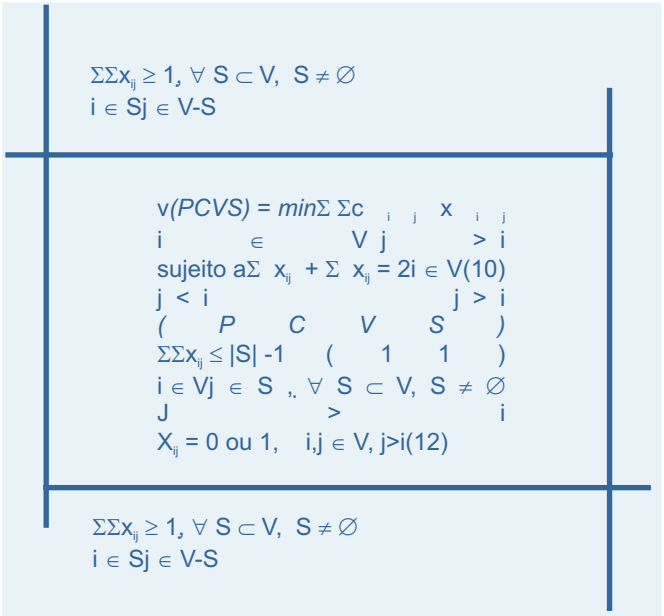
Comunicado Técnico 85

Dezembro, 2007
Campinas, SP



ISSN 1677-8464

Relaxação Lagsur como alternativa para obtenção de soluções viáveis de boa qualidade para os problemas da otimização combinatória

Marcelo Gonçalves Narciso[1]

Uma relaxação pode ser definida como a retirada ou transformação de algumas restrições com a finalidade de facilitar a resolução do problema original. Porém, a solução final obtida não necessariamente atenderá às restrições que foram retiradas ou transformadas. Entretanto, a solução obtida pode ser considerada um limite inferior ou superior, conforme o problema, e este estar muito próximo da solução ótima. A relaxação, portanto, fornece um valor que serve para dar uma noção de quão longe ou perto está uma solução viável da solução ótima. Existem várias relaxações conhecidas, como por exemplo a relaxação Lagrangeana, relaxação Surrogate e a relaxação Lagrangeana Surrogate, que é uma combinação das duas (Narciso & Lorena, 1999).

A relaxação Lagsur (Narciso & Lorena, 1999) mostra como se obter limites superiores (se o problema for de maximização) ou inferiores (se o problema for de minimização) muito próximos da solução ótima. Esta proposta veio a melhorar a relaxação Lagrangeana, descrita amplamente na literatura (Lorena & Narciso, 1996), diminuindo-se o tempo de execução desta, mantendo-se ou melhorando a qualidade da solução obtida.

Embora a solução dada pela relaxação não seja viável (factível), ela pode ser usada para fornecer uma solução viável do problema considerado. Para isto, pequenas modificações no algoritmo da relaxação devem ser feitas para que este venha também a fornecer solução viável de qualidade. Esta inserção no algoritmo será mostrada neste trabalho, bem como os resultados obtidos. O método aqui mostrado pode ser estendido para outros problemas, particularmente nos problemas formulados como de *Programação Linear Inteira zero-um.* Como estudo de caso, neste trabalho serão enfocados o Problema do Caixeiro Viajante (PCV), descrito por Lawler et al. (1985), e o Problema de Atribuição Generalizado (PAG), pesquisado por Narciso & Lorena (1999).

Relaxação Lagrangeana Surrogate (Lagsur)

A relaxação Lagrangeana tem sido empregada há muito tempo, com grande sucesso, como auxiliar no desenvolvimento de métodos para a busca de soluções ótimas aos problemas de Otimização Combinatória. As soluções das relaxações fornecem limites, que em conjunto com outros limites de soluções viáveis aos problemas, proporcionam indicações do valor dessas soluções viáveis. Podem participar de métodos enumerativos, ou no desenvolvimento de heurísticas. Em geral seu emprego está relacionado ao uso de um

[1]Doutor em Computação Aplicada, Pesquisador da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: narciso@cnptia.embrapa.br)

Embrapa

método dual de otimização por subgradientes (Poljak, 1969), também de consagrado uso e conhecimento para pesquisadores de métodos de otimização.

