

Um método exato para multiplicadores lagrangeano/surrogate

Marcelo G. Narciso

Embrapa Informática Agropecuária

Av. Dr. André Tosello, s/n, Unicamp

13083-970 – Campinas - SP, Brazil

narciso@cnptia.embrapa.br

Luíz A. N. Lorena

LAC/INPE- Instituto Nacional de Pesquisas Espaciais

Caixa Postal 515

12201-970 - São José dos Campos – SP, Brazil

lorena@lac.inpe.br

1 - Introdução

A relaxação *lagrangeana/surrogate* (*lagsur*) foi proposta como uma alternativa à relaxação *lagrangeana*, substituindo-a com vantagens em diversos problemas de otimização combinatória. As relaxações *lagrangeana* [7] e *surrogate* [6,10] são combinadas com objetivo de conseguir melhores tempos computacionais na aplicação de heurísticas baseadas em relaxações que usem métodos subgradientes [11,13].

Na formação de uma relaxação *lagsur*, para um dado problema de otimização combinatória, inicialmente é derivada uma relaxação *surrogate* de um conjunto adequado de restrições. Uma relaxação *lagrangeana* da restrição *surrogate* é então obtida. Vale a pena mencionar que neste caso o multiplicador *lagrangeano* é unidimensional.

A relaxação *lagsur* foi inspirada nos trabalhos de Lorena e Lopes [12] e Lorena e Narciso [13], observando-se os resultados da aplicação das relaxações *surrogate contínua* e *lagrangeana* para o problema de cobertura de conjuntos (PCC) e o *problema generalizado de atribuição (PGA)*, respectivamente. Foi constatado que na aplicação de um método de otimização por subgradientes, usando o mesmo multiplicador inicial, sobre os mesmos dados, a relaxação *surrogate contínua* proporcionava uma seqüência de limites mais estável que a correspondente da relaxação *lagrangeana*, convergindo mais rapidamente. Além

disso, os limites obtidos com a aplicação da relaxação *surrogate contínua* eram tão bons quanto os obtidos com a relaxação *lagrangeana*.

Conforme pode ser constatado nos resultados relativos às instâncias *PGA* [15], a relaxação *lagsur* é apropriada para problemas com um grande número de variáveis. Quanto maior for o número de variáveis do problema, melhor será o desempenho da relaxação *lagsur* em relação à relaxação *lagrangeana* no que se refere a tempo de execução.

A relaxação *lagsur* é usualmente resolvida usando-se uma heurística de subgradientes [15]. Em uma dada iteração, para um dado multiplicador multi-dimensional *surrogate*, faz-se uma busca para o encontrar o multiplicador *lagrangeano* unidimensional.

Este processo de busca se coloca como um desafio ao uso eficiente de uma relaxação *lagsur*. A relaxação *lagrangeana* usual considera t fixado no valor 1 (um) em todas as iterações do método de otimização por subgradientes. Um processo demorado de busca pelo melhor t , pode refletir no tempo geral a ser gasto na solução do dual, e favorecer ao uso da relaxação *lagrangeana*.

Uma implementação do *lagsur*, mostrada nos trabalho de Narciso e Lorena [15], conseguiu bater em tempos computacionais a correspondente implementação *lagrangeana* para um conjunto de instancias do *PGA*. Nestas o valor de t é calculado com uma busca abreviada, nas primeiras iterações do método subgradientes, e posteriormente é fixado no último valor obtido nas demais iterações do método. Propõe-se neste trabalho um novo algoritmo para calcular o melhor t de maneira exata, quando aplicado ao *PGA*, considerando a relaxação das restrições de capacidades. Em testes computacionais verificou-se que este novo algoritmo proporcionou a uma implementação da relaxação *lagsur* ainda melhor que a do trabalho de Narciso e Lorena [15].

