

# UMA HEURÍSTICA EFICIENTE PARA O PROBLEMA DE COBERTURA DE CONJUNTO POR PARES

**Luciana Brugiolo Gonçalves, Simone de Lima Martins, Luiz Satoru Ochi**

Universidade Federal Fluminense – Instituto de Computação

Rua Passo da Pátria, 156 – Niterói, RJ

{lgoncalves, simone, satoru}@ic.uff.br

**Mauricio G. C. Resende**

AT&T Labs-Research

180 Park Avenue – Florham Park, NJ

mgrc@research.att.com

## RESUMO

Este trabalho trata do Problema de Cobertura de Conjunto por Pares (PCCP). Este consiste numa generalização do Problema de Cobertura de Conjunto (PCC), que por sua vez é reconhecido como NP-difícil. A diferença entre ambos os problemas é que no PCCP um elemento do universo é considerado coberto se pelo menos dois objetos forem selecionados para cobri-lo. Neste contexto, uma heurística composta de duas fases, construção e busca local, é proposta. O algoritmo desenvolvido foi testado em 560 instâncias disponíveis na literatura, tendo sido capaz de alcançar a solução ótima de várias instâncias.

**PALAVRAS CHAVE.** Cobertura de Conjunto por Pares. Heurísticas. Otimização.

**Área principal:** Otimização Combinatória – OC.

## ABSTRACT

This paper deals with the Set Cover by Pairs Problem (SCPP). This problem is a generalization of the Set Cover Problem (SCP), which, in turn, is known to be NP-hard. The difference between both problems is that, in the PCCP, one element of the universe is admitted to be covered if there are at least two objects chosen to cover it. In this context, a heuristic composed by two phases, construction and local search, is proposed. The algorithm developed was tested in 560 instances available in the literature and it was capable of achieving optimal solutions to several instances.

**KEYWORDS.** Set Cover with Pairs. Heuristics. Optimization.

**Main area:** combinatorial optimization - CO

## 1. Introdução

Neste trabalho é abordado o problema de cobertura de conjunto por pares (PCCP), que consiste numa generalização do problema de cobertura de conjunto (PCC), conhecidamente NP-difícil (Garey e Johnson, 1979). O problema de cobertura de conjunto pode ser descrito da seguinte forma: dado um conjunto de elementos  $U$  e uma coleção  $S$  de subconjuntos de  $U$ , onde a cada elemento  $j \in S$  é associado um custo não negativo  $w_j$ , o objetivo é encontrar um subconjunto  $A$  de elementos de  $S$  de custo mínimo de tal forma que todos os elementos de  $U$  estejam cobertos por  $A$ . Ou seja, dada a função de cobertura  $C: S \rightarrow 2^U$  que especifica para cada elemento de  $S$ , os elementos de  $U$  que são cobertos, o PCC consiste em encontrar um subconjunto  $A \subseteq S$  de custo mínimo tal que  $\cup_{i \in A} C(i) = U$ .

No problema de Cobertura de Conjunto por Pares (PCCP), introduzido por Hassin e Segev (2005), são considerados pares de elementos de  $S$  na função de cobertura. Assim, para todo par de elementos  $\{i, j\} \in S$ , tem-se que  $C(i, j)$  resulta nos elementos de  $U$  cobertos simultaneamente por  $i$  e por  $j$ . Desta forma, o PCCP tem como objetivo minimizar  $w(A) = \sum_{i \in A} w_i$  onde  $\cup_{\{i, j\} \in A} C(i, j) = U$ .

Uma aplicação que pode ser modelada como um PCCP é o problema do Conjunto Mínimo de Monitoramento (PCMM) que foi definido em Breslau *et al.* (2007). No PCMM o objetivo é monitorar um conjunto de nós roteadores utilizando nós de monitoramento. Considerando  $B$  o conjunto de nós roteadores que deseja-se monitorar e  $M$  o conjunto de candidatos a monitores, para que um determinado roteador  $b \in B$  esteja coberto é necessário selecionar, entre os candidatos a nós de monitoramento do roteador  $b$ , um par de elementos,  $m'$  e  $m''$ , de tal forma que não exista interseção entre os caminhos que conectam estes nós ao roteador  $b$ .

