Otimização Aplicada ao Planejamento de Anéis Unidirecionais em Telecomunicações

Patricia R. A. Lopes [1], Marcia Fampa [2], Silvio Binato [3]

Resumo. É grande o número de problemas de otimização de origem no setor de telecomunicações. Neste trabalho iremos otimizar um problema de alocação de equipamentos que visa satisfazer demandas a um custo mínimo usando anéis unidirecionais e equipamentos Add Drop Mutiplexer (ADM) em uma rede de telecomunicações. Para resolução deste problema consideraremos a metaheuristica GRASP.
Palavra chave: Metaheurística GRASP

1. Introdução

Por Telecomunicações entende-se um *conjunto de dispositivos e técnicas para a transmissão de informações instantâneas a longa distância*. Essa transmissão pode ser de voz, sinais gráficos, dados, imagens ou sinais de televisão. Todos eles têm os mesmos princípios fundamentais, mas diferem na forma de manipular as informações e nos meios utilizados para transmití-las.

Problemas de otimização são abundantes na indústria de telecomunicações. Um problema de otimização combinatória pode ser definido como:

$$(P) \quad \text{Min } f(x), \quad x \in X, \tag{1.1}$$

onde $f : X \rightarrow \Re$ é uma função real e X é um conjunto finito de soluções viáveis. Objetiva-se encontrar uma solução ótima para (P), $x^* \in X$, tal que $f(x^*) \leq f(x)$, $\forall x \in X$. Definimos x^* como uma solução ótima global de (P).

Teoricamente é possível encontrar uma solução ótima para (P) utilizando enumeração, porém na prática a estratégia de enumeração completa se torna inviável, pois o número de combinações cresce exponencialmente com o tamanho do problema.

Alguns problemas podem ser muito grandes para o uso de algoritmos exatos, por exemplo alguns problemas ligados ao setor de telecomunicações [4]. A estes problemas usualmente emprega-se métodos heurísticos.

Neste trabalho propomos a metaheurística GRASP para resolver o problema de planejamento da expansão de uma rede telecomunicações usando anéis unidirecionais (PAURT). Este problema consiste em determinar onde devem ser instalados equipamentos de transmissão de forma que seja atendida toda a demanda da rede a um custo mínimo.

[1]patricia@posgrad.nce.ufrj.br; IM, Núcle de Computação Eletrônica, UFRJ
[2]orientadora, fampa@cos.ufrj.br, IM, COPPE, UFRJ
[3]co-orientador, silvio@binato.com.br, PSR Consultoria

Prodedimento `GRASP` (`MaxIter, Sem,` α)

`Leia Dados de Entrada ();`
Para $k = 1$, `...,MaxIter` **faça**
Início
$\hat{x} \leftarrow$ `Construção Randômica ( Sem,`α`);`
$\tilde{x} \leftarrow$ `Busca Local (`$\hat{x}$`);`
$\bar{x} \leftarrow$ `Atualização da Solução(`$\tilde{x}$`);`
Fim;
Retorna $\bar{x}$

Fim `GRASP`;

Figura 1: pseudo código GRASP

2. GRASP, conceitos básicos

A metaheurística é uma abordagem *multi-start* iterativa, onde cada iteração GRASP é composta de duas fases: construção e busca local.

A melhor solução achada em todas as iterações é exibida no resultado final. Ilustramos o pseudo código GRASP na figura 1, onde `MaxIter` é o número máximo de iterações GRASP, `Sem` é a semente inicial para o gerador de números pseudo aleatórios e α é um parâmetro dado.

Em cada iteração, utiliza-se uma função gulosa característica do problema, que mede o benefício de selecionar cada elemento e constroi-se uma lista ordenada com todos os elementos que podem ser adicionados à solução, denominada Lista Restrita de Candidatos (RCL). Em geral há inúmeras possibilidades para a função gulosa, portanto, a escolha adequada desta função é muito importante tendo em vista que os resultados do GRASP são extremamente influenciados por esta escolha.

A heurística é considerada adaptativa, pois os benefícios associados a cada elemento são atualizados em cada iteração da fase de construção para refletir as mudanças trazidas pela seleção do elemento anterior. Este componente é responsável pela fuga dos ótimos locais, uma vez que diferentes RCLs são geradas em cada passo da fase de construção do GRASP.

O componente probabilístico do GRASP é caracterizado pela escolha aleatória de um dos candidatos da RCL, não necessariamente o melhor candidato, diferentemente de um método puramente guloso, na qual o melhor elemento é incluído na solução. Observa-se que a qualidade média da solução depende da qualidade dos membros da RCL e que a diversidade da fase de construção depende diretamente da cardinalidade da mesma. A RCL também é responsável pela fuga dos ótimos locais, pois dependendo da escolha aleatória, elementos diferentes serão incluídos na solução.

Para controlar a qualidade dos elementos da RCL, considere h_{min} e h_{max} respectivamente o melhor e o pior valor da função gulosa na fase de construção e α assumindo valores entre zero e um. A RCL é definida em função do pârametro α, compreendendo todos aqueles elementos que ainda não fazem parte da solução, para os quais o valor da função gulosa está no intervalo $[h_{min}, h_{min} + \alpha(h_{max} - h_{min})]$. Observa-se que para um problema de minimização $\alpha = 0$ implica em uma escolha puramente gulosa enquanto $\alpha = 1$, implica em uma escolha aleatória, isto é, para $\alpha = 1$, a RCL contém todos os elementos possíveis.

Podemos concluir que α é um dos principais parâmetros a serem sintonizados no GRASP e este pode ser fixo, aleatório ou auto-ajustável (reativo)[3].

