limite e iterações sem melhora para parar o algoritmo também.

Através da distância de Hammining $\delta(\tilde{x}, x) \leq k$, são aplicados cortes de Local Branching geradas por $\mathcal{N}_k(\tilde{x}) = \{x \in X : \delta(\tilde{x}, x) \leq k\}$. Além desses cortes, podem ser introduzidos cortes canônicos [Balas e Jeroslow, 192] no algoritmo, fazendo com que o otimizador busque somente por soluções que sejam melhores que a solução atual, ou seja, em um problema de minimização é adicionada uma nova restrição na qual o valor da função objetivo buscada é menor ou igual ao valor da solução atual decrescida do valor 1, quando os coeficientes da função objetivo forem inteiros,

https://proceedings.science/p/106918?lang=pt-br

ou menor ou igual ao valor da solução atual decrescida de um valor ξ suficientemente pequeno, no caso dos coeficientes forem reais [Rezende et al., 2016].

No Algoritmo 2 são dados de entrada a solução inicial encontrada pelo RENS, o tamanho na vizinhança inicial k_0, tempo t_{sub} para resolver cada subproblema, tempo máximo t_{max} para rodar o VNDB e o número de iterações máxima sem melhora it_{max}. A saída será uma solução refinada $\tilde{x}$.

Algoritmo 2: VNDB

Entrada: $\tilde{x}, k_0, t_{sub}, t_{max}, it_{max}$
Saída: $\tilde{x}$

1 /* Seja $\mathcal{N}_k(\tilde{x}) = \{x \in X : \delta(\tilde{x}, x) \leq k\}$ */
2 $k \leftarrow k_0$; $it \leftarrow 0$
3 $t \leftarrow CPUTime()$
4 **repita**
5 $\tilde{x}' \leftarrow$ SoluçãoMIP($\mathcal{N}_k(\tilde{x})$)
6 **if** *Solução ótima foi encontrada* **then**
7 Adicione o pseudo-corte por $\delta(\tilde{x}, x) \geq k + 1$
8 **if** $c\tilde{x}' < c\tilde{x}$ **then**
9 $\tilde{x} \leftarrow \tilde{x}'$
10 $it \leftarrow 0$
11 $k \leftarrow k_0$
12 **else**
13 $k \leftarrow k + \lceil k \times 0.5 \rceil$
14 **end**
15 **else if** *Solução viável encontrada* & $c\tilde{x}' < c\tilde{x}$ **then**
16 Adicione o pseudo-corte por $\delta(\tilde{x}, x) \geq 1$
17 $\tilde{x} \leftarrow \tilde{x}'$
18 $it \leftarrow 0$
19 $k \leftarrow k_0$
20 **else if** *Solução inviável* **then**
21 Adicione o pseudo-corte por $\delta(\tilde{x}, x) \geq k + 1$
22 $k \leftarrow k + \lceil k \times 0.5 \rceil$
23 $it \leftarrow it + 1$
24 **até** $t \leq t_{max}$ *e* $it \leq it_{max}$;
25 **retorna** $\tilde{x}$

Na linha 2 o valor de k é setado para iniciar na última estrutura da vizinhança. Nas linhas 3 e 4 são inicializados as variáveis que marcam o tempo de execução e o número de iterações sem melhoria, elas são necessárias para encerrar o algoritmo. Na linha 5 é feita uma busca local em $\mathcal{N}_k(\tilde{x})$ para encontrar uma solução $\tilde{x}'$. Então é conferida sua viabilidade, e para cada caso é tomada uma decisão para continuar a iteratividade do algoritmo. Na linha 7, quando a solução ótima é encontrada em uma dada vizinhança, é feito um novo pseudo-corte por $\delta(\tilde{x}, x) \geq k + 1$. Se a nova solução for melhor que a atual, nas linhas 9, 10 e 11 são atualizados o valores da solução, recebe o valor da solução nova, do contador de iterações sem melhoria, volta a ser zero, e de k, volta para k_0 que é a primeira estrutura de vizinhança. Caso não houver melhoria, na linha 13 é atualizado o valor de k para k + $\lceil k \times 0.5 \rceil$ Na linha 16, quando a solução é viável e há melhoria na solução, é feito um novo pseudo-corte por $\delta(\tilde{x}, x) \leq 1$, e nas linhas 17, 18 e 19 são atualizados os valores

da solução, contador de iterações sem melhoria e k. Na linha 21, quando não há solução viável, é adicionado um novo pseudo-corte $\delta(\tilde{x}, x) \leq k + 1$, e na linha 22 é atualizado o valor de k para $k + \lceil k \times 0.5 \rceil$. Por fim, na linha 23 é iterado o número de iterações sem melhoria, e a linha 24 determina o encerramento do algoritmo, que é quando atinge o tempo máximo recebido na entrada, ou quando número de iterações máxima sem melhoria é alcançado.