Na Figura 1 é possível observar um exemplo desta aplicação, em que é apresentada uma rede contendo um total de cinco nós, sendo um roteador  $B = \{b\}$ , três nós candidatos a monitores  $M = \{m_1, m_2, m_3\}$  e um nó intermediário  $r$ , que não participa nem do conjunto  $B$  nem de  $M$ . Sendo  $P_b$  o conjunto dos pares de  $M$  que podem ser utilizados para cobrir  $b$ , e dado que o protocolo de roteamento produziu as rotas tracejadas destacadas na figura, então  $P_b = \{\{m_1, m_2\}, \{m_1, m_3\}\}$ . O par de candidatos  $\{m_2, m_3\}$  não pode ser utilizado para cobrir  $b$ , pois os caminhos que conectam estes nós a  $b$  compartilham, além do próprio nó  $b$ , o nó  $r$ .

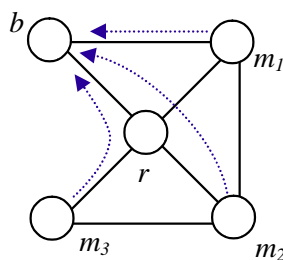


Figura 1. Exemplo de Rede.

Neste contexto, considere um grafo  $G = (V, E)$  que representa a topologia física da rede, onde  $V$  representa o conjunto de todos os nós da rede e  $E$  o conjunto de *links*. Além disso, considere dois subconjuntos de  $V$ , não necessariamente disjuntos,  $B$  e  $M$ , onde  $B$  representa o conjunto de roteadores que devem ser monitorados e  $M$  o conjunto de candidatos a nós de monitoramento, sendo que a cada nó  $m \in M$  é associado um mesmo custo  $c$ . Dados os caminhos entre cada par de nós  $(b, m)$ , onde  $b \in B$  e  $m \in M$ , o objetivo é selecionar um subconjunto  $A \subseteq M$  de tal forma que existam pelo menos dois caminhos disjuntos entre elementos de  $A$  para cada um dos elementos de  $B$ . Neste caso, a função de cobertura  $C(i, j)$ , onde  $i, j \in M$ , resulta nos nós  $b \in B$  que podem ser monitorados por  $i$  e  $j$  simultaneamente. Na prática, considerando o mesmo custo associado a cada elemento de  $M$ , o objetivo é realizar o monitoramento de todos os roteadores em

$B$  utilizando o menor número possível de nós escolhidos entre os elementos de  $M$ .

Como detalhado em Breslau *et al.* (2007), o PCMM pode ser visto como um caso especial do PCCP. Pode-se perceber tal relação entre os dois problemas associando ao conjunto  $U$  o conjunto  $B$  dos nós que devem ser cobertos e considerando  $S$  os possíveis nós candidatos a monitores  $M$ . Desta forma, um par de nós  $\{u, v\}$ , onde  $u, v \in M$ , pertence a  $P_b$  somente se não há interseção entre os caminhos que conectam  $u$  e  $v$  ao nó  $b$ .

As primeiras propostas para o PCCP foram apresentadas por Hassin e Segev (2005), que apresentaram um algoritmo guloso obtido a partir do algoritmo de Chvátal (1979) para o PCC, bem como outros algoritmos para casos particulares do problema de cobertura por pares.

Em Breslau *et al.* (2007) foram apresentadas algumas abordagens para o PCCP: uma simplificação do algoritmo de Hassin e Segev (2005), um Algoritmo Genético que explora a idéia de chave aleatória (Beans, 1994) e abordagens exatas para o problema (programação inteira e limites inferiores). Além destas abordagens, a heurística *Double Hitting Set* foi desenvolvida especificamente para o PCMM e explora características específicas da rede.