2. O Problema Generalizado de Atribuição

O problema de se maximizar o lucro (benefício, rendimento, etc.) ao se atribuir n tarefas a m agentes, $n > m$, tal que cada tarefa seja atribuída a apenas um único agente, levando-se em conta as restrições de capacidade de cada agente, é conhecido na literatura como *problema generalizado de atribuição (PGA)*. Muitos problemas da vida real podem ser modelados como um *PGA*. Podemos citar como exemplos, problemas de investimento de capitais [4], alocação de espaço de memória [3], projeto de redes de comunicação com restrições de capacidades para cada nó da rede [4], atribuição de tarefas de desenvolvimento de software a programadores [1], atribuição de tarefas a computadores em uma rede [3], subproblemas de roteamento de veículos [8], e outros. Este problema é NP-

hard [3, 9, 16] e poucas heurísticas existem na literatura para a resolução deste problema de forma a encontrar boas soluções viáveis.

O *PGA* é formulado matematicamente como:

$$\begin{aligned}
 v(PGA) &= \text{Max} \sum_{i=1}^m \sum_{j=1}^n p_{ij} x_{ij} \\
 (PGA) \quad \text{sujeito a} & \sum_{j=1}^n w_{ij} x_{ij} \leq b_i, \quad i \in M = \{1, \dots, m\} \quad (4.1)
 \end{aligned}$$

$$\sum_{i=1}^m x_{ij} = 1, \quad j \in N = \{1, \dots, n\} \quad (4.2)$$

$$x_{ij} \in \{0,1\}, \quad i \in M, j \in N \quad (4.3)$$

As restrições (4.1) impõem que os pesos dos itens escolhidos não devem exceder à capacidade de cada mochila e as restrições (4.2) impõem que cada item deve ser atribuído a somente uma mochila.

Vamos considerar a relaxação das restrições (4.1) conforme o modo *lagsur*. Seja λ um vetor de multiplicadores *surrogate* $\lambda_i \geq 0, i \in M$ e um multiplicador *lagsur* t , real e não nulo. Matematicamente, esta relaxação é escrita como:

$$\begin{aligned}
 v(L_t S_1 PGA) &= \text{Max} \sum_{i=1}^m \sum_{j=1}^n [(p_{ij} - t \cdot I_i w_{ij}) x_{ij}] + t \cdot \sum_{i=1}^m I_i b_i \\
 (L_t S_1 PGA) \quad \text{sujeito a} & \quad (4.2) \text{ e } (4.3)
 \end{aligned}$$

O dual local *lagsur*, em relação ao multiplicador t , é dado por

$$(DL_t S_1 cPGA) = \text{Min}_{t \geq 0} v(L_t S_1 PGA)$$

A *figura 1* mostra um comportamento típico da função $v(L_t S_1 PGA)$, para um dado λ fixo (linear por partes).