A relaxação *Surrogate* foi introduzida na programação matemática por Glover (1968). Basicamente, esta relaxação consiste em tomar um conjunto de restrições do problema original e, usando um vetor de multiplicadores, transformar estas restrições em apenas uma restrição. Esta nova restrição é chamada de restrição *Surrogate*.

No trabalho de Lopes & Lorena (1994), tem-se uma boa comparação do comportamento das duas relaxações para o problema de Cobertura de Conjuntos. Os resultados foram muito bons quanto ao uso da relaxação Surrogate, mas havia uma ressalva de que esta era difícil de se implementar.

As relaxações Lagrangeana e Surrogate, em 1997, foram combinadas para então dar origem a relaxação Lagsur, a qual tinha a grande vantagem de ser mais rápida, em termos de tempo de execução, do que a relaxação Lagrangeana e também mais fácil de se implementar do que a relaxação Surrogate. Mais detalhes sobre esta relaxação podem ser visto em Narciso & Lorena (1999).

As relaxações descritas anteriormente apenas calculavam um valor de limite para a solução ótima, e a solução obtida não era viável. Para entender melhor a relaxação, observar o problema (P) a seguir:

$$v(P) = \min cx$$
$$(P) \text{ sujeito a } Ax = b$$
$$Dx \leq f$$
$$X \geq 0 \text{ e inteiro.}$$

No problema (P) anterior, A e D são matrizes e b e f são vetores. Suponha que a restricao **Dx** $\leq$ **e** seja escolhida para ser relaxada.

Este problema, ao ser relaxado conforme o **modo Lagrangeano**, ficaria da seguinte forma:

$$v(L_\lambda) = \min (c - \lambda.D).x + \lambda.e \quad (\lambda \text{ é um vetor})$$
$$(L_\lambda) \text{ sujeito a } Ax = b$$
$$x \geq 0 \text{ e inteiro.}$$

O problema (P), ao ser relaxado conforme o **modo Surrogate**, ficaria da seguinte forma:

$$v(S_t) = \min (c - t.\Sigma D).x + t.\Sigma e \quad (t \text{ é um escalar})$$
$$(S_t) \text{ sujeito a } Ax = b$$
$$X \geq 0 \text{ e inteiro.}$$

No problema anterior, Σ significa um somatório. Na Relaxação Surrogate descrita, a restrição **Dx** $\leq$ **e** é somada e então relaxada por um escalar t, isto é, é adicionada à função objetivo. Na relaxação Lagrangeana, as restrições não são somadas conforme foi feito na relaxação Surrogate e o multiplicador (vetor) λ usado para multiplicar cada linha da restrição Dx $\leq$ e. No caso da relaxação Surrogate, é usado um escalar (t). Mais detalhes podem ser vistos em Narciso & Lorena (1999).

O problema (P), ao ser relaxado conforme o modo Lagsur, ficaria da seguinte forma:

$$v(L_tS_\lambda) = \min (c - t.\lambda.D).x + t.\lambda.e$$
$$(L_tS_\lambda) \text{ sujeito a } Ax = b$$
$$x \geq 0 \text{ e inteiro.}$$

O vetor λ pode conter valores nulos ou positivos somente e t é um escalar. O valor de t é obtido através de uma otimização, conforme o modo Surrogate e o valor de λ varia iterativamente, conforme o modo lagrangeano. Mais detalhes podem ser consultados em Narciso & Lorena (1999). Observa-se que a restrição Ax = b foi relaxada, isto é, foi retirada das restrições e adicionada à solução objetivo.