Neste trabalho são apresentadas duas heurísticas para o PCCP. Os resultados são comparados com aqueles obtidos por Breslau *et al.* (2007). Na Seção 2 será apresentada a heurística construtiva proposta e, na Seção 3, o algoritmo de busca local. Na Seção 4 são apresentadas as comparações com as heurísticas de Breslau *et al.* (2007) e, por fim, na Seção 5 são colocadas as conclusões do trabalho.

## 2. Heurística Construtiva

Considera-se que, para cada nó a ser coberto  $b \in B$ , todos os possíveis pares de nós de  $M$  que podem ser utilizados para sua cobertura são conhecidos. O algoritmo proposto é baseado na escolha dos melhores pares para cada um dos nós de  $B$ , por isso a heurística é denominada Heurística do Melhor Par (HMP). Uma descrição do algoritmo pode ser vista na Figura 2.

Algoritmo Construtivo HMP ( $B, M$ )	
1.	$S = \{ \}$
2.	ordenar ( $B$ )
3.	enquanto ( não viável )
4.	$b =$ próximo não coberto( $B$ )
5.	$\{m_1, m_2\} =$ melhor par( $b$ )
6.	$S = S \cup \{m_1, m_2\}$
7.	fim-enquanto
8.	retorna solução

Figura 2. Algoritmo Construtivo.

Na primeira linha do algoritmo, o conjunto de elementos selecionados que representa uma solução é inicializado vazio. Na linha 2 os nós de  $B$  são ordenados de forma não decrescente, de acordo com o número de possíveis pares de  $M$  que podem cobri-lo. Depois disto, enquanto existir algum nó não coberto em  $B$  (linha 3), será escolhido o melhor par para um dado nó  $b$  (linha 4) selecionado segundo a ordem estabelecida no início do algoritmo. Na próxima instrução, linha 5, é determinado o par de  $b$  com maior valor associado a função  $f(m_1, m_2)$  calculado de acordo com a Equação 1, onde  $\#cobertura(m_1, m_2)$  indica o número de novos elementos de  $B$  que são cobertos se este par for escolhido e  $custo(m_1, m_2)$  representa o custo que será acrescido na solução com a escolha de  $m_1$  e  $m_2$ . Caso algum elemento do par já tenha sido escolhido em iterações anteriores, este não implicará em nenhum acréscimo se estiver no próximo par selecionado. Quando considerado custo único para todos os elementos de  $M$ , o custo de uma solução é dado pelo número de elementos selecionados para participar da solução.

$$f(m_1, m_2) = \#cobertura(m_1, m_2) / custo(m_1, m_2) \quad (1)$$

Se houver empate no valor da função de avaliação, um dos pares é escolhido aleatoriamente. Após a seleção, na linha 6, os monitores  $m_1$  e  $m_2$  são incluídos na solução  $S$  e seu custo, que é dado pela soma dos custos dos elementos selecionados, é atualizado.

Considerando  $n = |B|$  e  $p = \max\{|P_b|, b \in B\}$ , como a estratégia adotada verifica, para cada elemento do conjunto  $B$ , o melhor entre os possíveis pares, a abordagem implementada possui complexidade máxima na ordem de  $n \times p$ , valor este inferior à complexidade dos algoritmos que verificam todos os possíveis pares de  $M$ .

### 3. Busca Local

Apesar da boa qualidade da solução obtida pela heurística construtiva para a maioria das instâncias, foi desenvolvido um algoritmo de busca local com o objetivo de escapar de possíveis ótimos locais, comuns em abordagens gulosas.

Como o custo de um par de elementos de  $M$  está associado ao estado atual da solução, na heurística construtiva apresentada na seção anterior a ordenação estabelecida entre os elementos de  $B$  é um dos pontos que afeta diretamente a escolha do melhor par para cobrir um determinado nó. No procedimento de busca local deseja-se alterar a ordem de escolha dos nós a serem cobertos de forma a alterar os nós de monitoramento selecionados, este procedimento está detalhado na Figura 3.