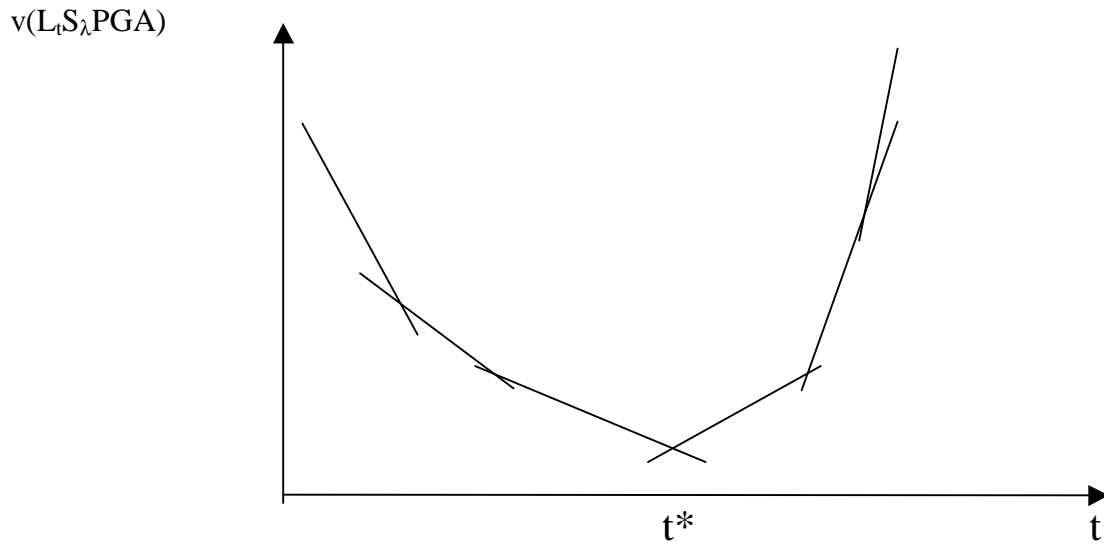


Figura 1- Comportamento da função $v(L_t S_\lambda PGA)$ versus t

3 – Nova proposta de busca para o multiplicador lagsur

Como a função *lagsur* é linear por partes [15], no ponto de mínimo, teremos dois segmentos de reta se encontrando, e o valor de t será o mesmo para os dois segmentos (veja figura 1). Entretanto, o valor de x será diferente, bem como os gradientes, para cada um dos segmentos de reta. O gradiente à esquerda de t será negativo e à direita, positivo. No ponto de encontro, seja x_1 tal que o gradiente de $v(L_t S_\lambda PGA)$ é negativo e x_2 , tal que o gradiente seja positivo. Desta forma, tem-se o seguinte:

$$\sum_{i=1}^m \sum_{j=1}^n [(p_{ij} - t \cdot I_i w_{ij}) \cdot x_{1ij}] + t \cdot \sum_{i=1}^m I_i b_i = \sum_{i=1}^m \sum_{j=1}^n [(p_{ij} - t \cdot I_i w_{ij}) \cdot x_{2ij}] + t \cdot \sum_{i=1}^m I_i b_i$$

Isolando-se t na equação, tem-se o seguinte resultado:

$$t = \frac{\sum_{i=1}^m \sum_{j=1}^n [p_{ij} (x_{2ij} - x_{1ij})]}{\sum_{i=1}^m \sum_{j=1}^n [I_i w_{ij} (x_{2ij} - x_{1ij})]},$$

Desta forma, para se obter o valor exato de t , é necessário descobrir os valores de x_1 e x_2 . Para isto, sabendo-se que o gradiente de $v(L_t S_\lambda PGA)$, em relação a t , é dado por

$$g(t, x) = \sum_{i=1}^m I_i (b_i - \sum_{j=1}^n w_{ij} x_{ij})$$

para x_1 teríamos $g(t, x_1) < 0$ e para x_2 , $g(t, x_2) > 0$. Para se obter o ponto de mínimo da curva. A idéia básica é a seguinte:

- retirar as variáveis dominadas em cada coluna. Estas variáveis não fazem parte da solução, qualquer que seja o valor de t . Para mais detalhes, ver [5] e [14];
- dentre as variáveis não dominadas, escolher em cada coluna um item e assim formar uma solução inicial. Em seguida, verificar se $g(t,x)$ é maior ou menor que zero. Se for maior que zero, modificar a escolha dos itens das variáveis não dominadas até que se obtenha o gradiente menor que zero. Feito isto, tem-se os valores de x_1 (solução na qual $g(t, x_1) < 0$) e x_2 (solução na qual $g(t, x_2) > 0$). Os valores de x_1 e x_2 irão diferir em apenas uma posição (coluna). O mesmo raciocínio vale para o caso do primeiro valor de solução na qual $g(t,x)$ é menor que zero.

Para se obter as variáveis não dominadas, procede-se da seguinte maneira:

- 1) Sejam i, k, j, s , inteiros. Se $w_{ij} \leq w_{kj}$ e $p_{ij} \leq p_{kj}$ então $x_{kj} = 0$
- 2) Se $w_{ij} \leq w_{kj} \leq w_{sj}$ e $p_{ij} \leq p_{kj} \leq p_{sj}$ então $x_{kj} = 0$

Pelo menos metade das variáveis são eliminadas, facilitando a resolução do problema.