Na fase de construção do *GRASP* básico o próximo elemento a ser inserido na solução é selecionado aleatoriamente entre os elementos candidatos na LRC. Cada elemento da LRC possui a mesma probabilidade de ser selecionado. Entretanto, toda a distribuição de probabilidade pode ser usada atribuindo uma maior probabilidade para seleção de determinados candidatos.

Bresina propõe em [1] uma seleção do elemento da LRC, onde a cada elemento da lista de canditatos restrita é atribuída uma probabilidade de seleção.

Diversas funções tendência são introduzidas, como logaritmica ($bias(r) = 1/log(r+1)$), linear ($bias(r) = 1/r$), polinomial ($bias(r) = 1/r^n$), exponencial ($bias(r) = 1/e^r$), aleatória ($bias(r) = 1$).

O valor da função gulosa de cada elemento $c \in C$ é calculado, uma vez que é necessário eleger os membros de C que farão parte da lista LRC. Daí, os elementos da LRC devem ser ordenados de acordo com o valor da função gulosa. Seja $rank(c), c \in C$ a posição relativa do elemento c na lista LRC de acordo com a ordenação dada pela função gulosa. A cada elemento da LRC é associado um valor, $bias(rank(c))$, onde $bias()$ é calculado de acordo com a função de tendência. A probabilidade de seleção de um elemento c da lista de candidatos é então determinada por:

$$p(c) = \frac{bias(rank(c))}{\sum_{\sigma \in \mathrm{LRC}} bias(rank(\sigma))}, c \in \mathrm{LRC} \tag{2.1}$$

A fase de busca local a partir da solução fornecida pela fase da construção, usa um procedimento de enumeração e uma estrutura da vizinhança, onde toda a vizinhança da solução atual é explorada. Se uma solução melhor é visitada, o procedimento de busca local toma esta como a nova solução do problema. O procedimento de busca local termina quando nenhuma solução melhor é encontrada dentro da vizinhança da solução atual. Isto garante que a solução atual é um mínimo local dentro da estrutura de vizinhança usada.

3. O Problema de Planejamento de Anéis Unidirecionais em Redes de Telecomunicação

Consideraremos neste trabalho uma rede de telecomunicação que se distribui em cidades, centros ou regiões. Para cada par de pontos i e j desta rede, define-se um ou mais pacotes de dados d_{ij}, que deverão ser transmitidos de i para j.

Definimos ciclo suporte como um subconjunto de pontos da rede que se interligam por fibra ótica. Quando instalamos equipamentos de transmissão nos ciclos suportes, estes passam a ser definidos como anel. A transmissão de dados na rede considerada será implementada através de uma arquitetura de anel [**?**]. Neste tipo de arquitetura a transmissão de dados entre cada par de pontos da rede é realizada através de um anel que contém estes pontos. Uma vez identificado os ciclos suportes da rede, buscamos construir os anéis, isto é, instalar os equipamentos de transmissão, de forma a atender toda a demanda.

A transmissão de dados de um ponto i para um ponto j da rede será então realizada através de um dos anéis da rede, o qual deverá ter equipamentos de transmissão de dados instalados tanto no ponto que envia os dados, como no ponto que os recebe. Observamos que cada pacote de dados deverá ser inteiramente transmitido em um mesmo anel, não podendo ser subdividido.

Contudo um nó pode fazer parte de mais de um anel e conseqüentemente ter nele instalados, equipamentos diferentes, porém estes não se interligam. Em um mesmo anel, todos os equipamentos de transmissão instalados deverão ser do mesmo tipo.

Utilizaremos o equipamento de transmissão *ADD-DROP Multiplexer* (ADM). Este possui tipo, capacidade máxima de transmissão e custo. A capacidade máxima de transmissão do equipamento utilizado definirá a capacidade de transmissão do anel, a qual independe portanto, do número de equipamentos utilizados. Por outro lado, o custo do anel é dependente do número de equipamentos que foi nele instalado e será dado pela soma dos custos destes equipamentos.

Utilizaremos anéis unidirecionais os quais são compostos pelo o tráfego normal que é transmitido em uma direção em uma fibra e o tráfego de proteção que é transmitido no sentido oposto em outra fibra, garantindo assim que mesmo que haja um rompimento ao longo do caminho, os dados cheguem ao seu destino [5].

Apresentaremos um modelo de programação linear inteira mista para o problema de planejamento de anéis unidirecionais em rede de telecomunicações. Representaremos no modelo, a rede de transmissão de dados por um grafo $G(V, E)$, onde o conjunto de vértices V, representa os pontos da rede de transmissão (cidades, centros, etc.) e o conjunto de arcos E representa as ligações entre os pontos.

No modelo (P1) apresentado a seguir, consideramos todos os ciclos contidos no grafo G. Eles representam os ciclos suportes, ou seja, as possíveis estruturas para formação dos anéis por onde os dados serão transmitidos na rede considerada.

Sejam $C_1, ..., C_L$ todos os ciclos do grafo G [2], $C = \{1, 2, ..., L\}$ o conjunto dos índices correspondentes e $\mathcal{N}^l$ o conjunto de nós que pertencem ao ciclo C_l, onde $l \in C$.

