Uma meta-heurística Greedy Randomized Adaptive Search Procedures (GRASP) para o problema p-hub de máxima cobertura não capacitado com alocação única

Warley Ferreira da Cunha

CEFET - MG

7 de fevereiro de 2024

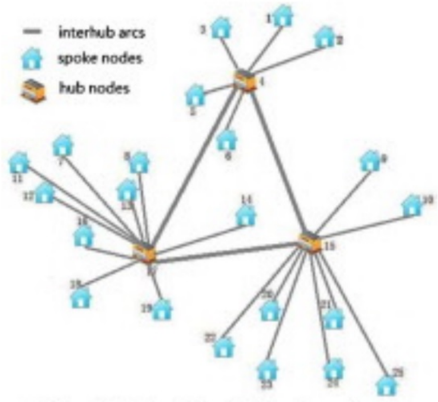
Introdução

O **problema** *p*-**hub de máxima cobertura não capacitado com alocação única** (**U**ncapacitated **S**ingle **A**llocation **p-H**ub **M**aximal **C**overing **P**roblem - **USApHMCP)** consiste em determinar p nós para serem hubs entre um conjunto de nós candidatos e alocar cada nó não hub a exatamente um único hub para maximizar a demanda total coberta entre pares de nós dentro de uma determinada distância.

Introdução

O **problema** p**-hub de máxima cobertura não capacitado com alocação única** (**U**ncapacitated **S**ingle **A**llocation **p-H**ub **M**aximal **C**overing **P**roblem - **USApHMCP)** consiste em determinar p nós para serem hubs entre um conjunto de nós candidatos e alocar cada nó não hub a exatamente um único hub para maximizar a demanda total coberta entre pares de nós dentro de uma determinada distância.

Neste trabalho, considera-se o caso em que todos os pares de hubs estão conectados e, a quantidade de hubs, denotada por p, é conhecida. Um nó não hub pode ser alocado a somente um único nó hub (alocação única) e os arcos e os hubs não possuem limite de capacidade (não capacitado).



Optimal hubs locations in 25 nodes network

Figura: Extraída de Yang et al,2013.

Caracterização do problema

$G = (N, A)$: grafo completo;

$N = \{1, 2, ..., i, ..., n\}$: conjunto de nós, $|N| = n$;

$A = \{(i, j)/i, j \in N\}$: conjunto de arcos;

H : conjunto de nós selecionados como hubs, $p = |H|$, H $\subset N$;

Caracterização do problema

$G = (N, A)$: grafo completo;

$N = \{1, 2, ..., i, ..., n\}$: conjunto de nós, $|N| = n$;

$A = \{(i, j)/i, j \in N\}$: conjunto de arcos;

H : conjunto de nós selecionados como hubs, $p = |H|$, H $\subset N$;

c_{ij} : custo unitário de transporte entre o nó i e o nó j,

$$d_{ij} = \gamma c_{ih_i} + \alpha c_{h_i h_j} + \delta c_{h_j j};$$

γ, α, δ : taxas unitárias para coleta (origem-hub), transferência (hub-hub) e distribuição (hub-destino), respectivamente. Espera-se $\alpha \leq \gamma; \alpha \leq \delta$.

Caracterização do problema

W_{ij} : fluxo do nó de origem i até o nó destino j.

Caracterização do problema

W_{ij} : fluxo do nó de origem i até o nó destino j.

O par (i, j) $O - D$ é coberto pelos hubs k e m se o custo de i para j via hubs k e m não excede o valor específico β_{ij} (custo de transporte que indica o nível de cobertura para o par $O - D$) ;

a_{ij}^{km} : parâmetro de cobertura,

$$a_{ij}^{km} = \begin{cases} 1, & \text{se} \quad \gamma c_{ik} + \alpha c_{km} + \delta c_{mj} \leq \beta_{ij}, \\ 0, & \text{caso contrário.} \end{cases}$$

Caracterização do problema

W_{ij} : fluxo do nó de origem i até o nó destino j.