O algoritmo Lagsur para se obter este limite e a conseqüente solução não viável está descrito a seguir:

Algoritmo Lagsur para problema de minimizacao
Seja $\lambda \geq 0$
Fazer lb $\leftarrow -\alpha$
Obter *uma solução viável* x_v *para (P). Seja ub este valor.*
Enquanto {nenhuma condição de parada for atendida} ***fazer***
Resolver *o problema relaxado* $v(Rel_\lambda)$. **Seja** $x_\lambda \in J$ a solução obtida.
Fazer lb = *máx* (lb, $v(Rel_\lambda)$),
Atualizar o multiplicador λ (λ = u.(ub - lb)||d||)
Testar condições de parada
Fim_enquanto.

No algoritmo descrito, $v(Rel_\lambda)$ é resolvido inicialmente com um valor λ combinado com um valor de t, o qual é um escalar positivo. Este valor de t, após algumas iterações, fica fixo e então somente varia o valor de λ. Informações detalhadas sobre como calcular ||d|| estão descritas em Lopes & Lorena (1994), e sobre como calcular t podem ser consultadas em Narciso & Lorena (1999).

Para que este algoritmo seja adaptado para fornecer soluções viáveis de boa qualidade, observar o exemplo a seguir:

Algoritmo Lagsur para problema de minimização - versão II

Seja $\lambda \geq 0$,

Fazer lb $\leftarrow -\infty$ e ub $\leftarrow +\infty$

Enquanto {não para as condições de parada} ***fazer***

Resolver (Rel_λ). **Seja** $x_\lambda \in J$ a solução obtida.

Obter uma solução viável x_v para *(P)* a partir de x_λ e o valor $c.x_v$,

Fazer lb = *máx* (lb, v*(Rel$_\lambda$)*),

ub = *min* (ub, $c.x_v$).

Atualizar o multiplicador λ ($\lambda = \lambda u.(ub - lb)/||d||$)

Testar condições de parada,

Fim_enquanto.

Quanto ao cálculo da solução viável x_v, basta substituir o valor da função objetivo CX, descrito anteriormente no problema (P), tomado como exemplo, por uma pequena alteração em C, isto é, $C(\lambda.t)X$, ou seja, deve ser tal como é feito para a relaxação Lagrangeana (Lopes & Lorena, 1994). Assim, o valor da função objetivo vai variar conforme o valor de λ de t variem. Em cada iteração, vai sendo computado o melhor valor obtido e assim, ao final do algoritmo referente ao Lagsur, tem-se a melhor solução viável obtida. Isto será exemplificado com os problemas PAG e PCV, neste documento.

Problema de Atribuição Generalizado

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 atribuição generalizado *(PAG)*. Muitos problemas da vida real podem ser modelados como um *(PAG)*. Pode-se citar como exemplos, problemas de investimento de capitais, alocação de espaço de memória, projeto de redes de comunicação com restrições de capacidades para cada nó da rede, atribuição de tarefas de desenvolvimento de software para programadores, atribuição de tarefas a computadores em uma rede, subproblemas de roteamento de veículo, e outros. Como uma conseqüência lógica de sua aplicação, um grande esforço tem sido realizado no desenvolvimento de algoritmos para este problema Narciso & Lorena (1999).

O PAG é NP-hard (Catrysse & Van Wassenhove, 1992) e matematicamente pode ser descrito como

$$v(PGA) = max \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij}.x_{ij} \quad (1)$$

(PGA)

$$\text{sujeito a} \quad \sum_{j=1}^{n} w_{ij}.x_{ij} \le b_i, \ i \in M = \{1,...,m\} \quad (2)$$

$$\sum_{i=1}^{M} x_{ij} = 1, \ j \in N = \{1,...,n\} \quad (3)$$

$$xij \in \{0,1\}, i \in M, j \in N \quad (4)$$

Na formulação anterior, x_{ij} indica se o item foi escolhido ($x_{ij} = 1$) ou não ($x_{ij} = 0$). O lucro de se escolher o item é indicado por p_{ij}. O peso ou custo por ter escolhido o lucro p_{ij} é dado por w_{ij}. O limite das somas dos custos dos pesos escolhidos é dado por b_i. Um exemplo deste problema seria a distribuição de n tarefas para m processadores, cada um com uma capacidade b_i, e esta capacidade não pode ser estourada pela capacidade consumida (custos) de cada tarefa, e isto está indicado pela desigualdade (2). A equação (3) indica que cada tarefa deve ser direcionada para apenas 1 processador, isto é, não pode ser fracionada. Assim, x_{ij} poderá ser 0 ou 1 somente.