Consideremos que p_k é o preço de cada equipamento utilizado no anel k, M é uma constante com valor suficientemente grande, d_{ij} é a demanda que passa do nó i para o nó j, cap_k é a capacidade do anel k, onde a capacidade do anel k é igual a capacidade do equipamento nele utilizado e NA^l é um limite superior para o número de anéis construidos sobre o ciclo C_l do problema. Este limite é dado pelo pior caso, onde cada anel atenderia apenas uma demanda, o que nos daria um limite superior para o número de anéis igual ao número de demandas. Porém como estamos trabalhando com dois tipos de equipamentos estas demandas poderão estar utilizando o equipamento 1 ou 2. Os anéis que utilizam o equipamento 1 serão caracterizados pelos índices 1 a $Ndem$ (número de demandas) e os anéis que utilizam o equipamento o equipamento 2 serão caracterizados pelos índices $Ndem + 1$ a $2Ndem$. NA^l é portanto igual ao número de demandas entre os pares de nós contidos no ciclo C_l vezes o número de tipos de ADM's. A variável binária e_{ikl} assumirá valor um se existe equipamento no nó i do anel k construido no ciclo l e zero caso contrário. A variável y_{ijkl} será um se d_{ij} é atendida pelo anel k construido no ciclo l e zero caso contrário.

..

$$
\begin{aligned}
&\text{Minimizar} \sum_{l \in \mathcal{C}} \sum_{i \in \mathcal{N}^l} \sum_{k=1}^{NA^l} p_k e_{ikl},\\
&\text{sujeito a:}\\
&\qquad \sum_{i,j \in \mathcal{N}^l, i \neq j} d_{ij} y_{ijkl} \leq cap_k, k = 1, \ldots, NA^l, l \in \mathcal{C},\\
&\qquad \sum_{j \in \mathcal{N}^l \setminus i | d_{ij} \neq 0} y_{ijkl} \leq M e_{ikl}, k = 1, \ldots, NA^l, i \in \mathcal{N}^l, l \in \mathcal{C},\\
&\qquad \sum_{i \in \mathcal{N}^l \setminus j | d_{ij} \neq 0} y_{ijkl} \leq M e_{jkl}, k = 1, \ldots, NA^l, j \in \mathcal{N}^l, l \in \mathcal{C},\\
&\qquad \sum_{l \in \mathcal{C}} \sum_{k=1}^{NA^l} y_{ijkl} = 1, i, j \in \mathcal{N}^{\hat{l}} | i \neq j | d_{ij} \neq 0,\\
&\qquad e_{ikl}, y_{ijkl} \in \{0,1\}, i, j \in \mathcal{N}^l | i \neq j, k = 1, \ldots, NA^l, l \in \mathcal{C}.
\end{aligned}
\tag{3.1}
$$

A função objetivo representa a minimização do custo total com a instalação dos equipamentos considerando todos os nós, todos os anéis e todos os ciclos. A primeira restrição, estabelece que a demanda total atendida no anel k construido no ciclo l, não deve ser maior que sua capacidade. A segunda restrição estabelece que não haverá fluxo partindo de i, no anel k construido no ciclo l, se não houver equipamento nele instalado, ou seja, se $e_{ikl} = 0$. Observe que M deve ser grande o suficiente para tornar esta restrição redundante quando $e_{ikl} = 1$. Desta forma, utilizamos $M = n$, onde n é o número de nós da rede. A terceira restrição, estabelece que não haverá fluxo chegando no nó j, no anel k construido no ciclo l, se não houver equipamento nele instalado. A quarta restrição, estabelece que a demanda d_{ij} será atendida por exatamente um anel construído em um dos ciclos.

Observe que o número dos ciclos em um dado grafo cresce exponencialmente com seu número de nós e o número de variáveis binárias pode se tornar muito grande. Como exemplo, consideremos a rede (G) ilustrada na figura 2, onde existem três ciclos (l_1, l_2 e l_3).

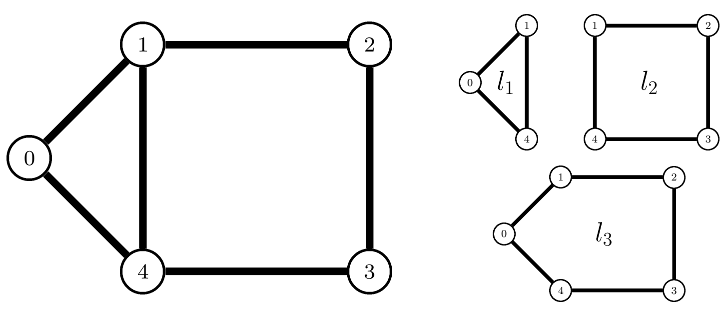

Figura 2: Grafo e Ciclo

Consideremos 2 tipos de ADM's e quatro demandas que deverão ser atendidas. Então o número de variáveis binárias será 600, e o número de restrições será 236. Se tomarmos 5 ADM's, o número de variáveis binárias passará para 1500 e o número de restrições será 560.

Procedimento GRASP-PAURT (MaxIter, Sem1,Sem2, α_1, α_2)

Leia Dados de Entrada ();
Para $k = 1, \ldots,$ MaxIter **faça**
Início
  Enquanto todas as demandas não forem atendidas **faça**
  Início
   $C_{\hat{l}} \leftarrow$ Sub-fase de Construção Ciclo (Sem1,α_1);
   $\hat{x} \leftarrow$ Sub-fase de Construção Anéis (Sem2,α_2, $C_{\hat{l}}$);
  Fim
  $\tilde{x} \leftarrow$ Busca Local ($\hat{x}$);
  $\bar{x} \leftarrow$ Atualização da Solução($\tilde{x}$);
Fim;
Retorna $\bar{x}$

Fim GRASP-PAURT;

Figura 3: pseudo código GRASP-PAURT

Concluimos que a utilização de um método de otimização combinatória clássico, como por exemplo *branch and bound*, tornaria muito difícil a resolução de problemas reais, levando-nos a optar por uma metaheurística que nos forneça uma solução próxima a exata porém com um tempo computacional razoável.