O par (i, j) $O - D$ é coberto pelos hubs k e m se o custo de i para j via hubs k e m não excede o valor específico β_{ij} (custo de transporte que indica o nível de cobertura para o par $O - D$) ;

a_{ij}^{km} : parâmetro de cobertura,

$$a_{ij}^{km} = \begin{cases} 1, & \text{se} \quad \gamma c_{ik} + \alpha c_{km} + \delta c_{mj} \leq \beta_{ij}, \\ 0, & \text{caso contrário.} \end{cases}$$

λ_{ij} : parâmetro auxiliar,

$$\lambda_{ij} = max_{km} a_{ij}^{km}, \quad \forall i, j \in N.$$

Caracterização do problema

x_{ij} : variável binária de decisão,

$$x_{ij} = \begin{cases} 1, & \text{se o nó } i \text{ for atribuído ao hub } j, \\ 0, & \text{caso contrário.} \end{cases}$$

Caracterização do problema

x_{ij} : variável binária de decisão,

$$x_{ij} = \begin{cases} 1, & \text{se o nó } i \text{ for atribuído ao hub } j, \\ 0, & \text{caso contrário.} \end{cases}$$

Além disso,

$$x_{ii} = \begin{cases} 1, & \text{se nó } i \text{ for hub,} \\ 0, & \text{caso contrário.} \end{cases}$$

z_{ij} : fração do fluxo roteado do nó de origem i para o nó de destino j que é coberto.

Modelo Matemático - USApHMCP

$$max \sum_{i \in N} \sum_{j \in N} W_{ij} z_{ij} \tag{1}$$

suj. a

$$\sum_{k \in N} x_{kk} = p \tag{2}$$

$$\sum_{k \in N} x_{ik} = 1, \quad \forall i \in N. \tag{3}$$

$$x_{ik} \leq x_{kk}, \quad \forall i, k \in N. \tag{4}$$

$$z_{ij} \leq \sum_{k \in N} a_{ij}^{km} x_{ik} + \lambda_{ij}(1 - x_{jm}), \quad \forall i, j \in N, m \in N. \tag{5}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in N. \tag{6}$$

$$z_{ij} \geq 0, \quad \forall i, j \in N. \tag{7}$$

Representação da Solução

A **solução foi representada por dois vetores** (s, H).

O vetor $s = (s_1, ..., s_n)$ possui n elementos e contém as alocações de nó, em que s_i indica o hub ao qual o nó i está alocado.

O vetor H, de tamanho p, contém os hubs selecionados, em que $H = (h_1, ..., h_p)$.

$$s = (7, 7, 8, 9, 8, 9, 7, 8, 9, 9)$$
$$H = (7, 8, 9)$$

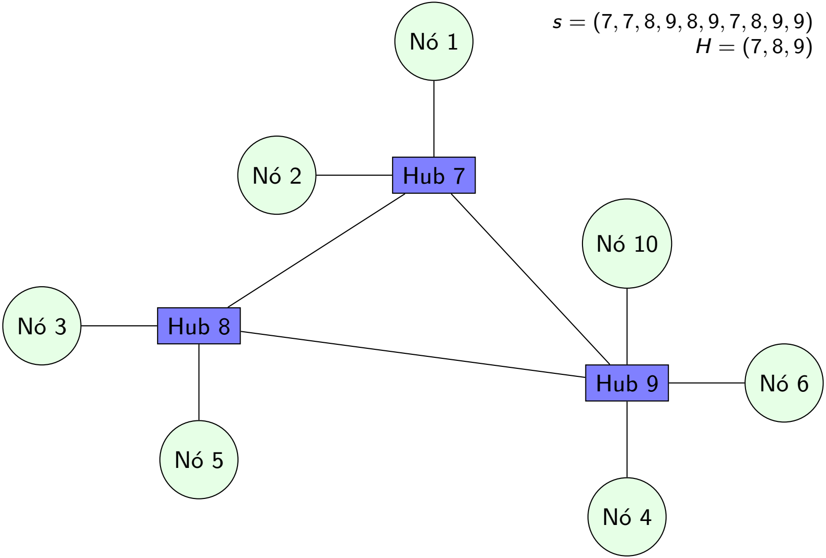


Figura: Representação de uma rede de hubs e não hubs e suas alocações.

Solução inicial

A solução inicial é construída de forma gulosa elemento a elemento. Para este trabalho do *USApHMCP*, **a construção inicial gulosa será feita escolhendo como hubs os p nós com maior fluxo** (que é dado pela soma das demandas originadas e as demandas destinadas ao nó).

Em seguida, é feita uma alocação inicial gulosa, **alocando cada nó não hub ao hub mais próximo**.

