

Algoritmos eficientes para o projeto de uma rede de telecomunicações com topologia em anel

Elder M. Macambira,

Depto de Estatística, UFPB,
58051-900, João Pessoa, PB
elder@de.ufpb.br

Cid C. de Souza,

Inst. de Computação, Unicamp,
13083-970, Campinas, SP
cid@ic.unicamp.br

Nelson Maculan Filho

PESC, COPPE, UFRJ,
21941-972, Rio de Janeiro, RJ
maculan@cos.ufrj.br

Luiz Satoru Ochi, Lucas de O. Bastos

Instituto de Computação, UFF,
24210-240, Niterói, RJ
satoru@ic.uff.br, lbastos@ic.uff.br

Resumo: *Os problemas de planejamento de redes de telecomunicações envolvem questões importantes como, por exemplo, a definição de uma topologia e o dimensionamento da rede. Geralmente, uma rede de telecomunicações possui mais de dois níveis (backbone, local e tributária). Neste trabalho, tratamos de um problema que aparece no planejamento de uma rede backbone com topologia em anel. Esse problema consiste em particionar um conjunto de localidades da rede backbone dentro de anéis SONET. Cada anel possuirá uma mesma capacidade e cada localidade deverá ser atribuída a um único anel. O tráfego de demandas entre as localidades pertencentes a anéis diferentes ficará a cargo de um anel especial denominado anel federal. O objetivo é projetar uma rede backbone com um número mínimo de anéis que atendam as restrições de capacidades e atribuição estabelecidas. Tal problema é denominado Problema de Atribuição de Localidades a Anéis SONET, denotado pela sigla SRAP (do inglês SONET Ring Assignment Problem). Neste trabalho, apresentaremos um estudo comparativo entre algumas heurísticas e metaheurísticas propostas para a resolução do SRAP. Estes algoritmos eficientes podem ser utilizados, por exemplo, na obtenção de bons limitantes superiores (iniciais) para um algoritmo empregado na resolução de forma exata do SRAP.*

Palavras-chave: problema de atribuição de localidades a anéis, particionamento de um grafo, otimização combinatória, heurísticas, metaheurísticas, telecomunicações.

1. Introdução

À medida que a indústria de telecomunicações introduz novas tecnologias no mercado, a natureza e o volume de serviços oferecidos aos usuários tendem a mudar. Vários exemplos destes serviços combinam, por exemplo, voz, dados e vídeo. Esta mudança implica em novos aspectos no planejamento de uma rede de telecomunicações, e conseqüentemente, a

necessidade de resolver problemas de otimização combinatória durante esta fase de planejamento.

Uma rede de telecomunicações é usualmente composta por três níveis: uma rede *backbone*, uma rede de acesso local e uma rede tributária. A rede *backbone* facilita o tráfego de informações entre os usuários, a rede de acesso local é encarregada de concentrar essas informações e disponibilizá-las as localidades da rede *backbone*, e por último a rede tributária é onde estão localizados os usuários.

Alguns problemas que surgem no projeto de uma rede de telecomunicações podem ser definidos como o auxílio de um grafo. Nestes casos, geralmente, deseja-se encontrar um subconjunto de vértices e (ou) arestas que satisfaça a determinados requisitos e, que otimize uma dada função objetivo. Esta última pode estar relacionada tanto com o custo de instalação da rede como com o atraso no envio de informações.

Neste trabalho, estudaremos um problema de particionamento de um grafo, denominado Problema de Atribuição de Localidades a Anéis, que surge no planejamento de uma rede *backbone*. Em particular, estaremos envolvidos com a definição da topologia desta rede, ou seja, nos preocuparemos em definir uma topologia em anel para ela.

A definição de uma topologia em anel de uma rede apresenta várias versões. Uma destas versões envolve uma estrutura hierárquica entre múltiplos anéis. Os anéis são considerados disjuntos. Além disso, temos a presença de um anel que realiza a conexão entre os demais anéis (c.f., [5]). Este anel especial recebe a denominação de *anel federal*. Em particular, o projeto da rede *backbone* que estudaremos neste trabalho seguirá a estrutura hierárquica desta classe.