4. Resultados Computacionais

Para a realização dos experimentos computacionais, foram usadas instâncias AP de 10 à 40 nós disponibilizadas na *OR-Library* de [Beasley, 1996]. Estas instâncias foram elaboradas, a partir de dados de um serviço postal australiano, por [Ernst e Krishnamoorthy, 1996].

Os algoritmos foram escritos na linguagem C++, utilizando-se o IDE *Visual Studio 2017* 15.9.2 e o compilador GCC 7.3.0. Para resolver o problema exato foi utilizado o solver do CPLEX 12.8 limitando o uso de apenas um núcleo de processamento. Para gerar a solução das instâncias, o $solver$ foi modificado para desativar as heurísticas e cortes, também foi enfatizado a viabilidade sobre otimização. Dessa forma, fica mais justo a comparação das soluções do CPLEX com as soluções geradas através dos métodos matheurísticos. Todos os experimentos computacionais foram executados em um computador Intel Core i5-3330 com 8GB de RAM e 3.0GHz de clock, sob sistema operacional Windows 10.

Os testes foram feitos utilizando dois valores diferentes de economia de escala, α = 0.2 e 0.5, o fator que multiplica o custo de instalação de arco usado foi β = 1. O tempo limite para solução do CPLEX foi de uma hora. O custo de transporte de um nó para outro, c_{ij}, é calculada pela distância euclidiana entre as os nós. Para solução inicial gerada por RENS foi estipulado o valor de tempo limite t_{RENS} = 100 segundos. E por fim, para o refinamento foram usadas duas soluções iniciais de RENS, uma com o tempo limite de 100 segundos e outra com uma hora. Os parâmetros de VNDB foram fixados em k_0 = 4, t_{sub} = 30 segundos , t_{max} = 3600 segundos e it_{max} = 10, ou seja, ele busca numa vizinhança de tamanho 4, gastando apenas 30 segundos em cada subproblema e no máximo uma hora de execução ou 10 iterações sem melhora.

A métrica utilizada para comparar os resultados obtidos pelos algoritmos, é dada pela Eq. 12. Nessa medida, calcula-se o desvio gap das soluções finais produzidas pelos algoritmos(f^{Alg}) em relação ao valor ótimo ou pela melhor solução encontrada até o momento (f^*).

$$gap = \frac{f^{Alg} - f^*}{f^*} \tag{12}$$

O nome das instâncias foram codificadas para AP $n.\alpha$, onde n é o número de nós e α = $\{2, 5\}$ representa o fator de desconto 0.2 e 0.5 respectivamente.

Tabela 1: Resultados CPLEX × Relaxado × Matheurísticas

	CPLEX		Relaxado		RENS			VNDB		
Instância	Resultado	t (s)	Resultado	t (s)	Resultado	gap (%)	t (s)	Resultado	gap (%)	t (s)
AP10.2	43950757	30,03	27791141	0,01	46971834	6,87	90,69	43989600	0,08	371,74
A20.2	61937622	3549,60	43223309	0,17	63864463	3,11	93,06	62158003	0,35	314,78
A25.2	63028481	3562,14	45609447	0,92	67894686	7,72	93,72	63238480	0,33	326,74
A40.2	-	3549,34	52221542	14,65	71586258	3,37	109,94	71586258	3,37	313,29
A10.5	56832789	6,35	41766123	0,01	56832789	0,00	93,79	56832789	0,00	404,60
A20.5	69201169	1452,37	54528047	0,16	69256732	0,08	88,25	69256732	0,08	375,98
A25.5	69639167	3554,23	54528047	0,58	69867169	0,32	93,29	69867169	0,32	377,52
A40.5	-	3549,345	60969771	10,36	74246393	0,72	99,94	74246393	0,72	394,00

Tabela 2: Resultados VNDB partindo de diferentes soluções iniciais

	RENS100s-VNDB			RENS1h-VNDB		
Instância	Resultado	gap (%)	t (s)	Resultado	gap (%)	t (s)
10.2LL	43989600	0,08	371,74	43950757	0,00	1755,46
20.2LL	62158003	0,35	314,78	62035060	0,15	3505,45
25.2LL	63238480	0,33	326,74	63108521	0,12	3615,44
40.2LL	71586258	3,37	313,29	69171470	0,00	3617,33
10.5LL	56832789	0,00	404,60	56832789	0,00	413,02
20.5LL	69256732	0,08	375,98	69201169	0,00	3605,03
25.5LL	69867169	0,32	377,52	69639167	0,00	3602,06
40.5LL	74246393	0,72	394,00	73729371	0,00	3602,98

A Tabela 1 mostra que na maioria das instâncias, o CPLEX não conseguiu encontrar a solução ótima, estourando o tempo limite de uma hora. Para as instâncias de 40 nós, nenhuma solução foi encontrada.