Algoritmo Busca Local (solucao S)	
1.	melhor = $S$
2.	enquanto ( iterações sem melhoria $< k$ )
3.	$M' = \text{Escolhe\_Selecionados}( S )$
4.	$S = S \setminus M'$
5.	$S2 = \text{Reconstruir}( S )$
6.	se ( Custo ( $S2$ ) $<$ Custo ( <i>melhor</i> ) ) então
7.	$melhor = S2$
8.	fim-se
9.	$S = S2$
10.	fim-enquanto

Figura 3. Algoritmo de Busca Local.

O algoritmo de busca local recebe como parâmetro uma solução  $S$  obtida pelo algoritmo construtivo descrito na Seção 2. A primeira etapa do algoritmo consiste em retirar alguns dos nós de monitoramento selecionados na solução corrente. Para isto, inicialmente são escolhidos aleatoriamente  $\lambda\%$  dos nós de monitoramento selecionados na solução corrente ( $S$ ) formando o conjunto  $M'$  (linha 3). Após a escolha, os elementos de  $M'$  são retirados na solução (linha 4), o que a torna inviável, ou seja, deixando descobertos alguns dos elementos de  $B$ . Para viabilizar a solução, gerando a solução  $S2$ , realiza-se a reconstrução utilizando a mesma idéia implementada no algoritmo construtivo apresentado na seção anterior, considerando apenas os elementos descobertos em  $B$  na solução parcial (linha 5).

Caso a solução  $S2$  tenha um custo inferior ao custo da melhor solução encontrada pelo procedimento, esta será atualizada de forma a armazenar sempre a melhor solução encontrada até o momento (linhas 6, 7 e 8). Este processo se repete enquanto melhores soluções estiverem sendo encontradas, sendo interrompido quando o algoritmo realizar  $k$  iterações sem atualizar a melhor solução.

### 4. Resultados Computacionais

Para avaliar os algoritmos propostos foram realizados experimentos em uma máquina Intel(R) Core(TM)2 Duo CPU T7250 com 2GHz e 2GB de RAM, onde os algoritmos

foram implementados usando a linguagem C++ e compilada utilizando g++ versão 4.1.3 (Linux Ubuntu 4.1.2-16).

No algoritmo de busca local, o valor de  $k$  foi fixado em 5 e o parâmetro  $\lambda$  em 30, ou seja, em cada iteração 30% dos elementos selecionados são retirados da solução, estes parâmetros foram ajustados empiricamente após testes preliminares.

Tabela 1. Resumo dos Resultados.

Instâncias			Heurísticas			Ótimo (PI)	Limite Alcançados	Ótimos Conhecidos
Grupo	B	M	AG	HH	HMP <sup>+</sup>			
M1_B1	N	N	76	57	79	31	30	61
M1_B2	N	N /2	76	69	80	30	34	64
M1_B4	N	N /4	76	65	80	30	43	73
M1_B8	N	N /8	78	74	80	31	42	73
M2_B2	N /2	N /2	80	71	79	51	17	68
M4_B4	N /4	N /4	79	77	80	71	4	75
M8_B8	N /8	N /8	79	76	80	80	0	80
Total de 560 instâncias			<b>544</b>	<b>489</b>	<b>558</b>	<b>324</b>	<b>170</b>	<b>494</b>

Para avaliação das heurísticas propostas foram utilizadas 560 instâncias disponíveis em Resende (2008). Em todas as instâncias é considerado que cada elemento de  $M$  possui um mesmo custo. Para melhor apresentar os resultados, as 560 instâncias foram distribuídas em sete grupos, como mostra a Tabela 1, de acordo com a forma como foi determinado o número de elementos dos grupos  $B$  e  $M$ . Cada um dos grupos possui 80 instâncias onde o total de nós  $|N|$  varia entre 26 e 558 elementos. O número de elementos de  $B$  e  $M$ , para cada um dos grupos, variou entre  $|N|$  e  $|N|/8$ .

Nas três primeiras colunas da Tabela 1 são apresentadas informações sobre as instâncias. Na coluna *Grupo* é apresentado o rótulo de cada conjunto de instâncias, na coluna  $|B|$  o número de nós a serem monitorados e na coluna  $|M|$  o número de nós onde é possível instalar equipamento de monitoramento.