3.1. Metaheurística GRASP-PAURT

Apresentaremos a seguir os procedimentos de construção e de busca local (GRASP) implementados para obter-se uma solução para o problemas PAURT.

Ao aplicarmos o GRASP ao PAURT, a fase de construção da metaheurística consiste basicamente na geração de anéis e alocação dos equipamentos para a transmissão de dados da rede. O procedimento de geração dos anéis é dividido em duas etapas (duas sub-fases). A primeira consiste em selecionar ciclos suporte (sub-fase de construção Ciclo) para os anéis e a segunda consiste em definir os anéis (sub-fase de construção Anéis), ou seja, estabelecer os equipamentos de transmissão que serão instalados nos nós da rede em cada um dos ciclos suporte selecionados, de forma a atender a demanda entre cada par de nós pertencentes ao ciclo.

A seguir detalharemos os principais procedimentos do algoritmo, cujo pseudo-código pode ser visto na figura 3, onde MaxIter é o número de iterações do GRASP-PAURT, $Sem1$, $Sem2$ são as sementes que serão utilizadas nas sub-fases de construção, α_1 e α_2 são os parâmetros que determinarão o tamanho das RCLs em cada sub-fase de construção.

Após a leitura dos dados de entrada executa-se a fase de construção (sub-fase de construção Ciclo e sub-fase de construção Anéis) até que tenhamos atendido todas as demandas da rede, ou seja, até que tenhamos uma solução viável para o problema de planejamento de anéis unidirecionais em uma rede de telecomunicações. Em seguida executa-se a busca local procurando-se obter um ótimo local. A solução encontrada será guardada caso esta seja melhor do que as soluções obtidas anteriormente. Estes procedimentos (construção e busca local) serão repetidos MaxIter vezes.

O procedimento *Leia Dados de Entrada* consiste em ler os dados necessários para que seja possível executarmos o GRASP-PAURT. Estes dados, em essência, são o número de nós (n) da

Procedimento `Construção Ciclo` (`sem1`, α_1, Ft)

`Inicializa o conjunto de Candidatos` C ;
$h_{min} = min\ \{\, g(t) | t \in C \,\}$;
$h_{max} = max\ \{\, g(t) | t \in C \,\}$;
$LRC = \{s \in C | g(s) \geq h_{min} + \alpha_1 (h_{max} - h_{min})\}$;
$C_{\hat{l}} \leftarrow$ `Seleção` s `da LRC (sem1,` Ft`)`;
`Retorna` $C_{\hat{l}}$;

Fim `Construção Ciclo`;

Figura 4: pseudo código Construção Ciclo

rede, o grafo $G = (N, E)$ que representa a rede de telecomunicações, o conjunto de demandas $D = \{d_{ij} \mid i = 1, ..., n,\ j = 1, ..., n, i \neq j\}$ que deverão ser atendidas entre cada par de nós da rede, o conjunto de ciclos, $\{C_1, ..., C_L\}$ que o grafo G possui, gerados previamente, os tipos de equipamentos de transmissão disponíveis, bem como suas capacidades máximas de transmissão e seus custos.

A próxima etapa, já interna ao loop de iterações da heurística GRASP, é o procedimento construtivo que gera soluções viáveis para o problema PAURT. Nossa proposta de heurística consiste de um procedimento construtivo dividido em dois níveis, onde no primeiro nível faz-se a seleção de um ciclo suporte que será utilizado e no segundo nível a geração dos anéis, responsáveis pela transmissão dos dados pela rede de telecomunicações. O procedimento construtivo a dois níveis termina somente quando todas as demandas forem atendidas.

O primeiro nível da heurística construtiva (*Construção Ciclo*) é responsável pela seleção de um ciclo de suporte. A escolha natural para o ciclo do grafo da rede de transmissão é selecionar o ciclo que suporta o maior número de demandas, ou seja, o ciclo que contém o maior número de nós com demandas entre eles. Esta escolha é dita natural pois ela permite a seleção de anéis com maior capacidade, e menor custo unitário. Logo, escolher ciclos do grafo que suportam o maior número de demandas ainda não atendidas tende a minimizar o custo final de expansão.

Esta medida, o número de demandas não atendidas pertencentes a cada um dos ciclos suportes, será utilizada como a função "gulosa" ($h_1(.)$) do procedimento construtivo do primeiro nível da fase de construção do GRASP-PAURT.

A lista restrita de candidatos (RCL) do primeiro nível é definida a partir do parâmetro α_1 $(0 \leq \alpha_1 \leq 1)$ e dos valores máximos e mínimos, respectivamente h_{max} e h_{min}, da função "gulosa", ou seja, pertencem a lista restrita de candidatos todos os ciclos cujo número de demandas em seus nós satisfaz $h_{max} \geq h(C_l) \geq h_{max} - \alpha_1 \, (h_{max} - h_{min})$.

A seleção do ciclo suporte é determinística quando assumimos $\alpha_1 = 0$, isto é, garantimos que a RCL conterá somente o ciclo que atende o maior número de demandas. Por outro lado, fazendo $\alpha_1 = 1$ obtemos uma lista restrita de candidatos com todos os elementos possíveis, isto é, todos os ciclos que suportam algum par de demandas, então diz-se que esta escolha para o parâmetro α_1 torna o procedimento construtivo de primeiro nível puramente aleatório.