No processo de busca do melhor t , procede-se como descrito acima e obtém-se os valores de x_1 e x_2 , e com estes valores, obtém-se o melhor multiplicador *lagsur*. O algoritmo para a busca do melhor t seria:

- 1 - Retirar as variáveis dominadas em cada coluna;
- 2 - Das variáveis que sobraram, obter uma solução inicial para $v(t,x)$
- 3 - Repita enquanto não fim
- 4 - Se $g(t,x) < 0$ então
 Substituir os maiores valores de w_{ij} relativos à solução encontrada por valores menores de tal que o valor de p_{ij} seja o menor possível
 Se, após uma dada substituição, $g(t,x) > 0$, então fim
 Fim_Se
- 5- Se $g(t,x) > 0$ então
 Substituir os menores valores de w_{ij} relativos à solução encontrada por valores maiores de tal que o valor de p_{ij} seja o menor possível
 Se, após a substituição, $g(t,x) > 0$, então fim
 Fim_Se
 Fim do Repita

$$6 - t = \frac{\sum_{i=1}^m \sum_{j=1}^n [p_{ij} (x_{2ij} - x_{1ij})]}{\sum_{i=1}^m \sum_{j=1}^n [I_i w_{ij} (x_{2ij} - x_{1ij})]},$$

- 7- Fim

Nesta nova forma de busca exata, em geral, tem-se um menor tempo do que quando o multiplicador *lagsur* é calculado de forma aproximada, considerando o *PGA* onde foram relaxadas as restrições de capacidades.

4 -Resultados

Para verificar o comportamento das relaxações *lagsur* exata e *lagsur* aproximada [como foi descrito em [15]] com o *PGA*, foram usadas instâncias obtidas da *OR-Library* [2]. Os resultados obtidos estão dispostos em tabelas (*tabelas 1 e 2* a seguir). Cada tabela contém instâncias consideradas larga escala. Foram utilizados 24 tipos de problemas com dimensões diferentes. Estes problemas são conhecidos como problemas de classes A, B, C, e D. As dimensões de cada um destes problemas são: (5 x 100), (5 x 200), (10 x 100), (10 x 200), (20 x 100) e (20 x 200). Estes testes foram executados em uma máquina SUN SPARC 5, com compilador C. A notação em cada coluna é a seguinte:

tempo := tempo final, em segundos, obtido pela heurística;
 iter. := número de vezes que a relaxação lagrangeana/lagsur foi resolvida;
 gap1 := ((solução ótima) – melhor solução viável obtida) / (solução ótima) ;
 gap2 := (melhor valor da relaxação lagsur - (solução ótima)) / (solução ótima) ;
 gap2% = 5%,...,0.5% := tempo computacional gasto para o gap2, em porcentagem, ser 5%, 4%, 3%, 2%, 1% e 0.5%.

prob	t	gap1 (10 ⁻³)	gap2 (10 ⁻³)	n_iter	5%	4%	3%	2%	1%	0.5%
A5x100	0.310	0.291	0.000	53	0.160	0.160	0.160	0.160	0.160	0.160
A5x200	0.810	0.229	0.000	47	0.580	0.580	0.580	0.580	0.580	0.580
A10x100	0.570	2.311	1.489	59	0.270	0.270	0.270	0.270	0.270	0.270
A10x200	1.300	0.426	0.000	50	0.980	0.980	0.980	0.980	0.980	0.980
A20x100	1.040	0.919	0.000	54	0.490	0.490	0.490	0.490	0.500	0.530
A20x200	3.350	0.999	0.00	66	2.050	2.050	2.050	2.050	2.050	2.050
B5x100	1.100	123.7	108.5	222	0.140	0.140	0.140	0.150	--	--
B5x200	2.390	3.641	2.117	176	0.580	0.580	0.580	0.580	0.580	0.580
B10x100	1.700	7.264	5.828	229	0.280	0.280	0.280	0.280	0.280	0.330
B10x200	5.650	3.523	2.701	306	1.080	1.080	1.080	1.080	1.080	1.080
B20x100	1.210	12.82	9.134	61	0.540	0.540	0.540	0.540	0.580	0.870
B20x200	7.730	4.762	2.585	225	2.160	2.160	2.160	2.160	2.160	2.160
C5x100	0.930	2.851	1.134	183	0.140	0.140	0.140	0.140	0.140	0.150
C5x200	2.400	8.638	7.068	173	0.570	0.570	0.570	0.570	0.570	0.570
C10x100	2.910	7.485	1.327	326	0.280	0.280	0.280	0.280	0.290	---
C10x200	4.300	3.863	1.622	221	1.070	1.070	1.070	1.070	1.070	1.070
C20x100	1.720	17.05	11.29	65	0.540	0.540	0.540	0.540	0.580	---
C20x200	9.730	5.323	2.185	250	2.150	2.150	2.150	2.150	2.150	2.210
D5x100	0.350	1.083	0.000	77	0.140	0.140	0.140	0.140	0.140	0.140
D5x200	1.370	1.211	0.000	139	0.550	0.550	0.550	0.550	0.550	0.550
D10x100	0.790	1.548	0.000	108	0.270	0.270	0.270	0.270	0.270	0.270
D10x200	2.620	1.631	0.000	139	1.080	1.080	1.080	1.080	1.080	1.080
D20x100	1.850	1.862	0.000	139	0.550	0.550	0.550	0.550	0.550	0.550
D20x200	5.340	1.855	0.000	170	2.170	2.170	2.170	2.170	2.170	2.170