A relevância deste trabalho, além de um estudo comparativo entre as diversas heurísticas e metaheurísticas propostas para a resolução do SRAP, deve-se a importância do cálculo de limitantes superiores (iniciais) para um algoritmo exato, como por exemplo um algoritmo *branch-and-price*, e conseqüentemente o emprego destes valores como uma

forma de melhorar o desempenho destes algoritmos (veja [7]).

Este trabalho está organizado em cinco seções. Os tópicos a serem cobertos em cada uma delas são descritos a seguir. Na seção 2, apresentamos uma descrição do Problema de Atribuição de Localidades a Anéis. Na seção 3, descrevemos as heurísticas gulosas e metaheurísticas propostas para a resolução do SRAP. Por último, na seção 4, apresentamos os resultados computacionais obtidos com o emprego das heurísticas e metaheurísticas. Nestes experimentos, foram utilizadas duas classes de instâncias existentes na literatura para o Problema de Atribuição de Localidades a Anéis. Algumas considerações finais são feitas na seção 5.

2. Descrição do problema

Nesta seção, será apresentada uma descrição do Problema de Atribuição de Localidades a Anéis. Considere o seguinte problema de otimização combinatória. Seja $G = (V, E)$ um grafo não direcionado com peso d_{uv} , associado a cada aresta $(u, v) \in E$, e B um inteiro positivo. Uma solução viável para este problema corresponde a uma partição do conjunto de vértices V em subconjuntos disjuntos V_1, V_2, \dots, V_k , tal que,

$$\sum_{\substack{u, v \in V_i \\ u < v}} d_{uv} + \sum_{\substack{u \in V_i \\ v \notin V_i}} d_{uv} \leq B, \text{ para todo } i \in \{1, \dots, k\} \text{ e,} \quad (1)$$

$$\sum_{i=1}^{k-1} \sum_{j=i+1}^k \sum_{u \in V_i, v \in V_j} d_{uv} \leq B. \quad (2)$$

O objetivo é encontrar uma solução viável que minimize o tamanho da partição, ou seja, o valor de k .

Este problema de otimização combinatória é conhecido na literatura como o Problema de Atribuição de Localidades a Anéis, denotado por SRAP (do inglês *SONET Ring Assignment Problem*).

O SRAP foi inicialmente estudado em [5]. Os autores provaram que o SRAP é NP-difícil. Além disso, os autores indicaram a relevância deste problema no projeto de redes de telecomunicações que fazem uso da tecnologia SONET (do inglês *Synchronous Optical Network*), e conseqüentemente aquelas que seguem a topologia em anel.

Logo, no projeto de redes SONET pode-se observar que os vértices do grafo estão associados as localidades da rede enquanto que as arestas estão associadas a existência (ou não) de tráfego entre pares de localidades. Os pesos associados as arestas indicam a demanda d_{uv} entre as localidades u e v da rede. Assim, dada uma solução viável para o problema de otimização combinatória definido acima, temos que os vértices em cada subconjunto da partição formam um *anel local* SONET (ou simplesmente anel). Além

disso, a demanda entre localidades, situadas em anéis diferentes, é controlada por um anel, denominado *anel federal*. A conexão entre um anel e o anel federal é feita através de um dispositivo conhecido como *Digital Cross-Connect* (DCS). Uma vez que a capacidade do anel federal é limitada por uma constante B , a soma dos pesos das arestas pertencentes aos cortes de uma solução viável não pode exceder a B . Finalmente, devido ao fato de que o equipamento DCS representa a parte mais cara para se projetar uma rede SONET, temos que o projeto de uma rede de baixo custo deve corresponder a uma solução que minimize o número de anéis locais, ou seja, o número de DCS.

3. Heurísticas e metaheurísticas

Nesta seção, apresentaremos alguns trabalhos propostos para a resolução, de forma heurística, do Problema de Atribuição de Localidades a Anéis. Inicialmente, descreveremos algumas heurísticas gulosas. Em seguida, apresentaremos uma breve descrição de algumas metaheurísticas existentes na literatura para o SRAP. Vale ressaltar que algumas destas técnicas foram propostas por nós.