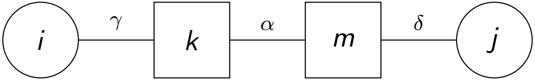


Figura: Rede de hubs e não hubs. Espera-se $\alpha \leq \gamma; \alpha \leq \delta$.

$s = (7, 7, 8, 9, 8, 9, 7, 8, 9, 9)$
$H = (7, 8, 9)$

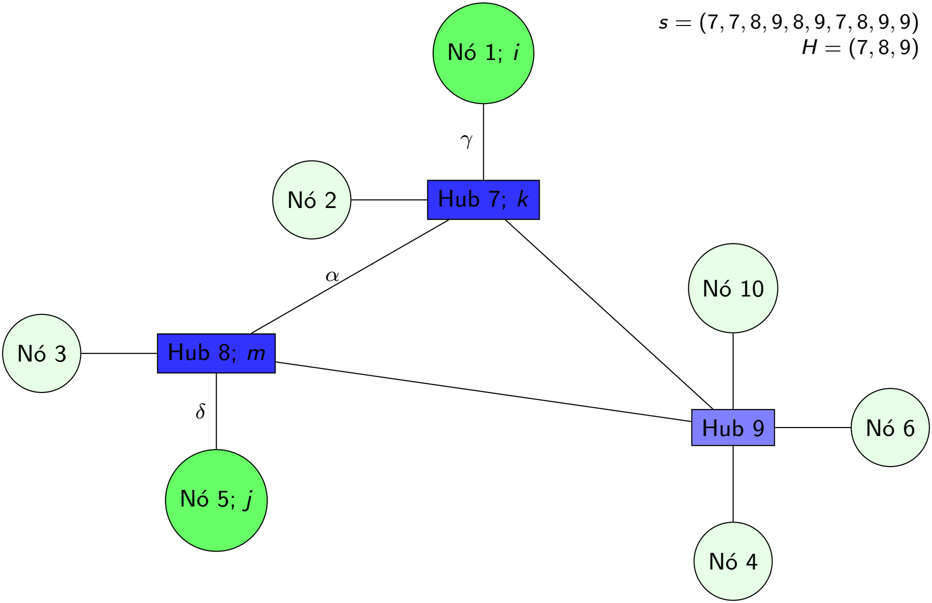


Figura: Representação de uma rede de hubs e não hubs e suas alocações.

Algorithm 1 Procedimento para calcular os parâmetros de cobertura a_{ij}^{km}

input: $N, \gamma, \alpha, \delta, \beta_{ij}$
output: todos os a_{ij}^{km}

for all $(i, j, k, m) \in N$ **do**
  if $\gamma C_{ik} + \alpha C_{km} + \delta C_{mj} \leq \beta_{ij}$ **then**
    $a_{ij}^{km} = 1$
  else
    $a_{ij}^{km} = 0$
  end if
end for

Busca local

A busca local utilizada será o método de **descida em vizinhança variável** *(Variable Neighborhood Descent - VND)*.

Proposto por Mladenovic e Hansen [1997] e explora o espaço de soluções por meio de trocas sistemáticas de estruturas de vizinhança, alterando gradativamente para vizinhanças mais distantes.

Movimento de troca de alocação

Um **movimento de troca de alocação** consiste em alocar um nó não hub a um outro nó hub e desalocá-lo do hub anterior.

Por exemplo, dados $N = \{1, 2, 3, 4, 5\}$, $H = (5, 4)$ e $s = (\mathbf{5}, 4, 5, 4, 5)$, uma mudança de alocação pode resultar no vetor de alocação $s_1 = (\mathbf{4}, 4, 5, 4, 5)$, em que o nó 1 deixou de ser alocado ao hub 5 para ser alocado ao hub 4.

Algorithm 2 Algoritmo de troca de alocação

Require: fo e (s, H), onde $s = (s_1, \ldots, s_n)$, $H = (h_1, \ldots, h_p)$

for $i \leftarrow 1$ até n **do**
  for $k \leftarrow 1$ até p **do**
    if $s_i \neq h_k$ **then**
      $\bar{s}_i \leftarrow h_k$
      $\bar{s}_j \leftarrow s_j, \forall\, j \neq i$
      $\bar{f} \leftarrow f(\bar{s})$
      if $\bar{f} > fo$ **then**
        $fo \leftarrow \bar{f}$
        $s \leftarrow \bar{s}$
      end if
    end if
  end for
end for
Retorne (s, H)

Movimento de troca de hub

Um **movimento de troca de hub** consiste em fechar um hub aberto e abrir um hub que estava previamente fechado.