Tabela 1 - Lagsur aproximado para problemas das classes A, B, C e D.

Vale a pena mencionar que na relaxação *lagsur* aproximada, o valor de t é calculado por uma busca unidimensional aproximada, e após a busca, quando o valor encontrado for

repetido em 5 iterações seguidas, fixa-se o valor de t para as demais iterações [15]. Para o *lagsur* exato, usamos o mesmo procedimento de fixação.

5 - Comentários sobre os resultados obtidos para o PGA

Os testes refletem de uma forma mais intensa o ganho em tempo computacional da relaxação *lagsur* com o valor de melhor t exato em comparação aos tempos da relaxação *lagsur* aproximada. Isto ocorre devido ao fato de se obter um valor de t de melhor qualidade, o qual faz mais rapidamente a aproximação dos valores duais aos da solução ótima, do que o pontos obtidos pelo dual *lagsur* aproximado.

A busca feita neste novo método é mais rápida devido ao número de *loops* ser menor que o da busca aproximada do multiplicador *lagsur* e também pelo fato do número de instruções aritméticas neste novo método ser menor. Em suma, o novo método, além de fornecer um valor exato para o multiplicador t , é menos complexo. Entretanto, não é possível fazer a busca em todas as iterações do dual, pois mesmo sendo mais rápido que ao busca aproximada, cada busca leva mais tempo de ser realizada do que a resolução do *lagrangeano tradicional*, pois este se resume a um único *loop*, onde é escolhido o maior valor do lucro/custo em cada coluna.

prob	t	gap1 (10^{-3})	gap2 (10^{-3})	n_iter	5%	4%	3%	2%	1%	0.5%
A5x100	0.190	0.291	0.000	34	0.830	0.830	0.830	0.830	0.830	0.830
A5x200	0.690	0.106	0.000	34	0.370	0.370	0.370	0.370	0.370	0.460
A10x100	0.480	2.311	1.489	40	0.220	0.220	0.220	0.220	0.220	0.220
A10x200	1.490	0.426	0.160	44	0.980	0.980	0.980	0.980	0.980	0.980
A20x100	0.950	0.919	0.000	54	0.370	0.370	0.370	0.370	0.430	0.490
A20x200	3.400	0.990	0.000	66	1.870	1.870	1.870	1.870	1.870	1.960
B5x100	0.860	2.921	0.000	191	0.100	0.100	0.100	0.100	0.100	0.120
B5x200	2.100	2.697	1.176	150	0.480	0.480	0.480	0.480	0.480	0.480
B10x100	1.600	6.418	1.489	207	0.220	0.220	0.220	0.220	0.220	0.220
B10x200	5.140	3.640	2.809	277	0.990	0.990	0.990	0.990	0.990	0.990
B20x100	0.730	12.64	8.927	37	0.390	0.390	0.390	0.390	0.390	0.430
B20x200	7.100	4.859	2.689	207	1.850	1.850	1.850	1.850	1.850	1.850
C5x100	0.810	2.846	1.587	162	0.105	0.105	0.105	0.105	0.105	0.105
C5x200	2.100	3.421	2.272	141	0.430	0.430	0.430	0.430	0.430	0.430
C10x100	2.270	7.503	2.871	270	0.240	0.240	0.240	0.240	0.240	0.540
C10x200	3.900	3.978	1.730	185	0.920	0.920	0.920	0.920	0.920	0.920
C20x100	1.370	16.85	11.29	44	0.410	0.410	0.410	0.410	0.410	---
C20x200	8.270	5.231	2.081	217	1.500	1.500	1.500	1.500	1.500	1.950
D5x100	0.390	1.418	0.000	99	0.130	0.130	0.130	0.130	0.170	0.190
D5x200	1.310	0.214	0.000	118	0.490	0.490	0.490	0.490	0.540	0.590
D10x100	0.580	0.761	0.000	91	0.190	0.190	0.190	0.250	0.290	0.320
D10x200	2.270	0.307	0.000	119	1.080	0.890	0.890	0.890	0.890	0.890
D20x100	1.300	0.500	0.000	90	0.410	0.410	0.410	0.410	0.650	0.750
D20x200	4.750	0.291	0.000	153	1.300	1.300	1.300	2.050	2.050	2.150