Uma heurística simples, que pode ser usada para se obter soluções viáveis para este problema, pode ser a seguinte:

1. em cada coluna, escolher a atribuição de um item ($x_{ij} = 1$) com o menor peso possível (w_{ij}) sem que estoure a capacidade b_i;
2. ao término das atribuições, melhorar a solução obtida por refazer pelo menos algumas atribuições procurando os valores de xij tais que os valores de pij respectivos sejam os maiores, enquanto a capacidade bi permitir.

Esta heurística permite soluções viáveis, porém estas não são ótimas ou quase ótimas. Ao se colocar esta heurística no algoritmo do Lagsur, substituindo-se pij por (pij - t.λi Σ wij), tal como seria se fosse a relaxação Lagrangeana ao se relaxar a restrição wij.xij Σ bi, tem-se uma variação da função objetivo a cada iteração e o melhor resultado obtido e então apresentado ao final do algoritmo. A heurística descrita e a relaxação Lagsur foram executadas, com problemas testes obtidos por Beasley (1990) e os resultados estão na Tabela 1.

Na Tabela 1, A5x100, por exemplo, indica que se tem cinco mochilas que receberam 100 itens. A Heurística com Lagsur considera como parte da função objetivo (pij - t.λi.wij) enquanto a Heurística considera apenas pij. O tempo descrito é relativo à relaxação Lagsur. Observa-se na Tabela 1 que os tempos obtidos pela relaxação Lagsur em um Pentium IV, 1 GHz de clock de processador, 128 MB RAM, foram razoáveis, mas os resultados obtidos em termos de valor de solução eram muito melhores com a relaxação Lagsur auxiliando a heurística do que a com heurística sozinha.

Tabela 1. Resultado da heurística com e sem relaxação Lagsur para o PAG.

Problema	Tempo(s)	Heurística com Lagsur	Heuristica sem Lagsur	Número de iterações com Lagsur	Solução Ótima
A5x100	0.310	4457	4503	53	4456
A5x200	0.810	8790	8853	47	8788
A10x100	0.570	4711	4793	59	4700
A10x200	1.530	9418	9521	47	9413
A20x100	1.340	4863	4972	77	4857
A20x200	3.730	9676	9676	77	9666
B5x100	1.100	4504	4672	222	**4008**
B5x200	2.390	8533	8663	176	**8502**
B10x100	1.700	4667	4762	229	4633
B10x200	5.650	9288	9369	306	**9255**
B20x100	1.210	4879	4991	61	4817
B20x200	7.730	9716	9872	225	**9670**
C5x100	0.930	4424	4560	183	4411
C5x200	2.400	8419	8602	173	**8347**
C10x100	2.910	4562	4651	326	**4528**
C10x200	4.300	9283	9349	221	**9247**
C20x100	1.720	4866	4973	65	**4784**
C20x200	9.730	9662	9872	250	**9611**
D5x100	0.350	9157	9341	77	9147
D5x200	1.370	18773	19331	139	18750
D10x100	0.790	10365	10651	108	10349
D10x200	2.620	20596	20890	139	20562
D20x100	1.850	10859	11123	139	10839
D20x200	5.340	21773	22137	170	21733

Problema do Caixeiro Viajante Simétrico (PCV)