Dada uma lista de hubs H, um vizinho da solução corrente possui lista de hubs H_1, em que H_1 corresponde a troca de um hub de H por um nó não hub do conjunto de nós N. Por exemplo, dados $N = \{1, 2, 3, 4, 5\}$ e $H = (\mathbf{5}, 4)$, ao retirar o hub 5 de H e inserir o hub 1, temos $H_1 = (\mathbf{1}, 4)$.

Algorithm 3 Algoritmo de Troca de Hub

Require: fo e (s, H), onde $s = (s_1, \ldots, s_n)$, $H = (h_1, \ldots, h_p)$

for $h \in H$ **do**
  for $i \in N \setminus \{h_1, \ldots, h_p\}$ **do**
    $\bar{H} \leftarrow$ Retire h de H e acrescente i em H
    $\bar{s} \leftarrow$ Alocação gulosa para $\bar{H}$
    $\bar{f} \leftarrow f(\bar{s})$
    if $\bar{f} > fo$ **then**
      $fo \leftarrow \bar{f}$
      $(s, H) \leftarrow (\bar{s}, \bar{H})$
    end if
    $\bar{s} \leftarrow$ Aplicação de movimentos de alocação na solução $(\bar{s}, \bar{H})$
    $\bar{f} \leftarrow f(\bar{s})$
    if $\bar{f} > fo$ **then**
      $fo \leftarrow \bar{f}$
      $(s, H) \leftarrow (\bar{s}, \bar{H})$
    end if
  end for
end for
Retorne (s, H)

Estratégia de busca local

Utilizamos a estratégia de busca de **melhor vizinho** (*Best Improvement*), que move para o melhor vizinho de melhora encontrado, após percorrer todos os vizinhos de uma determinada solução.

Seja (s, H) uma solução, enquanto esta solução não for um ótimo local com relação às duas estruturas de vizinhança, são feitos movimentos de troca de alocação e de troca de hubs. Se estes movimentos melhorarem a solução, a melhor solução encontrada e o valor de função objetivo são atualizados.

Algorithm 4 Algoritmo de Busca Local

Require: Solução (s, H)

while (s, H) não for um ótimo local para as duas estruturas de vizinhanças **do**
  $(\bar{s}, \bar{H}) \leftarrow$ movimento de alocação a partir de (s, H)
  if $f(\bar{s}) > f(s)$ **then**
    $(s, H) \leftarrow (\bar{s}, \bar{H})$
    $fo \leftarrow f(\bar{s})$
  end if
  $(\bar{s}, \bar{H}) \leftarrow$ movimento de hub a partir de (s, H)
  if $f(\bar{s}) > f(s)$ **then**
    $(s, H) \leftarrow (\bar{s}, \bar{H})$
    $fo \leftarrow f(\bar{s})$
  end if
end while
Retorne (s, H)

Greedy Randomized Adaptive Search Procedures (GRASP)

A meta-heurística *GRASP* é um processo iterativo para problemas combinatórios, em que cada iteração consiste basicamente em duas fases: **construção e busca local**.

O pseudocódigo representado pelo Algoritmo 5, onde $\alpha \in [0, 1]$ é um parâmetro do método e descreve a fase de construção do GRASP. Observamos que o parâmetro α controla o nível de gulosidade e aleatoriedade do procedimento *Construção*. Um valor $\alpha = 0$ faz gerar soluções puramente gulosas, enquanto $\alpha = 1$ faz produzir soluções totalmente aleatórias.