Tabela 2 - *Lagsur* exato para problemas das classes A, B, C e D.

Agradecimentos

O segundo autor agradece o apoio financeiro da FAPESP - Fundação para o Amparo à Pesquisa no Estado de São Paulo (proc. 96/04585-6), e também o apoio recebido do CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico (proc. 350034/91-5).

Referencias bibliográficas

- [1] Balachandran, V. An integer generalized transportation model for optimal job assignment in computers networks. **Operations Research**, v. 24, n.4, p. 742-749, 1976.
- [2] Beasley, J. E. OR-Library: Distributing test problems by eletronic mail. **Journal of Operational Research Society**, v. 41,n. 11, p. 1069-1072, 1990.
- [3] Catrysse, D. Van Wassenhove, L. N. A survey of algorithms for the Generalized Assignment Problem. **European Journal of Operational Research**. v. 60, p. 260-272, 1992.
- [4] De Maio, A., Roveda, C. An all zero-one algorithm for a certain class of transportation problems. **Operations Research**. v. 19, p. 1406-1418, 1971.
- [5] Dudzinsky, k., Waluckiewicz, S. A fast algorithm for the linear multiple-choice knapsack problem. **Operations Research Letters**. v. 3, n. 4, p. 205-209, 1984.
- [6] Dyer, M. E. Calculating surrogate constraints. **Mathematical Programming**. v. 19, p. 255-278, 1980.
- [7] Fisher, M. L. The lagrangian relaxation method of solving integer programming problems. **Management Science**. v. 27, p. 1-18, 1981.
- [8] Fisher, M. L., Jaikumar, R. A generalized assignment heuristic for vehicle routing. **Networks**. v. 11, p. 109-124, 1981.
- [9] Fisher, M. L., Jaikumar, R., Wassenhove, L. N. V. A multiplier adjustment method for the generalized assignmet problem. **Management Science**. v. 32, p. 1095-1103, 1986.
- [10] Greenberg, H. J., Pierskalla, W. P. Surrogate Mathematical Programming. **Operations Research**. v. 18, p. 924-939, 1970.
- [11] Lopes, F. B. Nova heurística para o problema de cobertura de conjuntos.(Tese de Mestrado em Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais. São José dos Campos, 1992. (INPE - 5471 - TDI/502).
- [12] Lorena, L. A. N., Lopes, F. B. A surrogate heuristic for set covering problems. **European Journal of Operational Research**. v. 79, n. 1, p. 138-150, 1994.
- [13] Lorena, L. A. N., Narciso, M. G. Relaxation heuristics for a generalized assignment problem. **European Journal of Operational Research**. v. 91, n. 1, p. 600-610, 1996.

- [14] Narciso, M. G. Novas heurísticas para o problema generalizado de atribuição". INPE, 1994. 158p. Dissertação de Mestrado.
- [15] Narciso, M. G., Lorena, L. A.N. Lagrangean/surrogate relaxation for generalized assignment problem. **European Journal of Operational Research**. 114(1), 165-177, 1999
- [16] Ross, G. T., Soland, M. S. A branch and bound algorithm for the generalized assignment problem. **Mathematical Programming**, v. 8, p. 91-103, 1975.