O próximo nível da fase de construção é o procedimento *Construção Anéis*, cujo objetivo é a alocação de equipamentos de transmissão de dados (ADM's) para a formação de anéis sobre o ciclo suporte $C_{\hat{l}}$ selecionado no primeiro nível da heurística construtiva. Cada vez que executarmos este procedimento, $D_{\hat{l}} = \{d_{ij} \in C_{\hat{l}} | d_{ij} \neq 0\}$ será inicializado com o subconjunto das demandas associadas ao ciclo $C_{\hat{l}}$, selecionado na sub-fase de Construção Ciclo, que ainda

Procedimento `Construção Anéis` (`sem1`, α_1, $C_{\hat{l}}$, Ft)

$D_{\hat{l}} = \{d_{ij} \in C_{\hat{l}} | d_{ij} \neq 0\}$;
$E = \{e_{ik\hat{l}}$ `e` $(e_{ik\hat{l}}, e_{jk\hat{l}})$ $\forall$ $i,$ $j \in N^{\hat{l}}$ `e` $k = 1, .., NA^{\hat{l}}$ $\}$;
$\hat{X} = \emptyset$;
Enquanto $D_{\hat{l}} \neq \emptyset$ **faça**
Início
  Para `cada elemento t de` E **faça**
  Início;
   `Fixar variáveis` $e_{ik\hat{l}}$ `e` $y_{ijk\hat{l}}$ `considerando` $\hat{X}$ `e t`;
   `Resolver o problema` **??** ;
  Fim;
  $h_{min} = min$ $\{ \Delta Z(t) | t \in E$ `e` $\Delta Z(t) > 0\}$;
  $h_{max} = max$ $\{ \Delta Z(t) | t \in E$ `e` $\Delta Z(t) > 0\}$;
  $LRC = \{s \in E | \Delta Z(s) \geq h_{min} + \alpha_2 (h_{max} - h_{min})\}$;
  `Ordenar LRC segundo o valor de` $\Delta Z(t)$;
  `Selecione` s `da LRC utilizando a função tendência` ;
  $\hat{X} = \hat{X} \bigcup \{s\}$;
  $D_{\hat{l}} = D_{\hat{l}} - \{$ `demandas já atendidas` $\}$;
  `Atualização do conjunto` E ;
Fim;
`Retorna` $\hat{X}$;

Fim `Construção Anéis`;

Figura 5: pseudo código Construção Anéis

não foram atendidas pela solução em construção.

Quando fixamos o ciclo suporte no procedimento de construção Ciclo o problema 3.1 transforma-se no problema 3.2, onde um único ciclo suporte é considerado, o ciclo $C_{\hat{l}}$, e somente as demandas associados ao ciclo $C_{\hat{l}}$ são consideradas.

$$\text{Minimizar } Z = \sum_{i \in \mathcal{N}^{\hat{l}}} \sum_{k=1}^{NA^{\hat{l}}} p_k e_{ik\hat{l}},$$

$$\begin{aligned}
&\text{sujeito a:} \\
&\sum_{i,j \in \mathcal{N}^{\hat{l}}, i \neq j} d_{ij} y_{ijk\hat{l}} \leq cap_k, k = 1, \ldots, NA^{\hat{l}}, \\
&\sum_{j \in \mathcal{N}^{\hat{l}} \setminus i | d_{ij} \neq 0} y_{ijk\hat{l}} \leq M e_{ik\hat{l}}, k = 1, \ldots, NA^{\hat{l}}, i \in \mathcal{N}^{\hat{l}}, \\
&\sum_{i \in \mathcal{N}^{\hat{l}} \setminus j | d_{ij} \neq 0} y_{ijk\hat{l}} \leq M e_{jk\hat{l}}, k = 1, \ldots, NA^{\hat{l}}, j \in \mathcal{N}^{\hat{l}}, \\
&\sum_{k=1}^{NA^{\hat{l}}} y_{ijk\hat{l}} = 1, i, j \in \mathcal{N}^{\hat{l}}, i \neq j \mid d_{ij} \neq 0, \\
&e_{ik\hat{l}}, y_{ijk\hat{l}} \in \{0, 1\}, i, j \in \mathcal{N}^{\hat{l}} \mid i \neq j, k = 1, \ldots, NA^{\hat{l}},
\end{aligned} \tag{3.2}$$

Seja Ω o conjunto de candidatos a serem incorporados à solução. Inicialmente Ω é inicializado com o conjunto de todas as variáveis $e_{ik\hat{l}}$ e dos pares de variáveis ($e_{ik\hat{l}}$,$e_{jk\hat{l}}$) para todo i, $j \in N^{\hat{l}}$, $i \neq j$, e $k = 1,..,NA^{\hat{l}}$.

A seleção de um elemento de Ω ($e_{ik\hat{l}}$ ou ($e_{ik\hat{l}}, e_{jk\hat{l}}$)) representa a escolha desta variável (ou destas variáveis) para ser(em) fixada(s) em 1, ou seja, representa a escolha do nó i (ou dos nós i e j) para alocarmos nele(s) um ADM que pertencerá(ão) ao anel k.

Seja $\hat{X}$ o conjunto das variáveis $e_{ik\hat{l}}$ já fixadas até uma dada iteração da Sub-fase Construção Anéis.

Para decidir qual elemento de Ω será adicionado ao conjunto $\hat{X}$, propomos avaliar cada elemento de Ω pelo benefício que sua inclusão em $\hat{X}$ traz ao sistema. Este benefício pode ser calculado da seguinte forma: seja ω um elemento do conjunto Ω. A idéia para calcular este benefício é resolver o problema (3.2) para $\{\omega \cup \hat{X}\}$.