Suponha que um caixeiro, a partir de sua cidade de origem, precise visitar um conjunto de n - 1 cidades. Cada cidade deverá ser visitada somente uma vez. Após visitar todas as n - 1 cidades, então o caixeiro deverá voltar a sua cidade inicial. O caixeiro poderá escolher a primeira cidade a ser visitada, a segunda, e assim por diante. Entretanto, caso o caixeiro queira economizar tempo e energia (ou ainda gasolina, recursos, etc.), então ele deve procurar o menor caminho tal, que todas as cidades sejam visitadas somente uma vez e então retornar para a sua cidade de origem. Se o caixeiro sabe a distância entre as cidades, então ele pode encontrar o caminho mais curto, para a sua viagem simplesmente por comparar todas as maneiras possíveis de se visitar as n - 1 cidades e voltar. Entretanto, admitindo-se que exista sempre um caminho ligando uma cidade a outra, a medida em que o número de cidades vai crescendo, o número de maneiras de se visitar todas as cidades cresce também. O problema de se determinar o caminho mais curto, tal que todas as cidades sejam visitadas uma vez é conhecido como *Problema do Caixeiro Viajante (PCV)* e é considerado um problema clássico da Otimização Combinatória (Lawler et al., 1985).

O *(PCV)* é NP-Completo (Lawler et al., 1985). Desta forma, é pouco provável que qualquer algoritmo, por mais eficiente que seja, encontre um caminho ótimo quando o número de cidades for grande e, além disso, a execução seja rápida.

O PCV pode ser descrito da seguinte forma. Seja um grafo não direcionado G=(V,E) com n vértices, onde V={1,2,...,n} e o conjunto de arcos E = {(i,j) tal que i,j $\in$ V e seja c_{ij} o custo não negativo associado ao arco i-j. O Problema do caixeiro viajante pode ser definido como:

$$v(PCV) = min \sum_{i \in V} \sum_{j \in V} c_{ij}x_{ij}$$

$$\text{sujeito a } \sum_{j \in V} x_{ij} = 1 \quad i \in V \quad (5)$$

$$\sum_{i \in V} x_{ij} = 1 \quad j \in V \quad (6)$$

$$\sum_{i \in V} \sum_{j \in V} x_{ij} \leq |S| - 1, \quad \forall S \subset V, \; S \neq \emptyset \quad (7)$$

$$x_{ij} = 0 \text{ ou } 1, \quad i,j \in V, \quad (8)$$

(PCV)

onde $x_{ij} = 1$ se o arco i-j pertence a solução e $x_{ij} = 0$, se o arco i-j não pertence a solução.

A função objetivo do problema indica a minimização das distâncias ao se visitar todas as cidades e voltar a cidade de origem. A restrição (5) indica que, a partir da cidade i, deve-se ir a apenas uma cidade j, j $\in$ V-{i}. Analogamente, a restrição (6) indica que, a partir de uma cidade j, deve-se ir apenas a uma cidade i, i $\in$ V - {j}. A restrição (7) serve para eliminar ciclos dentro de todo o trajeto. Desta forma, se existe um conjunto S com k cidades, k < n, estas k cidades não serão interligadas de forma a existir um ciclo entre elas. O único ciclo que pode existir é o trajeto começando de uma cidade inicial, percorrendo as outras n-1 cidades e voltando à cidade inicial. Esta restrição também pode ser escrita como:

$$\sum_{i \in S} \sum_{j \in V-S} x_{ij} \geq 1, \quad \forall S \subset V, \; S \neq \emptyset \quad (9)$$

A restrição (8) indica que, para xij = 1, o arco que liga i a j é escolhido para fazer parte da solução. Se o valor for 0, então o arco que liga i a j não é escolhido para o trajeto.

O *(PCV)* é *simétrico* quando $c_{ij} = c_{ji}$, para todo i,j $\in$ V. Sendo assim, o *(PCV)* simétrico pode ser definido como um grafo não direcionado completo G= (V,E) com n vértices, V={1,2,...,n}, onde E é o conjunto de todos os arcos conhecidos de i a j ou j a i, e $c_{ij} = c_{ji}$. Desta forma, o *(PCV)* simétrico, doravante chamado de *(PCVS)*, pode ser definido como:

$$v(PCVS) = min \sum_{i \in V} \sum_{j > i} c_{ij}x_{ij}$$

$$\text{sujeito a } \sum_{j<i} x_{ij} + \sum_{j>i} x_{ij} = 2 \quad i \in V \quad (10)$$