3.1 Heurísticas gulosas

Uma heurística é um algoritmo projetado para encontrar uma boa solução, mas para a qual não se pode dar uma garantia da qualidade em relação à solução ótima.

Uma heurística é única, ou seja, só pode ser aplicada para aquele problema específico o qual ela foi projetada. Porém, qualquer heurística possui dois aspectos em comum: rapidez e alguma estratégia aplicada para o seu projeto.

Quando nos referimos à rapidez, se exige da heurística que ela consiga produzir uma solução “satisfatória” em uma quantidade de tempo razoável, mesmo quando a entrada do problema aumenta exponencialmente.

Já as estratégias se referem a possíveis maneiras de como podemos gerar uma solução. Duas são as estratégias comumente utilizadas: estratégia gulosa e busca local. As heurísticas que descreveremos nesta subseção fazem uso da estratégia gulosa.

A estratégia gulosa é uma das mais comuns, e talvez a mais importante das estratégias utilizadas no projeto de heurísticas. A solução é construída passo-a-passo, e em cada um desses passos um elemento é selecionado para ser adicionado na solução.

Em [5], os autores propuseram três heurísticas gulosas para resolver o Problema de Atribuição de Localidades a Anéis: *edge-based*, *cut-based* e *node-based*. Uma pequena descrição dessas heurísticas é apresentada logo abaixo.

3.1.1 Heurísticas *edge-based* e *cut-based*

As heurísticas *edge-based* (EB) e *cut-based* (CB) atribuem, inicialmente, cada localidade a um anel diferente; e em seguida, a cada iteração, dois anéis distintos são unidos, caso o anel resultante desta união seja viável.

Na heurística *edge-based*, as arestas são ordenadas em ordem crescente de suas demandas, e a escolha dos anéis para a união é feita, tomando-se a aresta de maior peso que ainda não foi selecionada. Na heurística *cut-based*, a escolha dos anéis é feita, tomando-se o tráfego máximo entre dois anéis quaisquer.

3.1.2 Heurística *node-based*

Já a heurística *node-based* (NB) recebe um valor k , correspondente ao número inicial de anéis. Estes anéis estão vazios *a priori*. Em seguida, escolhe-se, de forma aleatória, uma localidade u que será atribuída a um anel vazio. Este processo é repetido para cada um dos anéis ainda vazios.

Após a atribuição destas localidades, começa-se a trabalhar com as localidades que ainda não foram atribuídas a um anel. Neste caso, a cada iteração escolhe-se um anel r que possua a menor capacidade, e em seguida, escolhe-se uma localidade u que apresente o maior tráfego com o anel r .

3.2 Metaheurísticas

Uma metaheurística consiste de várias heurísticas de caráter genérico que se adaptam facilmente às estruturas de arquiteturas paralelas e são direcionadas à otimização global de um problema, podendo conter diferentes procedimentos heurísticos de busca local em sua estrutura.

Nas últimas décadas, surgiram vários procedimentos, enquadrados como metaheurísticas, empregados na resolução de problemas de otimização combinatória. Dentre elas, destacam-se busca tabu e GRASP. Uma referência recente sobre metaheurísticas pode ser encontrada em [9].

Em seguida, apresentamos uma pequena descrição de algumas metaheurísticas empregadas na resolução do Problema de Atribuição de Localidades a Anéis.

3.2.1 Busca tabu

Em [1], os autores propuseram uma metaheurística busca tabu, denotada por DMN, para a resolução do Problema de Atribuição de Localidades a Anéis. Para essa metaheurística foram implementadas duas estratégias de vizinhança.

A primeira vizinhança, denotada pelos autores por N1, inicia a partir de uma solução (viável ou inviável) e move uma localidade de um anel para outro anel, tal que o anel resultante seja viável. Já a segunda

vizinhança, denotada por N2, trabalha com uma solução obtida a partir da vizinhança N1, com o intuito de diminuir o número de anéis existentes na solução. Para isso, escolhe-se um anel que possua poucas localidades, e em seguida, se distribui as localidades desse anel para os outros anéis, independentemente da viabilidade da solução. Após esvaziar esse anel, a vizinhança N1 é executada novamente.

