

---

# Método do Volume Revisado para o problema Set Covering

João C. Abreu<sup>1</sup>

<sup>1</sup>*Universidade Federal de Minas Gerais, DCC,  
Avenida Antônio Carlos 6627, Belo Horizonte, Brazil,  
joao.junior@dcc.ufmg.br*

Abstract      Esse relatório apresenta o algoritmo do volume revisado para o problema Set Covering.

Keywords:    Set Covering, Volume Revisado

## 1. Introdução

Dados  $M = \{1, \dots, m\}$  e  $N = \{1, \dots, n\}$  dois conjuntos. Seja  $M_1, M_2, \dots, M_n$  uma coleção de subconjuntos de  $M$  com um custo  $c_j$  associado a cada um desses subconjuntos. Uma cobertura de  $M$  é um subconjunto  $F \subset N$  tal que  $\cup_{j \in F} M_j = M$ . O problema que consiste em encontrar esse subconjunto  $F \subset N$  de custo mínimo é denominado o problema Set Covering (*SCP*). Segundo Balas[1] esse problema é *np-difícil*.

Para formular o *SCP* como um problema de otimização inteira, é introduzida uma matriz de incidência  $A$  de tamanho  $m \times n$  para a coleção de subconjuntos  $M_j, \forall j \in N$ , com as entradas dadas por:

$$a_{ij} = \begin{cases} 1; & \text{se } i \in M_j, \\ 0; & \text{caso contrário} \end{cases}$$

A formulação para o *SCP* proposta em [?] é composta pela função objetivo (1) e pelas restrições (2)-(3). Nessa formulação, a função objetivo (1) minimiza o custo da cobertura procurada. Os valores constantes  $c_j, \forall j \in N$  são os custos de cada subconjunto  $M_j$ . A variável de decisão  $x_j$  é igual a 1 quando  $j \in F$  e 0, caso contrário, onde  $F \subset N$  é a cobertura procurada.

$$\min \sum_{j \in N} c_j x_j \quad (1)$$

Sujeito à:

$$Ax \geq 1 \quad (2)$$

$$x \in \{0, 1\}^n \quad (3)$$

O problema lagrangeano será composto pela associação dos multiplicadores as restrições em (2) e ficará como:

$$\theta(\pi) = \min \sum_{j \in N} c_j x_j + \pi(Ax - 1) \quad (4)$$

Sujeito à:

$$x \in \{0, 1\}^n \quad (5)$$

## 2. Algoritmo

A figura 1 apresenta o algoritmo do volume revisado adaptado para o problema lagrangeano  $\theta(\pi)$  retirado de [2].