Os resultados obtidos pelas heurísticas são resumidos nas colunas AG, HH e HMP<sup>+</sup>. Vale destacar que as colunas AG e HH trazem, respectivamente, os resultados preliminares do Algoritmo Genético (AG) e da heurística *Double Hitting Set* (HH) - Breslau *et al.* (2007). A coluna HMP<sup>+</sup> mostra os resultados obtidos a partir de uma execução da heurística construtiva HMP seguida pela aplicação da busca local proposta neste trabalho. Em cada uma destas colunas (AG, HH e HMP<sup>+</sup>) é possível verificar, para cada uma das heurísticas, o número de vezes que estas alcançaram o melhor resultado conhecido, lembrando que cada grupo possui 80 instâncias.

A heurística gulosa apresentada em Breslau *et al.* (2007) não foi analisada por apresentar resultados inferiores aos demais algoritmos propostos pelos autores. A heurística gulosa de Hassin e Segev (2005) foi implementada, mas o seu resultado não será considerado neste trabalho devido ao elevado custo computacional do algoritmo.

Na Tabela 1, pode-se observar que para uma instância do grupo *M1-B1* e para outra do grupo *M2-B2* a heurística proposta HMP<sup>+</sup> não conseguiu encontrar o melhor resultado conhecido, ou seja, em 558 das 560 instâncias a heurística HMP<sup>+</sup> alcançou ou superou as heurísticas AG e HH de Breslau *et al.* (2007), o que representa 99,64% das instâncias. As heurísticas AG e HH alcançaram a melhor solução, respectivamente, em 97,14% e 87,32% das instâncias.

Nas três últimas colunas são oferecidas informações sobre o número de instâncias onde o custo da solução ótima é conhecido. Em 324 instâncias o valor ótimo foi obtido através de programação inteira e, dentre as demais, em 170 instâncias os limites inferiores foram alcançados por algum dos algoritmos heurísticos, totalizando 494 instâncias com ótimos conhecidos. Tanto os limites inferiores quanto os resultados ótimos foram obtidos por Breslau *et al.* (2007). Para todas as instâncias com ótimo conhecido, a heurística proposta foi capaz de encontrar este valor.

Para a maioria das instâncias as heurísticas comparadas encontraram soluções de mesmo custo, em 81 das 560 instâncias houve divergência entre os custos obtidos. Para ressaltar esta diferença, estas 81 instâncias são apresentadas na Tabela 2, agrupadas da mesma forma como especificado na tabela anterior. Na primeira coluna são apresentadas as instâncias, nas três colunas seguintes pode-se observar o desvio percentual de cada abordagem em relação ao melhor resultado conhecido, que é apresentado na coluna intitulada H\*. Para aquelas heurísticas que obtiveram solução com custo H\*, as células correspondentes foram preenchidas com sinal “\*”. Nas colunas com rótulos IP e LB são apresentados, respectivamente, o valor do ótimo (quando este é desconhecido utiliza-se o sinal “?”) e do limite inferior. Na coluna H\*, o valor está em negrito quando se trata do valor ótimo da instância.

Tabela 2. Instâncias onde algoritmos divergem.