(PCVS)

$$\sum_{i \in V} \sum_{\substack{j \in, \\ J>i}} x_{ij} \leq |S| - 1 \quad (11)$$

$$\forall S \subset V, S \neq \emptyset$$

$$X_{ij} = 0 \text{ ou } 1, \quad i,j \in V, j>i \quad (12)$$

A expressão (11) pode ser também escrita como:

$$\sum_{i \in V} \sum_{j \in S} x_{ij} + \sum_{i \in V} \sum_{j \in S} x_{ij} \geq 2, \ \forall S \subset V, S \neq \emptyset \quad (13)$$

$$j>i \qquad j>i$$

Uma heurística para resolver o PCV simétrico (todo caminho de i a j tem o caminho de volta j a i de mesmo tamanho), seria a seguinte, descrita em três passos:

1. verificar os menores caminhos entre dois nós (ou cidades) existentes e ir adicionando à solução de tal forma que cada nó ou cidade tenha ligações com outras duas distintas apenas, e que não exista ciclo no caminho. Com isto, tem-se uma solução inicial;
2. com a solução inicial, verificar se não existe cruzamento de caminhos e retirar estes cruzamentos caso existam. Por exemplo, se um caminho A-B cruza um caminho X-Y, então desfazer estes caminhos e considerar os caminhos A-X e B-Y como nova parte da solução;
3. após tirar todos os cruzamentos, o valor da solução melhora (diminui a distância final). Divide-se então a solução em k caminhos, de tal forma que cada caminho 1-2-3...-S (nós componentes 1,2,3,...,S) componente da solução final tenha sua distância de 1 a S (rearranjo do caminho de 1 a S com os mesmos nós anteriores 2,3,...) melhorada por um método exato (por exemplo, cada parte teria S cidades e se acharia o melhor caminho do início ao fim da parte por enumeração). Após isto, tem-se k caminhos ótimos e unidos entre si.

Para se executar esta heurística, quando esta estiver inserida no contexto da relaxação Lagsur, o que vai variar em função de λ e t será a distância entre os nós. Suponha os nós u e l, que são dois nós pertencentes ao conjunto de nós que se quer fazer o tour, conforme o modo do caixeiro viajante. Seja a distância entre os nós igual a dista[u][l]. Ao se executar a heurística, mantém-se a distância conforme foi calculada entre os pontos u e l. Ao se inserir a heurística no contexto da relaxação Lagsur, a distância vai variar da seguinte forma:

$$dista[u][l] + t.\lambda[u] + t.\lambda[l] .$$

Esta variação seria tal como seria feito quanto à variação da distância na função objetivo da relaxação Lagsur. Mais informações de como se faz a relaxação Lagsur ou Lagrangeana relativo ao PCV está descrito em Lorena & Narciso (2002). A Tabela 2 apresenta os resultados, tomando-se as instâncias testes obtidas em Universität Heidelberg (2007).

A Tabela 2, mostra que os resultados com a Heurística variam de 1,63 a 6,85% em relação à solução ótima. Os resultados obtidos com a ajuda da Heurística variam de 0 a 3,9%, melhorando-se consideravelmente a solução viável. O tempo citado na Tabela se refere a Heurística combinada com a relaxação Lagsur.

Tanto o PAG quanto o PCV simétrico tiveram seus algorimos de resolução feitos em linguagem C e os testes foram feitos em um pentium IV, 1GHz de clock e 128 MB RAM, com sistema operacional Debian.

Tabela 2. Resultado da heurística para o PCV com e sem a relaxação Lagsur.