Além dessas duas estratégias de vizinhança, os autores também implementaram outras estratégias para a busca tabu, com o intuito de intensificar a busca dentro de regiões promissoras ou de diversificá-la gerando novas soluções. Dentre as estratégias implementadas pelos autores podemos citar: *scatter search*, *path relinking* e *eXploring tabu search*.

3.2.2 GRASP básico

Um GRASP (*Greedy Randomized Adaptive Search Procedures*) é um procedimento iterativo que combina várias propriedades favoráveis de outras heurísticas (vide [4]). Mais especificamente, cada iteração do GRASP consiste de dois estágios: uma *fase de construção* da solução e uma *fase de busca local*. Em cada iteração, uma solução é encontrada. A melhor solução obtida dentre todas as iterações é considerada a solução final.

Em [8], os autores propuseram uma metaheurística GRASP básico (GB) para o SRAP. Em seguida, descrevemos os dois estágios deste algoritmo.

Fase de construção

Na fase de construção, iniciamos com uma solução S constituída por n anéis, ou seja, cada localidade $u \in V$ é atribuída a um anel de mesmo índice. Além disso, consideramos que cada aresta $(u,v) \in E$ é uma aresta candidata para induzir a união viável entre os dois anéis r e t quaisquer com $u \in r$ e $v \in t$.

Um outro aspecto que deve ser levantado nesta fase é a obtenção *LRC*. A cada iteração, são escolhidas as β melhores arestas (u,v) candidatas. Este valor β , denominado *restrição de cardinalidade*, limita o tamanho de *LRC*. A lista *LRC* é obtida em duas etapas. Na primeira, construímos um conjunto de arestas que ainda não foram selecionadas, ou seja, as arestas cujos extremos estão em anéis diferentes. Na segunda etapa, utilizamos o algoritmo Estatística de Ordem, para obter as β primeiras arestas daquele conjunto, sem precisar ordená-las a cada iteração, o que demandaria um maior tempo computacional.

Por último, após a obtenção de *LRC*, selecionamos uma aresta $(u,v) \in LRC$ de forma aleatória, onde $u \in r$ e $v \in t$. Em seguida, realizamos o processo de adaptação, que consiste em unir os dois anéis r e t , compondo um único anel r . Caso a união dos anéis r e t não seja viável, apenas marcamos a aresta (u,v) como selecionada.

Fase de busca local

Na fase de busca local, as soluções vizinhas à solução obtida na fase de construção serão geradas, utilizando-se os movimentos de melhorias empregados, propostos em [6], para problemas de particionamento de grafos.

Dada uma solução viável S , tentamos iterativamente melhorar o valor da função objetivo. Esses melhoramentos podem ser obtidos através de dois procedimentos de perturbação. São eles:

- Pr1: considere dois anéis r e t da solução S . Uma localidade u pertencente ao anel r é movida para o anel t ;
- Pr2: nesse segundo movimento, realizamos uma troca de localidades entre dois anéis, ou seja, uma localidade u pertencente ao anel r é movida para o anel t e uma localidade v pertencente ao anel t é movida para o anel r .

Estas trocas possibilitam a formação de soluções vizinhas à solução corrente S . Em ambos os movimentos, Pr1 e Pr2, a solução corrente é atualizada, à medida que se encontra uma solução melhor na sua vizinhança.

3.2.3 GRASP básico com vizinhança relativa

Em [2], os autores propuseram uma nova heurística, denominada de Algoritmo Construtivo baseado em Vizinhança Relativa (ACVR), para a fase de construção do algoritmo GRASP básico definido em [8]. Esta nova versão do GRASP foi denominada de GB+ACVR.

Fase de construção

O Algoritmo Construtivo baseado em Vizinhança Relativa estabelece, inicialmente, quais localidades serão consideradas localidades relativamente vizinhas. A partir desta identificação, a construção dos anéis, na solução inicial, procurará manter localidades vizinhas em um mesmo anel.