## 3. Experimentos Computacionais

Os experimentos computacionais foram executados em uma máquina Intel Dual-Core de 2.81 GHz de clock e 2GB de memória RAM, rodando o sistema operacional Linux. O modelo matemático composto pela função objetivo (1) e as restrições (2)-(3) foi implementado no Ilog CPLEX 12.5.1 e o algoritmo da seção 2 foi implementado em Python 2.7, sendo que o otimizador utilizado para resolver o problema lagrangeano  $\theta(\pi)$  presente nesse algoritmo foi o Ilog CPLEX 12.5.1. Foram utilizados quatro conjuntos de instâncias de testes nos experimentos computacionais e essas instâncias foram retiradas de [3]. Nessas instâncias de testes cada linha da matriz de incidência  $A$  é coberta por pelo menos duas colunas e cada coluna cobre pelo menos uma linha. O custo  $c_j$  de cada coluna  $j$  está entre  $[1, 100]$ . A tabela 1 resume esses conjuntos de instâncias. A coluna 1 dessa tabela representa o identificador do conjunto da instância de teste, as colunas 2 e 3 mostram, respectivamente, o número  $m$  de linhas e  $n$  de colunas da matriz de

```

1 inicio
2   Passo0: Seja  $m_1 \in (0, 1)$  uma tolerância. Dado um multiplicador  $\pi_0$  inicial,
   resolva o problema lagrangeano  $\theta(\pi_0)$  e obtenha os valores de  $x$ . Faça
    $v_0 = Ax_0 - 1$ . Inicialize  $z_1 = x_0$ ,  $\pi_1^a = \pi_0$ ,  $w_1 = v_0$ ,  $p_1 = \pi_0$  e  $\epsilon_1 = 0$ . Faça
    $k = t = 1$  e  $T_s = \emptyset$ .
3   Passo1: Tendo um centro  $\pi_k^a$  e um passo  $s_t > 0$ , calcule:
4      $\pi_t = \pi^k + s_t w_t$ 
5     e
6      $\delta_t = s_t \|w_t\|^2 + | \langle w_t, \pi_k^a - p_t \rangle | + \epsilon_t$ 
7     if  $\delta_t \leq \delta_{min}$  then
8       Pare
9     end
10  Passo2: resolva o problema lagrangeano  $\theta(\pi_t)$  e obtenha os valores de  $x_t$  e
   faça  $v_t = Ax_t - 1$ .
11  Passo3: if  $\theta_t \geq \theta_k^a + m_1 \delta_t$  then
12     $\pi_{k+1}^a = \pi_t$ 
13     $t_k = t$ 
14     $T_s = T_s \cup \{t_k\}$ 
15     $k = k + 1$ 
16  end
17  Passo4: Calcule um novo passo  $s_{t+1}$ .
18  Calcule:
19     $z_{t+1} = \alpha_t x_t + (1 - \alpha_t) z_t$ 
20     $w_{t+1} = \alpha_t v_t + (1 - \alpha_t) w_t$ 
21     $p_{t+1} = \alpha_t \pi_t + (1 - \alpha_t) p_t$ 
22     $\epsilon_{t+1} = \alpha_t \sigma_t + (1 - \alpha_t) \epsilon_t$ 
23  com
24     $\alpha_t = \min_{\alpha \in [0,1]} \frac{s_{t+1}}{2} \|\alpha v_t + (1 - \alpha) w_t\|^2 + \alpha \epsilon_t + (1 - \alpha) \epsilon_t^a$ 
25     $e_t = \langle v_t, \pi_k^a - \pi_t \rangle$ 
26     $e_t^a = \langle w_t, \pi_k^a - p_t \rangle + \epsilon_t$ 
27     $\sigma_t = (1 - \alpha_t) \langle v_t - w_t, p_t - \pi_t \rangle$ 
28  faça  $t = t + 1$  e faça um loop a partir de passo1.
29 fin

```

Figure 1: Algoritmo do método do volume revisado para o problema lagrangeano  $\theta(\pi)$

incidência  $A$ . A coluna 4 representa a densidade da matriz  $A$  que é calculado pelo quantidade de 1's dessa matriz dividido pela quantidade total de elementos de  $A$  que é igual a  $mn$  e a coluna 5 mostra a quantidade de problemas em cada conjunto. Os dez problemas do conjunto 4 são nomeados como scp41-scp410, os cinco problemas do conjunto 6 são nomeados como scp61-scp65, e os problemas do conjunto A e B são nomeados respectivamente como scp41-scp45 e scp61-scp65.

Conjunto	Linhas	Colunas	Densidade	Problemas
4	200	1000	2	10
6	200	1000	5	5
A	300	3000	2	5
B	300	3000	5	5

Table 1: Detalhes das instâncias de testes utilizadas

No experimento desse trabalho foi comparado a performance do modelo matemático para o *SCP* que será chamado aqui de *IP*, com o algoritmo proposto na seção 2, que será chamado *VolumeRevisado*. O modelo *IP* foi executado através do CPLEX com todos os parâmetros default. O modelo presente no algoritmo *VolumeRevisado* foi executado pelo CPLEX também com os valores default. O algoritmo *VolumeRevisado* foi executado com os parâmetros  $m_1 = 0.0001$ ,  $\delta_{min} = 0.00001$ , com um número máximo de iterações igual a 500000. O modelo *IP* e o algoritmo *VolumeRevisado* foram executados com um tempo de execução máximo de 7200 segundos.