Algorithm 5 Procedimento *Construção*$(g(\cdot), \alpha, s)$

```
s ← ∅;
Inicialize o conjunto LC de candidatos;
while LC ≠ ∅ do
    g(t_min) = min{g(t) | t ∈ LC};
    g(t_max) = max{g(t) | t ∈ LC};
    LCR = {t ∈ C | g(t) ≥ g(t_max) − α(g(t_max) − g(t_min))};
    Selecione aleatoriamente um elemento t ∈ LCR;
    s ← s ∪ {t};
    Atualize o conjunto LC de candidatos;
end while;
Retorne s;
fim Construção;
```

Algorithm 6 Procedimento *BuscaLocal* $(f(\cdot), N(\cdot), s)$

$V = \{s' \in N(s) \mid f(s') > f(s)\}$;
while $|V| > 0$ **do**
    Selecione $s' \in V$;
    $s \leftarrow s'$;
    $V = \{s' \in N(s) \mid f(s') > f(s)\}$;
end while;
Retorne s;
fim *BuscaLocal*;

Algorithm 7 Procedimento *GRASP* $(f(\cdot),\ g(\cdot),\ N(\cdot),\ GRASP_{\text{max}},\ \alpha,\ s)$

$f^* \leftarrow \infty$;
for Iter $= 1$ até $GRASP_{\text{max}}$ **do**
  Construção$(g(\cdot),\ \alpha,\ s)$;
  BuscaLocal$(f(\cdot),\ N(\cdot),\ s)$;
  if $f(s) > f^*$ **then**
    $s^* \leftarrow s$;
    $f^* \leftarrow f(s)$;
  end if
end for
$s \leftarrow s^*$;
Retorne s;
fim *GRASP*;

Greedy Randomized Adaptive Search Procedures (GRASP)

O Algoritmo 6 descreve o pseudocódigo do procedimento de *Refinamento* para o nosso problema de maximização. A busca local explora as estruturas de vizinhanças através dos movimentos de troca de alocação e troca de hub, alocando nó não hub ao hub mais próximo.

O pseudocódigo representado pelo Algoritmo 7 ilustra o procedimento GRASP utilizado. Este algoritmo tem, como entrada, apenas dois parâmetros, a saber: α (fator de aleatoriedade/gulosidade da fase de construção) e $GRASP_{max}$ (número de iterações em que o método é aplicado).

Neste trabalho, utilizamos $\alpha = 0,6$ e $GRASP_{max} = 30$.

Algorithm 8 Procedimento aplicado ao USApHMCP

input:Instâncias APs, α, β_{ij}

1: Leia os dados de entrada.
2: Calcular os parâmetros a_{ij}^{km} de acordo com o (Algorithm 1).
3: Para cada nó $i \in N$, calcule o fluxo total com destino (D_i) e com origem (O_i) no nó i: $O_i = \sum_{j \in N} W_{ij}$ e $D_i = \sum_{j \in N} W_{ji}$.
4: Ordene os nós $i \in N$ em ordem decrescente de $(O_i + D_i)$.
5: Selecione os p primeiros nós com maior fluxo de entrada e saída como hubs.
6: Gere uma solução inicial usando alocação por menor distância e calcule o valor de função objetivo.
7: Busca local (Algorithm 2, Algorithm 3 - Algorithm 4).
8: Meta-heurística GRASP (Algorithm 5, Algorithm 6 - Algorithm 7).

Resultados dos Testes Computacionais

Todos os experimentos computacionais deste trabalho foram realizados em um notebook com um sistema operacional Windows 11 com processador 11^a Geração do Intel(R) Core(TM) $i7-1165G7$ 2.80 GHz e 16 GB de memória RAM.

Para a obtenção dos resultados computacionais foi utilizado um conjunto de dados conhecido na literatura como *Australian Post* (AP), com $n = 10, 20, 25, 40$ nós, $p = 3$ e 5, $\gamma = \delta = 1$ e $\alpha = 0,75$.

Durante a execução dos algoritmos nenhum outro processo foi executado paralelamente na máquina.

Resultados dos Testes Computacionais

Resultados para $\beta_{ij} = 2609$.

Instâncias	*fo CPLEX*	t_{CPLEX}	*fo Heur*	t_{Heur}	*GAP*
AP10$_3$L	477,66	0,03	362,146	0,12	31,89
AP20$_3$L	247,689	0,06	188,287	0,19	31,54
AP25$_3$L	352,841	0,09	247,277	0,20	42,69
AP40$_3$L	302,05	0,19	208,31	0,21	45

Resultados dos Testes Computacionais

Resultados para $\beta_{ij} = 25095$.