Instância	AG(%)	HH(%)	HMP(%)	H*	IP	LB	Instância	AG(%)	HH(%)	HMP(%)	H*	IP	LB	
M1-B1	n100-i9-m100-b100	3,3	*	*	<b>30</b>	30	30	n26-i9-n26-b7	*	25,0	*	<b>4</b>	4	4
	n190-i3-m190-b190	*	2,3	*	<b>44</b>	?	44	n300-i1-n300-b75	2,9	*	*	<b>34</b>	?	34
	n190-i7-m190-b190	*	2,0	*	<b>50</b>	?	50	n300-i2-n300-b75	*	2,7	*	<b>37</b>	?	37
	n220-i1-n220-b220	*	1,8	*	<b>57</b>	?	57	n300-i4-n300-b75	*	2,8	*	<b>36</b>	?	36
	n220-i7-n220-b220	*	2,1	*	<b>48</b>	?	48	n300-i5-n300-b75	*	3,3	*	<b>30</b>	?	30
	n250-i0-n250-b250	*	1,8	*	<b>57</b>	?	57	n300-i6-n300-b75	*	8,6	*	<b>35</b>	?	35
	n250-i9-n250-b250	*	3,6	*	55	?	53	n300-i7-n300-b75	*	8,6	*	<b>35</b>	?	35
	n26-i1-n26-b26	20,0	*	*	<b>5</b>	5	5	n558-i0-n558-b140	*	3,6	*	56	?	55
	n300-i1-n300-b300	*	1,6	*	<b>63</b>	?	63	n558-i2-n558-b140	*	1,8	*	56	?	55
	n300-i2-n300-b300	*	3,0	*	<b>66</b>	?	66	n558-i4-n558-b140	*	1,9	*	<b>54</b>	?	54
	n300-i4-n300-b300	*	2,8	*	71	?	70	n558-i6-n558-b140	1,8	3,6	*	<b>55</b>	?	55
	n300-i6-n300-b300	*	2,7	*	<b>73</b>	?	73	n558-i7-n558-b140	*	1,7	*	59	?	58
	n300-i8-n300-b300	*	1,6	*	<b>62</b>	?	62	n558-i8-n558-b140	1,7	1,7	*	59	?	57
	n300-i9-n300-b300	*	1,5	*	67	?	66	n558-i9-n558-b140	*	1,8	*	<b>55</b>	?	55
	n50-i7-n50-b50	*	7,7	*	<b>13</b>	13	13	n100-i8-m100-b13	*	14,3	*	<b>7</b>	7	7
	n558-i0-n558-b558	0,9	1,7	*	115	?	112	n26-i9-n26-b4	*	33,3	*	<b>3</b>	3	2
	n558-i1-n558-b558	*	0,9	*	115	?	114	n300-i1-n300-b38	4,3	*	*	<b>23</b>	?	23
	n558-i2-n558-b558	*	0,9	*	110	?	107	n300-i6-n300-b38	*	4,0	*	<b>25</b>	?	25
	n558-i3-n558-b558	1,0	3,9	*	103	?	102	n300-i7-n300-b38	*	16,7	*	24	?	23
	n558-i4-n558-b558	*	0,9	*	112	?	109	n558-i0-n558-b70	*	2,6	*	39	?	38
n558-i5-n558-b558	*	1,0	*	<b>102</b>	?	102	n558-i7-n558-b70	*	2,6	*	<b>38</b>	?	38	
n558-i6-n558-b558	*	1,8	*	109	?	106	n558-i9-n558-b70	2,4	*	*	41	?	40	
n558-i7-n558-b558	*	0,9	*	106	?	101	n100-i3-n50-b50	*	5,6	*	<b>18</b>	18	18	
n558-i8-n558-b558	*	1,9	*	106	?	103	n190-i7-n95-b95	*	2,8	*	<b>36</b>	36	36	
n558-i9-n558-b558	*	3,7	0,9	109	?	105	n250-i2-m250-b125	*	2,3	*	<b>44</b>	?	44	
M1-B2	n190-i7-m190-b95	*	2,9	*	<b>34</b>	?	34	n26-i8-m13-b13	*	25,0	*	<b>4</b>	4	4
	n220-i7-m220-b110	*	2,9	*	<b>35</b>	?	35	n300-i6-m150-b150	*	12,2	*	<b>49</b>	?	49
	n220-i8-m220-b110	*	2,6	*	<b>38</b>	?	38	n558-i3-n279-b279	*	1,3	1,3	79	?	77
	n250-i2-m250-b125	*	2,5	*	<b>40</b>	?	40	n558-i7-n279-b279	*	1,3	*	77	?	76
	n26-i1-n26-b13	25,0	*	*	<b>4</b>	4	4	n558-i8-n279-b279	*	3,7	*	82	?	78
	n300-i1-n300-b150	*	2,2	*	<b>46</b>	?	46	n558-i9-n279-b279	*	1,3	*	79	?	78
	n300-i4-n300-b150	*	2,2	*	<b>46</b>	?	46	n26-i4-n7-b7	25,0	*	*	<b>4</b>	4	4
	n300-i6-n300-b150	*	4,2	*	<b>48</b>	?	48	n300-i6-n75-b75	*	8,3	*	<b>36</b>	36	36
	n50-i7-n50-b25	12,5	12,5	*	<b>8</b>	8	8	n558-i2-m140-b140	*	1,7	*	60	?	58
	n558-i2-n558-b279	1,2	1,2	*	81	?	78	n558-i6-m140-b140	*	1,6	*	<b>61</b>	?	61
	n558-i5-n558-b279	1,3	*	*	75	?	73	n300-i6-n88-b38	*	3,7	*	<b>27</b>	27	26
	n558-i6-n558-b279	*	1,3	*	76	?	73	n300-i7-n88-b38	*	26,1	*	<b>23</b>	23	22
n558-i9-n558-b279	*	3,7	*	81	?	80	n50-i0-n7-b7	*	25,0	*	<b>4</b>	4	4	
M1-B4	n100-i5-m100-b25	*	8,3	*	<b>12</b>	12	12	n50-i7-n7-b7	25,0	*	*	<b>4</b>	4	4
	n100-i6-m100-b25	*	8,3	*	<b>12</b>	12	12	n558-i6-m70-b70	*	2,3	*	<b>43</b>	43	42
	n190-i8-m190-b48	3,7	*	*	<b>27</b>	?	27							