Para isto, seja $\bar{e}_{ik\hat{l}}$ tal que:

$$\bar{e}_{ik\hat{l}} = \begin{cases} 1 & \text{se } e_{ik\hat{l}} \in \{\omega \cup \hat{X}\} \\ 0 & \text{caso contrário.} \end{cases}$$

Substituindo $\bar{e}_{ik\hat{l}}$ no problema (3.2), obtemos:

$$\begin{aligned} &\text{Minimizar } Y = -W \sum_{k=1}^{NA^{\hat{l}}} \sum_{i=1}^{n} \sum_{j=1}^{n} y_{ijk\hat{l}} d_{ij}, \\ &\text{sujeito a:} \\ &\quad \sum_{i,j \in \mathcal{N}^{\hat{l}}, i \neq j} d_{ij} y_{ijk\hat{l}} \leq cap_k, k = 1, \ldots, NA^{\hat{l}}, \\ &\quad \sum_{j \in \mathcal{N}^{\hat{l}} \setminus i | d_{ij} \neq 0} y_{ijk\hat{l}} \leq M \bar{e}_{ik\hat{l}}, k = 1, \ldots, NA^{\hat{l}}, i \in \mathcal{N}^{\hat{l}}, \\ &\quad \sum_{i \in \mathcal{N}^{\hat{l}} \setminus j | d_{ij} \neq 0} y_{ijk\hat{l}} \leq M \bar{e}_{ik\hat{l}}, k = 1, \ldots, NA^{\hat{l}}, j \in \mathcal{N}^{\hat{l}}, \\ &\quad \sum_{k=1}^{NA^{\hat{l}}} y_{ijk\hat{l}} \leq 1, i, j \in \mathcal{N}^{\hat{l}}, i \neq j \mid d_{ij} \neq 0, \\ &\quad y_{ijk\hat{l}} \in \{0,1\}, i, j \in \mathcal{N}^{\hat{l}} \mid i \neq j, k = 1, \ldots, NA^{\hat{l}}. \end{aligned} \tag{3.3}$$

O parâmetro W deve ser grande o suficiente para que $Z(\hat{X} \bigcup \{t\})$ seja negativo sempre existir pelo menos uma demanda associada ao ciclo de suporte $C_{\hat{l}}$ ainda não atendida.

Desta forma, o benefício de incorporar ω à solução em construção $\hat{X}$ pode ser medido pela diferença

$$\Delta Z(t) = Z(\hat{X}) - Z(\hat{X} \cup \{t\}),$$

onde $Z(X)$ equivale a $\sum_{i \in \mathcal{N}^{\hat{l}}} \sum_{k=1}^{NA^{\hat{l}}} p_k e_{ik\hat{l}} - Y$, onde Y é o valor da solução ótima do problema (3.3).

O procedimento *Construção Anéis* tem por objetivo construir iterativamente uma solução viável, logo a cada iteração um elemento do conjunto Ω é incorporado ao conjunto $\hat{X}$ (que é inicialmente um conjunto vazio) até que $\hat{X}$ seja uma solução viável para o problema (3.2).

Observe que no problema (3.3) a quarta restrição foi modificada de uma restrição de igualdade para uma restrição de desigualdade (≤ 1). Isto se deve ao fato que no problema (3.3) não podemos mais garantir que todas as demandas pertencentes ao ciclo $C_{\hat{l}}$ sejam atendidas pois estamos fixando os nós que recebem equipamentos (fixamos as variáveis $e_{ik\hat{l}} = 1$ ou $e_{ik\hat{l}} = 0$). Para encorajar o atendimento as demandas, em (3.3), foi incluído um termo na função objetivo que penaliza o não atendimento as demandas.

Note ainda que podemos preprocessar este problema, reduzindo assim a complexidade de sua solução. Retirando as variáveis y que estão fixas em zero e, consequentemente, retirando também todas as restrições $\sum_{j\in\mathcal{N}^{\hat{l}}\setminus i|d_{ij}\neq 0} y_{ijk\hat{l}} \leq Me_{ik\hat{l}}$ e $\sum_{i\in\mathcal{N}^{\hat{l}}\setminus j|d_{ij}\neq 0} y_{ijk\hat{l}} \leq Me_{jk\hat{l}}$, pois elas têm como finalidade fixar em zero as variáveis $y_{ijk\hat{l}}$ quando $e_{ik\hat{l}}$ assumir valor zero. Estas restrições já foram atendidas, portanto, ao fixarmos um subconjunto das variáveis $y_{ijk\hat{l}}$ em zero, obtemos o problema reduzido representado a seguir:

$$
\begin{aligned}
\text{Maximizar } & \sum_{k=1}^{NA^{\hat{l}}} \sum_{i=1}^{n} \sum_{j=1}^{n} y_{ijk\hat{l}} d_{ij}, \\
\text{sujeito a:} & \\
& \sum_{i,j\in\mathcal{N}^{\hat{l}}, i\neq j} d_{ij} y_{ijk\hat{l}} \leq cap_k, k = 1, \ldots, NA^{\hat{l}}, \\
& \sum_{k=1}^{NA^{\hat{l}}} y_{ijk\hat{l}} \leq 1, i, j \in \mathcal{N}^{\hat{l}}, i \neq j \mid d_{ij} \neq 0, \\
& y_{ijk\hat{l}} \in \{0, 1\}, \text{se } e_{ik\hat{l}} = 1, i, j \in \mathcal{N}^{\hat{l}} \mid i \neq j, k = 1, \ldots, NA^{\hat{l}}, \\
& y_{ijk\hat{l}} = 0, \text{se } e_{ik\hat{l}} = 0, i, j \in \mathcal{N}^{\hat{l}} \mid i \neq j, k = 1, \ldots, NA^{\hat{l}}.
\end{aligned} \tag{3.4}
$$

Finalmente observe que a função objetivo do problema (3.4) modifica-se para maximizar a utilização do atendimento a demanda, dado uma proposta de solução $\hat{X}$. Maximizar esta função objetivo é equivalente a minimizar a função objetivo do problema (3.3) pois o termo de penalização é o incentivo para a maximização do atendimento a demanda.