Instâncias	*fo CPLEX*	t_{CPLEX}	*fo Heur*	t_{Heur}	*GAP*
AP10$_3$L	3031,81	0,23	2638.61	0.069	14,9
AP20$_3$L	2915,95	3,3	2366.22	0,826	23,23
AP25$_3$L	2829,16	8,45	2555.25	1,479	10,71
AP25$_5$L	3190,86	9	3236.75	2.517	-1,41

Utilização de Algoritmos Genéticos para o USApHMCP

Um Algoritmo Genético (AG) é uma técnica evolucionária, isto é, utiliza analogias aos processos de evolução dos seres vivos para gerar e melhorar soluções para um determinado problema.

O AG é um método iterativo e adaptativo de busca global, com boa capacidade para fugir de ótimos locais.

Trabalha com uma população de indivíduos, onde **cada indivíduo representa uma possível solução para o problema** e, a cada iteração, gera uma nova população através de operações genéticas com **cruzamento, mutação e seleção natural**, buscando gerar filhos melhores que as soluções-pai.

Utilização de Algoritmos Genéticos para o USApHMCP

Cada indivíduo é avaliado conforme uma **função de aptidão (fitness)**, que mede a qualidade da solução que este indivíduo representa, e os que têm melhor fitness têm mais chances de serem passados para a próxima população.

O método é probabilístico, de forma que os indivíduos melhores avaliados tenham maior probabilidade tanto de serem submetidos às operações genéticas quanto de sobreviverem para a próxima população.

AG	USApHMCP
Indivíduo	Solução do problema.
População	É o conjunto de soluções.
Cromossomo	Representação de uma solução. 0 0 0 0 1 1 0 0 0 0 5 6 5 6 5 6 5 6 5 5
Gene	Parte da representação da solução.
Alelo	Valor que um gene pode assumir.
Crossover	Operadores de busca. Recombinação de genes para geração de nova população.
Mutação	Operadores de busca. Forma de sair de uma solução ótima local. Mudanças aleatórias nos hubs/spokes.
Aptidão "fitness"	Forma de medição de capacidade dos indivíduos. $f_{apt}(x) = f_o(x)$.

Tabela: Relação entre AG e o USApHMCP

Algorithm 9 Descrição do AG

1: Crie uma população inicial aleatória de indivíduos (ou cromossomos) com tamanho N_p.
2: Aplique a função de avaliação (fitness) a cada indivíduo, obtendo o valor da função objetivo para cada um.
Enquanto (critério de parada não foi satisfeito) **faça**
3: Gere novos indivíduos a partir da população atual, através de operadores evolutivos como: crossover, mutação e elitismo.
4: Aplique a função de avaliação aos novos cromossomos.
5: Selecione os mais aptos, ou seja, os que mais melhoram a função objetivo, e substitua os indivíduos de menor aptidão na população.
Fim-enquanto.
6: **Retorne** a solução mais apta.

Representação da solução

Visando representar computacionalmente a solução (ou, indivíduo do AG) para o USApHMCP, foi determinado **dois vetores de tamanho** n: um para armazenar a localização dos hubs e outro para armazenar as alocações dos nós não hubs (spokes) aos hubs.

Nestes vetores, cada posição corresponde a um nó da rede. O primeiro corresponde a um vetor binário em que cada posição armazena o valor 0, no caso do nó correspondente a esta posição ser um spoke, ou 1, no caso deste nó ser um hub. Já o outro vetor armazena o índice do hub, ao qual o nó correspondente está associado.

0	0	0	0	1	1	0	0	0	0
5	6	5	6	5	6	5	6	5	5

Tabela: Representação do cromossomo para o USApHMCP

Construção da Solução Inicial

O procedimento de construção da solução inicial para o USApHMCP foi pensando de maneira a possibilitar avaliar o percentual de cobertura de dois nós como hubs. Isto é, avaliamos todas as possíveis combinações de pares como hubs, e a partir disso optamos pela combinação que apresenta maior cobertura.

Como exemplo, utilizando a Tabela 1, após o procedimento que analisou todas as combinações, o resultado foi a seleção dos nós 5 e 6 como hubs, por proporcionarem a maior cobertura possível para dois nós.

Construção da Solução Inicial

No entanto, para instâncias que possuem $p > 2$, após encontrar uma solução com dois hubs, os demais hubs vão sendo incorporados à solução corrente de forma a maximizar a cobertura do grafo como um todo.

Isto é, após a inserção de dois hubs, o algoritmo vai adicionando, um por um, até que o número de p seja satisfeito.