81 Instâncias

Pode-se observar na Tabela 2 que a heurística HH apresentou o pior desempenho nestas comparações, deixando de alcançar  $H^*$  em 71 das 81 instâncias, enquanto o AG não alcançou em 16 e o  $HMP^+$  em apenas 2. Para uma das instâncias do grupo M1-B4, n558-i6-m558-b140, a heurística proposta alcançou o limite inferior, que não tinha sido encontrado pelas demais heurísticas até então, ou seja, foi provado o custo da solução ótima desta instância.

Para avaliar o custo em relação ao tempo de execução, na Figura 4 é apresentado o gráfico do tempo de execução, em segundos, utilizado pela heurística construtiva. A informação referente a cada um dos grupos foi colocada separadamente, ou seja, para cada um dos sete grupos é apresentada uma curva onde se mostra o tempo médio de execução de 10 instâncias que possuem o mesmo número de nós na rede.

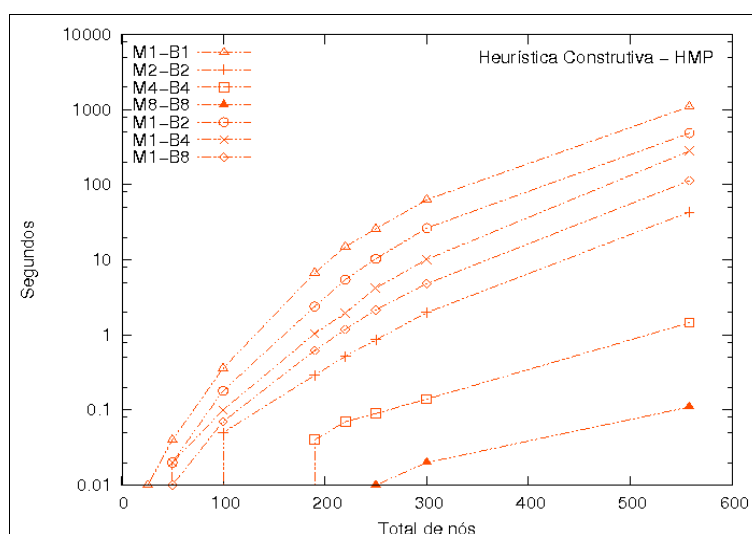


Figura 4. Tempo de CPU do algoritmo construtivo.