É possível notar que as soluções encontradas pela matheurística RENS distancia em até 7,22% da solução encontrada pelo CPLEX em um tempo satisfatório. Outro ponto a se destacar é o fato de RENS encontrar uma solução para as instâncias de 40 nós e ainda em tão pouco tempo de execução. Os resultados mostram que acoplar a matheurística RENS no $solver$ melhoraria seu desempenho em encontrar soluções viáveis.

Fazendo um refinamento VNDB na solução inicial foi possível melhorar ainda mais as soluções encontradas por RENS quando $\alpha = 0.2$, distanciando em até 0,35% da solução do CPLEX para instâncias de até 25 nós, sem comprometer muito o tempo. Dependendo da aplicação em que será aplicada a matheurística, é preferível tolerar um pouco mais o tempo de execução para obter uma solução melhor. A execução do VNDB foi interrompida antes do tempo limite de uma hora em ambos parâmetros de α, sendo assim ele foi encerrado porque ficou dez iterações sem melhoria.

Na Tabela 2, há um comparativo do refinamento VNDB sendo aplicado em soluções iniciais diferentes. Na coluna RENS100s-VNDB é exibido os resultados após RENS rodar por 100 segundos, e a RENS1h-VNDB após RENS rodar por uma hora como tempo limite. Com o aumento do tempo para gerar a solução inicial, é possível encontrar soluções melhores, principalmente para as instâncias de 40 nós que o refinamento não conseguiu melhorar a solução inicial. Por outro lado, o tempo de solução aumentou cerca de dez vezes mais. A decisão de priorizar o tempo de execução

ou a qualidade da solução varia de aplicação para aplicação e deve ser tomada com bastante delicadeza.

5. Conclusões e trabalhos futuros

Este trabalho teve seu foco na resolução do problema de localização de concentradores incapacitado de alocação múltipla. Foi aplicado o método matheurístico RENS que resolve a versão relaxada do problema, fixa as variáveis que assumiram valores inteiros e resolve o sub-MIP com as variáveis restantes. Depois é aplicado o método de refinamento VNDB que explora o espaço de soluções por meio de trocas exaustivas de estruturas de vizinhança.

Para comparar os resultados, foi implementado o modelado matemático do problema, e através de um solver comercial é obtida a solução ótima. As instâncias propostas por [Ernst e Krishnamoorthy, 1996] mostram que o uso de matheurísticas facilita muito o processo obtenção da solução. O CPLEX chegou no tempo limite para gerar solução do CPLEX nas instâncias em geral, enquanto a solução através de RENS levou cem segundos sem distanciar muito da solução do CPLEX. A solução pode ser melhorada com um refinamento VNDB sem comprometer muito o tempo.

Aumentando o tempo de solução da matheurística é possível melhorar a qualidade da solução. Usando t_{RENS} = 3600 e ainda fazendo um refinamento VNDB, a solução foi melhorada. Algumas inclusive foram idênticas à solução do CPLEX. ainda mais o mesmo tipo de vizinhança.

As matheurísticas encontraram soluções para instâncias com 40 nós, testes que mais se aproximam da realidade, mostrando uma superioridade ao método de solução exata que não encontrou nenhuma solução viável no tempo limite. Sendo assim é bastante apropriado o uso de matheurísticas em aplicações que não é preciso da solução exata e também no $solver$, melhorando seu desempenho.

Como trabalhos futuros, apontam-se os seguintes: (*i*) Introduzidas novas matheurísticas apos um estudo literatura; (*ii*) Utilizar o *IRACE* para calibrar os parâmetros do VNDB; (*iii*) Realizar mais baterias de testes variando o valor de k_0 e os tempos limites, iterações máximas sem melhora it_{max} do VNDB.

Referências

Abdinnour-Helm, S. e Venkataramanan, M. A. (1998). Solution approaches to the hub location problems. *Annals of Operations Research*, 78:31–50.

Balas, E. e Jeroslow, R. (192). Canonical cuts on the unit hypercube. siam journal on applied mathematics. *SIAM Journal on Applied Mathematics*, 23:61–69.