O critério de incorporação de um nó como hub respeita a proposta anterior, isto é, o nó que possibilita o aumento mais expressivo no percentual de cobertura da rede é adicionado. Uma vez que se tem uma solução com o total de p de hubs, a solução inicial está definida.

Função de aptidão

O nosso problema é de maximização. Portanto, para avaliação de aptidão dos indivíduos utilizaremos a própria função de avaliação do USApHMCP, ou seja:

$$\max \sum_{i \in N} \sum_{j \in N} W_{ij} a_{ij}^{S_i S_j}$$

Operadores de Busca: Crossover

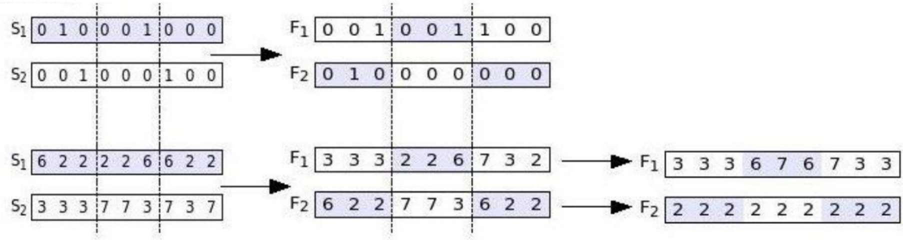
O método de cruzamento (crossover) escolhido para este trabalho foi o de dois pontos de corte.

O operador de crossover consiste no sorteio de dois pontos de cruzamento e na troca dos materiais genéticos. Os genes que estão entre os dois pontos de corte do primeiro cromossomo, unem-se com os genes que estão antes e depois do primeiro e segundo ponto de cruzamento, no outro cromossomo.

Isto é realizado tanto para o vetor de hubs quanto para o vetor de alocações dos nós aos hubs.

Operadores de Busca: Crossover

Após a troca dos genes, um operador se responsabiliza em verificar se o nó *i*, no vetor que armazena as alocações dos nós não hubs aos hubs, está associado a um hub válido do vetor que armazena os hubs, caso contrário, este nó deve ser realocado ao seu hub mais próximo.



Operadores de Busca: Mutação

O operador de mutação proposto neste trabalho é composto por duas fases, denominadas *shift* e *exchange*.

O *shift* sorteia um nó não hub (spoke) e o associa a outro hub; se o cromossomo possuir apenas um hub, essa função não se aplica.

A função *exchange* seleciona dois spokes aleatoriamente e troca suas associações. Um pré-requisito para esse operador é a existência de pelo menos dois hubs e nós não hubs. No caso de haver apenas um hub, ou um único spoke, esse método não é executado.

Trabalhos futuros

- Rever a implementação visando melhorar o GAP;
- Testar outras instâncias da literatura;
- Calibrar parâmetros com o IRACE;
- Implementar o AG no USApHMCP.

Referências

- Silva, Fernando; Sá, Elisângela M.; Souza, Sérgio R. *Uma meta-heurística Iterated Local Search para o problema p-hub de máxima cobertura não capacitado com alocação única*. In: Anais do Simpósio Brasileiro de Pesquisa Operacional, 2022, Juiz de Fora.

- Silva, M. R. e Cunha, C. B. *A tabu search heuristic for the uncapacitated single allocation p-hub maximal covering problem*. European Journal of Operational Research, (2017), 262 : 954–965

- Silva, Jardell F.; Martins, Flávio V. C.; Silva, Maria A.L.; Souza, Sérgio R.; *Algoritmo Genético Aplicado à Solução do Problema p-hub Centro Não Capacitado de Múltiplas Alocações*. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2019 . p. $856 - 867.ISSN2763 - 9061$.

Referências

- Almeida, Wesley G., Senne, Edson L. F., Yanasse, Horacio H. *Abordagens meta-heurísticas para o problema de localização de concentradores com restrições de capacidade*. Anais [...] São José dos Campos: INPE, 2010. On-line. IBI:$8JMKD3MGP8W/38BJHQ5$.Disponível em:$http://urlib.net/ibi/8JMKD3MGP8W/38BJHQ5$.

- Brás, Glender; Aguiar, Edgar L.; *Utilização de Algoritmos Genéticos para Resolução de Problemas de Localização de Concentradores*. Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, 2019.