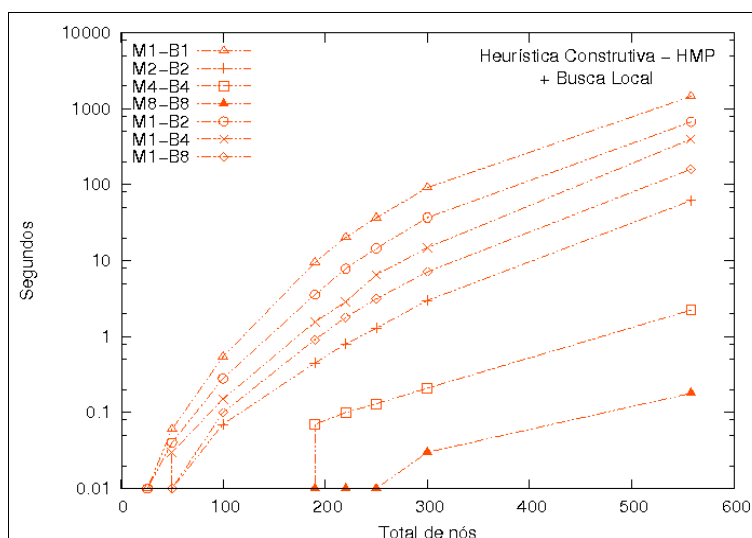


Figura 5. Tempo de CPU do construtivo + busca local.

Como era esperado, pode ser observado na Figura 4 o aumento do tempo de execução de acordo com o número de nós dos conjuntos  $M$  e  $B$ . O tempo de execução foi, no máximo, próximo a 1000 segundos para as instâncias com maior número de possíveis combinações.

Devido à boa qualidade das soluções geradas pelo algoritmo de construção, a busca local

não é dispendiosa, pois não consegue melhorar a solução corrente por muitas iterações. No gráfico da Figura 5 é apresentado, de maneira similar ao que foi feito na figura anterior, o tempo total, ou seja, o tempo do algoritmo construtivo somado ao tempo da busca local.

A busca local não causou aumento significativo no tempo de execução e sua utilização permitiu encontrar soluções melhores para algumas instâncias. Como em Breslau *et al.* (2007) não foi especificada a configuração das máquinas utilizadas nos testes computacionais, não foi possível realizar uma comparação justa entre os tempos obtidos com os tempos dos algoritmos da literatura.

## 5. Conclusões

Neste trabalho foi apresentada uma heurística HMP<sup>+</sup> composta por um algoritmo construtivo e uma busca local para o PCCP. Para analisar o desempenho do algoritmo proposto foi utilizado um conjunto de 560 instâncias disponíveis em Resende (2008). Os resultados obtidos mostram um grande potencial da heurística proposta, visto que em 558 das instâncias os resultados obtidos foram iguais ou melhores aos resultados da literatura e, nas 2 instâncias restantes, HMP<sup>+</sup> ficou a apenas 1,3% e 0,9% do custo da melhor solução conhecida.

## Agradecimentos

Ao CNPQ, CAPES e FAPERJ que financiaram parcialmente este trabalho.

## Referências

**Beans, J.** (2004), Genetics and random keys for sequencing and optimization. *ORSA Journal on Computing*, 6, 154-160.

**Breslau, L. Diakonikolas, I. Duffield, N. Gu, Y. Hajiaghayi, M. Johnson, D.S.Karloff, H. Resende, M.G.C. Sen, S. e Towsley, D.** "Optimal node placement for path disjoining network monitoring", *Working paper*, AT&T Labs Research, Florham Park, NJ 07932 USA, 2007.

**Chvátal, V.** (1979), A greedy heuristic for the set covering problem. *Mathematics of Operations Research*, 4, 233–235.

**Garey, M. e Johnson, D.** *Computers and Intractability: A guide to the theory of NP-completeness*, W.H. Freeman, San Francisco, 1979.

**Hassin, R. e Segev, D.** (2005), The Set Cover with Pairs Problem. *Proceedings of the 25<sup>th</sup> Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, LNCS 3821, 164-176.

**Resende, M.** (2008) url: <http://mauricio.resende.info/covering-by-pairs/>, acessado em 03/2008.