Beasley, J. E. (1996). Obtaining test problems via internet. *Journal of Global Optimization*, 8(4): 429–433.

Berthold, T. (2007). Rens-relaxation enforced neighborhood search. *Technical report tr*, p. 07–28.

Camargo, R. S., Miranda Jr, G., e Luna, H. P. (2009). Benders decomposition for hub location problems with economies of scale. *Transportation Science*, 43:86–97.

Campbell, J. F. (1994). Integer programming formulations of discrete hub location problems. *European Journal of Operational Research*, 72:387–405.

Correia, I., Nickel, S., e Saldanha-da Gama, F. (2018). A stochastic multi-period capacitated multiple allocation hub location problem: Formulation and inequalities. *Omega*, 74(C):122–134. URL https://ideas.repec.org/a/eee/jomega/v74y2018icp122-134.html.

Cunha, C. e Silva, M. (2007). A genetic algorithm for the problem of configuring a hub-and-spoke network for a ltl trucking company in brazil. *European Journal of Operational Research*, 179: 747–758.

Ebery, J., Krishnamoorthy, M., Ernst, A., e Boland, N. (2000). The capacitated multiple allocation hub location problem: Formulations and algorithms. 120:614–631.

Ernst, A. T. e Krishnamoorthy, M. (1996). Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science*, 4:139–154.

E.Z., S. e Alumur, S. (2016). The design of capacitated intermodal hub networks with different vehicle types. *Transportation Research Part B: Methodological*, 86:51–65. URL https://doi.org/10.1016/j.trb.2016.01.011.

Garey, M. R. e Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York.

Hanafi, S. e Todosijević, R. (2017). Mathematical programming based heuristics for the 0–1 mip: a survey. *Journal of Heuristics*, 23(4):165–206.

Hansen, P., Maniezzo, V., e Voß, S. (2009). Special issue on mathematical contributions to metaheuristics editorial. *International Work-conference on the Interplay Between Natural and Artificial Computation*, 15:197–199.

Martín, I. R. e González, J. J. S. (2008). Solving a capacitated hub location problem. *European Journal of Operational Research*, 184:468–479.

Martins de Sá, E., Contreras, I., Cordeau, J.-F., Saraiva de Camargo, R., e de Miranda, G. (2015). The hub line location problem. *Transportation Science*, 49(3):500–518.

Martins de Sá, E., Morabito, R., e Saraiva de Camargo, R. (2017). Efficient benders decomposition algorithms for the robust multiple allocation incomplete hub location problem with service time requirements. 93.

Merakl, M. e Yaman, H. (2017). A capacitated hub location problem under hose demand uncertainty. *Comput. Oper. Res.*, 88(C):58–70. ISSN 0305-0548. URL https://doi.org/10.1016/j.cor.2017.06.011.

Miranda, G., Luna, H., de Camargo, R., e Pinto, L. (2011). Tree network design avoiding congestion. *Applied Mathematical Modelling*, 35(9):4175 – 4188. ISSN 0307-904X. URL http://www.sciencedirect.com/science/article/pii/S0307904X1100117X.

O'Kelly, M. E. (1992). Hub facility location with fixed costs. *Papers in Regional Science*, 71(3): 293–306.

O'Kelly, M. E. e Miller, H. J. (1994). The hub network design problem: A review and synthesis. *Journal of Transport Geography*, 2(1):31–40. ISSN 0966-6923.

https://proceedings.science/p/106918?lang=pt-br

Puchinger, J. e Raidl, G. R. (2005). Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. *International Work-conference on the Interplay Between Natural and Artificial Computation*, 3562:41–53.

Rezende, J. C., Souza, M. J., e Cota, L. P. (2016). Um novo algoritmo híbrido para resolução de problemas binários. *Anais do XLVIII SBPO*.

Smith, K., Krishnamoorthy, M., e Palaniswami, M. (1996). Neural versus traditional approaches to the location of interacting hub facilities. *Location Science*, 4(3):155 – 171. ISSN 0966-8349. URL `http://www.sciencedirect.com/science/article/pii/S0966834996000174`. Hub Location.

Topcuoglu, H., Corut, F., Ermis, M., e Yilmaz, G. (2005). Solving the uncapacitated hub location problem using genetic algorithms. *Computers and Operations Research*, 32(4):967–984.

Yaman, H., Kara, B. Y., e Ç. Tansel, B. (2007). The latest arrival hub location problem for cargo delivery systems with stopovers. *Transportation Research Part B: Methodological*, 41(8):906 – 919. ISSN 0191-2615.