A tabela 2 apresenta os resultados obtidos para o conjuntos de instância 4 e 6 e a tabela 3 apresenta os resultados para o conjuntos de instância A e B. Nessas tabelas a coluna 1 mostra o nome da instância de teste, as colunas 2 e 3 são resultados referentes ao modelo *IP* e as colunas 4,5 e 6 são resultados referentes ao algoritmo *VolumeRevisado*. A coluna 2 apresenta o custo da solução obtido pela modelo *IP* e a coluna 3 apresenta o tempo consumido para encontrar essa solução. A coluna 4 apresenta o custo da solução obtido pelo algoritmo *VolumeRevisado*, a coluna 5 traz o número de iterações do algoritmo e a coluna 6 mostra o tempo consumido pelo algoritmo *VolumeRevisado*.

Para todas as instâncias do conjunto 4 e 6 o CPLEX conseguiu encontrar soluções ótimas em um tempo muito pequeno, conforme pode ser observado pelas colunas 2 e 3 da tabela 2. Para as instâncias scp41, scp46 e scp410 o algoritmo *VolumeRevisado* parou com uma solução de

custo muito próxima da solução do modelo matemático, conforme pode ser observado na coluna 4, linhas 1,6 e 10 da tabela 2.

Instância	<i>IP</i>		<i>VolumeRevisado</i>		
	Custo Solução	Tempo(s)	Custo Solução	#Iterações	Tempo(s)
scp41	429	0.84	428.21	38	7.84
scp42	512	0.85	492.33	35436	7200.00
scp43	516	0.86	479.98	34061	7200.00
scp44	494	0.86	480.21	34520	7200.00
scp45	512	0.85	499.30	35192	7200.00
scp46	560	0.89	559.10	41	8.24
scp47	430	0.83	295.89	34368	7200.00
scp48	492	0.97	467.75	34983	7200.00
scp49	641	0.90	443.61	34813	7200.00
scp410	514	0.90	513.06	31	6.31
scp61	138	1.23	105.98	34797	7200.00
scp62	146	2.06	76.75	35879	7200.00
scp63	145	1.26	83.10	35389	7200.00
scp64	131	0.96	127.38	35493	7200.00
scp65	161	1.94	118.19	35540	7200.00

Table 2: Comparação entre os custos da solução e tempos obtidos entre o modelo *IP* e o algoritmo *VolumeRevisado* para as instâncias do conjunto 4 e 6.

Para todas as instâncias do conjunto A e B o CPLEX conseguiu encontrar soluções ótimas em um tempo pequeno, conforme pode ser observado pelas colunas 2 e 3 da tabela 3. Para essas instâncias o algoritmo *VolumeRevisado* não conseguiu encontrar a solução ótima ou próxima da ótima para nenhuma delas, conforme pode ser observado pela coluna 4 da tabela 3.

Instância	<i>IP</i>		<i>VolumeRevisado</i>		
	Custo Solução	Tempo(s)	Custo Solução	#Iterações	Tempo(s)
scpa1	253	9.95	153.88	9859	7200.00
scpa2	252	9.87	209.80	10010	7200.00
scpa3	232	9.48	208.89	9804	7200.00
scpa4	234	8.76	207.49	10025	7200.00
scpa5	236	8.64	186.64	10006	7200.00
scpb1	69	10.08	55.89	10078	7200.00
scpb2	76	10.87	45.80	9867	7200.00
scpb3	80	9.67	55.92	9820	7200.00
scpb4	79	11.52	66.59	10024	7200.00
scpb5	72	9.86	53.16	10012	7200.00

Table 3: Comparação entre os custos da solução e tempos obtidos entre o modelo *IP* e o algoritmo *VolumeRevisado* para as instâncias do conjunto A e B.

Todo código fonte produzido por esse trabalho pode ser obtido no endereço eletrônico: [https://github.com/joaojunior/volume\\_revisado2scp](https://github.com/joaojunior/volume_revisado2scp)



## References

- [1] Balas, Egon and Ng, ShuMing. On the Set Covering Polytope: I. All the Facets with coefficients in  $\{0,1,2\}$ . *Mathematical Programming*, 43:57–69, 1989.
- [2] L Bahiense, N. Maculan and C. Sagastizábal. The volume algorithm revisited: relation with bundle methods. *Mathematical Programming*, 94:41–69, 2002.
- [3] Beasley, J. E. OR-Library: distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41:1069–1072, 1990.