# Cidades	Tempo(s)	Heurística sem Lags	Heurística + Lagsur	Sol Ótima	Desvio % da Heurística	Desvio % da Heurística + Lagsur
51	6	437	430	430	1,63	0
52	5	7800	7544	7544	3,39	0
100	9	22746	21298	21285	6,86	0,06
101	5	670	642	642	4,36	0
130	10	6591	6115	6110	7,87	0,08
150	11	6729	6565	6532	3,02	0,51
225	33	4011	3905	3859	3,94	1,19
442	131	53445	51789	50783	5,24	1,98
1002	1317	273143	265587	259067	5,43	2,52
2392	6763	403949	392808	378062	6,85	3,9

Considerações Finais

Com os resultados obtidos com os problemas PAG e PCV simétrico, pode-se afirmar que a relaxação Lagsur pode ser usada para melhorar o valor de heurísticas, desde que estas também possam variar, de alguma forma, com os valores dos multiplicadores t (escalar) e λ (vetor) usados na relaxação.

Observou-se uma sensível melhora dos valores das soluções viáveis em um tempo razoável de cálculo. A vantagem é que a heurística utilizada para o cálculo da solução viável pode ser qualquer uma, e esta abordagem pode ser estendida a qualquer tipo de problema da otimização combinatória.

Referências

BEASLEY, J. E. OR-Library: distributing test problems by electronic mail. **Journal of the Operational Research Society**, v. 41, n. 11, p. 1069-1072, 1990.

CATRYSSE, D.; VAN WASSENHOVE, L. N. A survey of algorithms for the generalized assignment problem. **European Journal of Operational Research**, v. 60, n. 3, Aug. p. 260-272, 1992.

GLOVER, F. Surrogate constraints. **Operations Research**, v. 16, n. 4, p. 741-749, July/Aug. 1968.

LAWLER, E. L.; LENSTRA, J. K.; RINNOOY KAN, A. H. G.; SHMOYS, D. B. (Ed.). **The traveling salesman problem**. New York: John Wiley, 1985. 463 p.

LOPES, F. B.; LORENA, L. A. N. A surrogate heuristic for set covering problems. **European Journal of Operational Research**, v. 79, n. 1, p. 138-150, Nov. 1994.

LORENA, L. A. N.; NARCISO, M. G. Relaxation heuristics for a generalized assignment problem. **European Journal of Operational**

LORENA, L. A. N.; NARCISO, M. G. Using logical surrogate information in Lagrangean relaxation: an application to symmetric travelling salesman problems. **European Journal of Operational Research**, v. 138, n. 3, p. 473-483, May, 2002.

NARCISO, M. G.; LORENA, L. A. N. Lagrangean/surrogate relaxation for generalized assignment problem. **European Journal of Operational Research**, v. 114, n. 1, p. 165-177, Apr. 1999.

POLJAK, B. T. Minimization of unsmooth functionals. **USSR Computational Mathematics and Mathematical Physics**, v. 9, n. 3, p. 14-29, 1969.

UNIVERSITÄT HEIDELBERG. **Traveling salesman problem**. Disponível em: <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>. Acesso em: 12 jul. 2007.

Comunicado Técnico, 85

Ministério da Agricultura, Pecuária e Abastecimento

GOVERNO FEDERAL

Embrapa Informática Agropecuária
Área de Comunicação e Negócios (ACN)
Endereço: Caixa Postal 6041 - Barão Geraldo
13083-970 - Campinas, SP
Fone: (19) 3789-5743
Fax: (19) 3289-9594
e-mail: sac@cnptia.embrapa.com.br

1ª edição on-line - 2007

© Todos os direitos reservados.

Comitê de Publicações

Presidente: *Kleber Xavier Sampaio de Souza.*
Membros Efetivos: *Adriana Farah Gonzalez (secretária), Ivanilde Dispato, Marcia Izabel Fugisawa Souza, Martha Delphino Bambini, Sílvia Maria Fonseca Silveira Massruhá, Stanley Robson de Medeiros Oliveira.*

Suplentes: *Goran Neshich, Leandro Henrique Mendonça de Oliveira, Luiz Manoel Silva Cunha, Maria Goretti Gurgel Praxedes.*

Expediente

Supervisor editorial: *Ivanilde Dispato*
Normalização bibliográfica: *Marcia Izabel Fugisawa Souza*
Editoração eletrônica: *Área de Comunicação e Negócios*