Então se $\Delta Z(t) \leq 0$, ou seja, $Z(\hat{X} \cup \{t\}) \geq Z(\hat{X})$, podemos concluir que a incorporação de t ao conjunto $\hat{X}$ não possibilitou o atendimento de nenhuma demanda a mais. Sendo assim, o elemento t, tal que $\Delta Z(t) \leq 0$, não será considerado para a formação da lista restrita de candidatos, LRC. Observemos que quanto mais positiva for a diferença $\Delta Z(t)$, maior o benefício de incorporar os equipamentos, analisados no momento, a solução.

Definimos então a LRC para a sub-fase Construção Anéis, conforme apresentado na figura 5, onde h_{max} e h_{min} são o melhor e o pior valor da função gulosa para este procedimento, ou seja, o maior e o menor valor de ΔZ.

O parâmetro α_2 será responsável pelo tamanho da LRC, assumindo valor fixo ($0 \leq \alpha_2 \leq 1$).

Após selecionarmos um dos elementos da LRC para ser incorporado ao conjunto $\hat{X}$, o conjunto Ω será atualizado. Dele serão excluídos os elementos formados pelas variáveis que já estão fixas em 1 e os elementos formados por pares de variáveis, nos quais uma delas já está fixa em 1. O procedimento de exclusão é necessário pois não podemos criar novamente um anel já criado, como também nao podemos colocar novos equipamentos nos nos dos anéis criados, onde já exite equipamento.

```
Procedimento Busca Local (x̂)
    Preencher as estruturas tab-anel e L_d[k];
    Ordenar tab-anel pelo número de demandas (Nd);
    ind ← posição inicial de tab-anel;
    fim ← posição final de tab-anel;
    indf ← posição final de tab-anel;
    Enquanto ind < fim  faça
     Início
       Enquanto indf > ind ou L_d[k] ≠ ∅  faça
        Início
          Para cada d_ij ∈ L_d[k] faça
           Início
             procedimento Avalia-Troca (d_ij , L_d[k_indf] )
              Se Avalia-Troca = verdadeiro  faça
              Início
               Atualizar k, Sb, Pr, Nd e x̂
              Fim
           Fim
           indf ← indf − 1
         Fim
         ind ← ind + 1
     Fim
     Retorna (x̂);
Fim Busca Local;
```

Figura 6: pseudo código Busca Local

A cada iteração da sub-fase Construção Anéis adicionamos então à solução, um novo elemento que pode ser uma ou duas variáveis $e_{ik\hat{l}}$ fixas em 1. O procedimento será repetido até que seja obtida uma solução onde todas as demandas do ciclo $C_{\hat{l}}$ tenham sido atendidas.

Conclui-se que o procedimento *Construção Anéis* tem como objetivo a seleção dos melhores elementos (equipamentos), isto é, os que atendem as demandas com o menor custo possível. A seleção torna-se puramente gulosa quando assumimos $\alpha_2 = 1$, e caso tenhamos $\alpha_2 = 0$ obteremos uma LRC com todos os elementos possíveis.

Partindo de cada solução construída na fase de construção do GRASP, emprega-se um procedimento de pesquisa local na tentiva de melhorar a solução.

O procedimento de *Busca Local* que podemos ver na figura 6, descreve a fase de Busca Local do GRASP-PAURT.

Para cada tipo de anel (k) construído, calculamos sua capacidade livre (Sb), o preço do equipamento ADM (Pr), o número de demandas atendidas (Nd) e o ciclo suporte utilizado ($\hat{l}$) pelo mesmo. Esta estrutura estará ordenada crescentemente pelo número de demandas atendidas e a denominaremos *tab-anel*. Para cada anel definido em *tab-anel* estará associado uma lista de demandas (L_d) atendida pelo mesmo. Esta possui nó inicial (de onde partiu o fluxo), o nó final (destino do fluxo), o valor do fluxo e os nós que compõe o ciclo suporte utilizado.

Para atender o seu objetivo, o procedimento de busca local proposto se baseia em alocar a demanda de um anel para outro de forma a economizar na instalação de ADMs e eventualmente, eliminar algum anel da rede.

```
Procedimento Avalia-Troca (d_ij, L_d[k_indf], k_ind)
 Avalia-Troca = falso
 Se i e j pertencem ao ciclo l̂ faça
 Início
    Se d_ij ≤ Sb[k_indf] faça
    Início
       Se existe equipamento no nó i e no nó j faça
        Início
         Avalia-Troca = verdadeiro;
        Fim
       Se existe equipamento no nó i ou no nó j faça
        Início
          Se Pr[k_indf] < Pr[k_ind]  faça
          Início
          Avalia-Troca = verdadeiro;
          Fim
        Fim
       Se não existe equipamento no nó i nem no nó j faça
        Início
          Se Pr[k_indf] < Pr[k_ind]  faça
          Início
          Avalia-Troca = verdadeiro;
          Fim
        Fim
    Fim
 Fim
 Retorna (Avalia-Troca );
Fim Avalia - Troca;
```

Figura 7: pseudo código Avalia - Troca

Para isto, procuramos alocar as demandas pertencentes aos anéis que atendem ao menor número de demandas nas sobras de capacidade dos outros anéis da rede, com a intenção de otimizar a utilização da rede de transmissão. É interessante ressaltar que, definido que o procedimento de busca local troca uma demanda de um anel para outro estamos definindo o movimento na vizinhança que caracteriza o procedimento de busca local.