O funcionamento do ACVR é descrito em seguida. Dado o grafo $G = (V, E)$, duas localidades u e $v \in V$, com $u \neq v$, serão consideradas vizinhas se a demanda d_{uv} entre elas for maior ou igual à maior demanda de todas as demandas envolvendo as localidades u e v .

Após a determinação das vizinhanças relativas, as localidades vizinhas são reunidas em um mesmo anel para a obtenção da solução inicial. Este procedimento gera no mínimo um anel e no máximo $n/2$ anéis. As demais localidades são atribuídas aos anéis iniciais com probabilidade $p = f * D(u, a)$ onde f é um fator aleatório e $D(u, a)$ é a demanda total entre a localidade u e o anel a . Em seguida, o algoritmo tenta unir os

anéis, com o intuito de minimizar o número deles na solução inicial, da mesma forma que a heurística *cut-based*.

Todas as heurísticas construtivas descritas aqui geram soluções com anéis viáveis. Dessa forma, se uma solução for inviável é devido a inviabilidade do anel federal. Logo, durante a fase construtiva do GB+ACVR, utiliza-se duas heurísticas construtivas (construtiva básica e por vizinhança relativa) num esquema de fila circular, isto é, se uma heurística executa um dado número de iterações sem melhorar a solução a próxima heurística na fila é ativada e passa a gerar soluções iniciais.

Busca local

Durante a busca local escolhe-se o anel com menor tráfego total e tenta-se mover suas localidades para outros anéis. Assim, pode-se melhorar a qualidade tanto de soluções viáveis como de soluções inviáveis.

Esses movimentos das localidades são feitos com base nos procedimentos de perturbação Pr1 e Pr2 definidos para o GRASP básico da seção 3.2.2.

3.2.4 GRASP com *path-relinking*

Apesar do GRASP descrito na seção 3.2.3, com uma fase construtiva baseada num critério de vizinhança relativa, apresentar bons resultados, a solução ótima em algumas instâncias não tinha sido alcançada.

Assim, com o propósito de aumentar a robustez do algoritmo GB+AVCR, e conseqüentemente, procurar melhorar a qualidade das soluções obtidas, foi desenvolvida e implementada uma heurística *path-relinking* (PR). Este novo algoritmo GRASP foi denominado de GB+ACVR+PR (veja [3]).

Heurística *path-relinking*

A idéia principal por trás de uma heurística *path-relinking* é: dadas duas soluções S e T , deseja-se encontrar uma solução melhor no caminho de transformação de S em T ou de T em S . Com isto, o *path-relinking* procura funcionar como uma fase de intensificação sobre as soluções obtidas na fase de busca local, ou seja, procura melhorar a qualidade das soluções.

Obviamente, cada problema possui suas particularidades e o processo de transformação de uma solução em outra pode diferir. Para o SRAP, definimos uma solução S como sendo um vetor de tamanho n . Cada posição u de S contém o número do anel ao qual a localidade u pertence, isto é, $S(u) = u$.

Sejam S e T duas soluções obtidas após a fase de busca local. O passo seguinte do procedimento *path-relinking* seria analisar o processo iterativo de transformar uma solução S na solução T . Este processo

de conversão de S em T consiste em fazer $S(i) = T(i)$, onde a localidade i é escolhida dentre todas as localidades j com $S(j) \neq T(j)$.

O critério de escolha da localidade i é guloso: escolhe a localidade que produz uma redução no valor da função objetivo. Ao final de n iterações do procedimento *path-relinking*, a solução S será igual à solução T , com isso foram geradas n soluções intermediárias. Se uma destas soluções intermediárias possuir um custo menor do que o melhor custo, então a solução global será atualizada.