Quando avaliarmos os anéis com menor número de demandas (Nd_{min}) estaremos avaliando a estrutura *tab-anel* de cima para baixo, isto é, tentando alocar no anel com o maior número de demandas (Nd_{max}, última posição de *tab-anel*), as demanas atendidas no anel com o menor número de demandas avaliado no momento.

Avaliaremos se a demanda pertencente a $L_d[k]$, onde k é a posição com o Nd_{min}, poderá ser alocada no anel com Nd_{max}, caso contrário avaliaremos a próxima demanda pertencente a $L_d[k]$.

Cada vez que trocamos um elemento (demanda) atualiza-se o anel de onde saiu a demanda e o anel onde a demanda foi alocada, a capacidade livre (Sb), o preço do equipamento (Pr), o número de demandas atendidas (Nd), lista de demandas (L_d) atendida dos respectivos anéis. Consequentemente estaremos atualizando a solução $\hat{x}$.

Finalizamos quando todas as demandas pertencente a $L_d[k]$ tiverem sido trocadas para o anel com Nd_{max}. Caso contrário, faremos a mesma tentativa, com as demandas que ainda não foram transferidas, para o penúltimo anel e assim sucessivamente até que $L_d[k]$ esteja vazio ou não exista possibilidade de troca.

Na tentativa de melhorar cada solução construída, avaliaremos todas as possibilidades de retirarmos alguns dos equipamentos alocados (um de cada vez), isto é, a possibilidade de transferir alguma demanda (ou algumas demandas) para outro anel (anéis) de forma que possamos retirar algum(ns) equipamento sem que deixemos de atender todas as demandas da rede. Com isto teremos a possibilidade de eliminar algum(ns) anel (anéis).

Nesta tentativa analisamos todas as possibilidades de alocar algum(ns) equipamento (um ou dois) e a possibilidade de trocar a demanda sem a necessidade de instalarmos equipamentos, sem que haja aumento no valor da função objetivo do problema $P1$. Estes procedimentos se repetem até que não tenhamos a possibilidade de trocar mais nenhuma demanda.

Deste modo, para um problema de minimização, pesquisamos toda a vizinhança da solução x_p, construida na fase de construção, em busca de uma nova configuração x_{p+1}, tal que $f(x_{p+1}) < f(x_p)$. Se tal configuração existe torna-se a nova solução e o processo se repete para x_{p+1}, caso contrário, x_p é um ótimo local dentro de $N(x_p)$.

4. Conclusões

Neste trabalho apresentamos um novo procedimento heurístico para a resolução de problemas de planejamento da expansão de redes de telecomunições utilizando anéis unidirecionais. Este novo procedimento heurístico consiste na adaptação dos conceitos da metaheurística GRASP para a resolução deste problema. É importante ressaltar neste processo, uma contribuição realizada por este trabalho que foi a utilização de uma heurística em dois níveis, onde em um primeiro nível seleciona-se um elemento que é utilizado no segundo nível para a geração da solução propriamente dita – alocação dos equipamentos de transmissão de dados (ADMs) nos nós da rede de telecomunicação para formação dos anéis unidirecionais e, conseqüentemente, atendimento das demandas de transmissão de dados.

Realizamos testes computacionais com o algoritmo proposto, variando os parâmetros que definem o tamanho da lista restrita de candidatos (RCL), nas duas sub-fases de construção.

Resolvemos inicialmente uma instância pequena do problema (5 nós) para comprovar a qualidade dos resultados obtidos. Ao compará-los com a solução ótima desta instância, obtidas através da solução do modelo de programação inteira apresentado para o problema (problema $P1$), observamos que a solução ótima foi obtida em todos as aplicações do GRASP implementadas.

Observamos que quando escolhemos α próximo ao guloso o tempo computacional tende a diminuir. Com α próximo ao guloso a lista restrita de candidatos (RCL) conterá elementos com melhor qualidade, isto é, que aceleram a obteção da solução no procedimento de construção, consequentemente diminuindo o tempo computacional.

Planejamos dar continuidade a este trabalho, desenvolvendo e testando novos procedimentos de busca local, funções gulosas alternativas para a fase de construção e finalmente tornando a implementação do procedimento mais robusta de forma a permitir a resolução de problemas de maior porte.

Desenvolvimentos futuros envolvem a implementação de novas funções “gulosas”, aperfeiçoamentos na fase de busca local, assim como a pesquisa de metaheurísticas híbridas, como a hibridização de GRASP com Path Relinking por exemplo.

Referências

[1] J. Bresina, Heuristic-biased stochastic sampling, In: Proceedings of the thirteenth national conference on artificial intelligence (AAAI-96), p. 271-278. American Association for Artificial Intelligence, (1996).

[2] J.L. Szwarcfiter , P.E. Lauer,A search strategy for the elementary cycles of a directed graph, *BIT* **16**(1975), 192-204.

[3] N. Maculan, M. M. Passini, J. A. M. Brito, A. Lisser, Column Generation Method for Network Design, Kluwer Academic Publishers, Netherlands, (2000).

[4] M.G.C.Resende e C.C. Ribeiro, Greedy randomized adaptive search procedure, AT&T Labs Research Technical Report, set 13 (2001).

[5] M.G.C.Resende, Combinatorial optimization in telecomunications, AT&T Labs Research Technical Report,july (2001).

[6] P.R.A.Lopes, Otimização Aplicada ao Planejamento de Anéis Unidirecionais em Telecomunicações, Tese de Mestrado, IM, NCE, UFRJ, Rio de Janeiro, Brasil, 2004.