Um outro aspecto importante é a escolha adequada das soluções S e T candidatas para o procedimento *path-relinking*. Levando em conta que o SRAP é um problema que procura minimizar o número de anéis, pode ocorrer que o número de anéis de S e T sejam diferentes e isto está diretamente ligado à qualidade das soluções intermediárias resultantes da aplicação do procedimento *path-relinking*. Dessa forma, é preciso avaliar se vale a pena executar o este tipo de procedimento sobre essas soluções. Para tomar essa decisão, duas soluções S e T são escolhidas somente se elas atendem as seguintes condições:

- i) $cr(S) = cr(T)$ e ambas as soluções são inviáveis;
- ii) $cr(S) < cr(T)$ e S é inviável;
- iii) $cr(S) > cr(T)$ e T é inviável.

onde $cr(S)$ corresponde ao número de anéis em S .

Caso a condição (ii) seja atendida, nós reduzimos o número de anéis da solução T . Isto pode ser feito através de uma mudança das localidades pertencentes ao menor anel de T para os outros anéis de T . Por outro lado, caso a condição (iii) seja satisfeita, a mesma estratégia de redução do número de anéis pode ser aplicada, só que com a solução S . Note que ao fazer isso, em uma determinada iteração as condições (ii) e (iii) não serão verdadeiras, e sim a condição (i).

Depois da escolha das soluções S e T , a heurística *path-relinking* é executada nos dois sentidos, ou seja, de S para T e de T para S . Com isso são gerados dois caminhos, e conseqüentemente a possibilidade de se visitar soluções intermediárias diferentes.

Um outro ponto importante no projeto do *path-relinking* é o gerenciamento do repositório (*pool*) de boas soluções. Para isso é definido o conceito de número meta de anéis (NMA). Este número serve como um guia para conduzir o *path-relinking* na escolha de soluções que apresentem o número de anéis igual ao valor NMA, e conseqüentemente, utilizadas para substituir as soluções mais antigas do repositório.

O valor utilizado para o NMA pertence ao intervalo $[k_{lb}, cr(s^*)]$, onde k_{lb} é o número mínimo de anéis para a rede e $cr(s^*)$ corresponde ao número de anéis da melhor solução s^* até o momento.

4. Resultados computacionais

Nesta seção, apresentamos os resultados computacionais obtidos com as metaheurísticas GRASP descrita anteriormente. Estas metaheurísticas foram implementadas utilizando a linguagem C. Todos os experimentos computacionais foram realizados em um computador com processador AMD K6 450MHz e com 256 MB de memória RAM.

4.1 Problemas testes

Para investigarmos o desempenho computacional das heurísticas e metaheurísticas, descritas nas seções anteriores, foram utilizadas duas classes de instâncias. Essas instâncias correspondem às mesmas definidas em [1,5].

Até o final desse texto denominaremos por classe C1 as instâncias geradas em [5], e por classe C2 as instâncias geradas em [1]. Para cada classe de instância, trabalhamos com dois tipos de instâncias, geométricas e aleatórias, denotados por “G” e “R”, respectivamente, e com dois valores, $B = 155$ Mbs (*low*) e $B = 622$ Mbs (*high*), para as capacidades dos anéis, denotados por “L” e “H”, respectivamente.

4.2 Experimentos

Para cada instância considerada durante os experimentos computacionais, nós fixamos a solução alvo igual ao valor da solução ótima. Em [7], o autor implementou um algoritmo exato que obteve os valores ótimos de todas as instâncias da classe C1 e C2. Além disso, para cada instância, foram feitas 1000 iterações e 10 execuções. Cada execução usando uma semente diferente.

Nas Tabelas 1 e 2, mostramos os resultados obtidos para as instâncias das classes C1 e C2, respectivamente, para cada algoritmo. Cada tabela lista o nome da instância (tipo), percentual de soluções ótimas encontradas e o tempo computacional médio gasto para obter a solução alvo (solução ótima).

Essas tabelas ilustram o efeito do procedimento *path-relinking* no algoritmo GB+AVCR. Apesar de computacionalmente caro, o procedimento melhorou a eficiência do algoritmo GRASP e conseguiu encontrar a solução ótima em 340 instâncias enquanto que o GB+AVCR encontrou o valor ótimo em 338 das 340 instâncias testadas.

Tipo	GB+AVCR		GB+AVCR+PR	
	OS (%)	tempo	OS (%)	tempo
GL	100.0	0.047	100.00	0.010
GH	100.0	0.054	100.00	0.014
RL	100.00	0.072	100.00	0.024
RH	100.00	0.046	100.00	0.042

Tabela 1: Resultados computacionais para C1.

Tipo	GB+AVCR		GB+AVCR+PR	
	OS (%)	tempo	OS (%)	tempo
GL	100.0	0.314	100.00	0.407
GH	97.1	0.384	100.00	1.630
RL	98.6	1.044	98.6	1.094
RH	100.00	0.747	100.00	0.204

Tabela 2: Resultados computacionais para C2.

Finalmente, na Tabela 3 comparamos os resultados obtidos com o algoritmo GB+AVCR+PR com os resultados obtidos pelos algoritmos existentes na literatura. Os resultados indicam que o GB+ACVR+PR obteve os melhores resultados em termos de qualidade da solução.

Descrição		GB + ACVR	EB, CB, NB	GB+ ACV+ PR
Classe	# Instância	OS (%)	OS (%)	OS (%)
C1	111	100.0	97.46	100.0
C2	230	99.57	-	98.26

Tabela 3: Comparação entre os algoritmos.

O tempo computacional gasto na resolução das instâncias não foi comparado por que os experimentos computacionais foram realizados em computadores diferentes. Entretanto, acreditamos que o algoritmo GB+ACVR+PR seja competitivo em relação aos demais algoritmos existentes na literatura.

5. Conclusão

Neste trabalho, apresentamos e descrevemos algumas heurísticas e metaheurísticas existentes na literatura para a resolução do Problema de Atribuição de Localidades a Anéis.

Alguns experimentos computacionais foram realizados com estas heurísticas e metaheurísticas com o intuito de verificar a eficiência de cada uma delas na obtenção de soluções de boa qualidade para o SRAP.

Através dos resultados obtidos, para ambas as classes de instâncias C1 e C2, nota-se que na maioria das instâncias testadas foi possível obter limitantes superiores de boa qualidade através das metaheurísticas GRASP, principalmente quando fizemos de uma heurística de *path-relinking*.

Referências

[1] R. Aringhieri e M. Dell'Amico. Solution of the SONET Ring Assignment Problem with Capacity Constraints, Proceedings of the 4th Metaheuristic International Conference, Porto, Portugal, 2001.

[2] L. de O. Bastos, L. S. Ochi e E. M. Macambira. A Relative Neighborhood GRASP for the SONET Ring Assignment Problem, Proceedings of the International Network Optimization, pp. 833–838, 2005.

[3] L. de O. Bastos, L. S. Ochi e E. M. Macambira. A GRASP with Path-Relinking for the SONET Ring Assignment Problem. Proceedings of the 5th International Conference on Hybrid Intelligent Systems, 2005 (aceito para publicação).

[4] T. A. Feo e M. G. C. Resende. Greedy Randomized Adaptive Search Procedures. Journal of Global Optimization, Vol. 6, 1995, pp. 109-133.

[5] O. Goldschmidt, A. Laugier e E. V. Olinick. SONET/SDH Ring Assignment with Capacity Constraints, Discrete Applied Mathematics, Vol. 129, 2003, pp. 99-128.

[6] B. W. Kernighan e S. Lin, S. An Efficient Heuristic Procedure for Partitioning Graphs. Bell Systems Technical Journal, Vol. 49, 1970, 291-307.

[7] E. M. Macambira. Modelos e Algoritmos de Programação Inteira no Projeto de Redes de Telecomunicações, UFRJ / COPPE / PESC, Tese de Doutorado, 2003.

[8] E. M. Macambira, N. Maculan Filho e C. C. de Souza. Projeto de uma Rede de Telecomunicações usando Metaheurísticas, Anais do XXXV Simpósio Brasileiro de Pesquisa Operacional, 2003.

[9] P. M. Pardalos e M. G. C. Resende. Combinatorial Optimization, em “Handbook of Applied Optimization” (P. M. Pardalos e M. G. C. Resende, eds), pp. 51-234. Oxford University Press, 2002.