

Relaxação Lagrangeana com Divisão em *Clusters* Aplicada ao Problema de Localização de Facilidades Não-capacitado

Francisco de Assis Corrêa
LAC - INPE
São José dos Campos – SP, Brasil
fcorrea@directnet.com.br

Luiz Antonio Nogueira Lorena
LAC - INPE
São José dos Campos – SP, Brasil
lorena@lac.inpe.br

Resumo

Uma estratégia usada para a solução de um problema de grande porte é a sua divisão em problemas menores, resolvendo-os com a ajuda de solvers comerciais. Nesse contexto, este trabalho examina uma técnica que pode ser aplicada a problemas combinatoriais que podem ser representados em grafos de conflitos e considera o particionamento de grafos, de forma a se obter aglomerados (clusters) de vértices e arestas. Ao remover as arestas que ligam os agrupamentos, divide-se o grafo de conflitos em subgrafos, obtendo-se partes com as mesmas características do problema original. Ao relaxar essas arestas no sentido lagrangeano, os subproblemas são resolvidos e obtém-se uma aproximação da solução do problema original. Essa é a idéia da relaxação lagrangeana baseada em clusters (LagClus). Por obter subgrafos com as mesmas características do problema original, tem-se a vantagem de se obter melhores limitantes do que a relaxação lagrangeana tradicional. A LagClus foi testada e apresentou bons resultados para o problema de localização de facilidades não-capacitado (UFLP).

Palavras-chave: Relaxação lagrangeana, particionamento, grafo de conflitos, localização de facilidades.

1. Introdução

A idéia central do estudo de problemas de localização reside na escolha de locais adequados para implantar facilidades (centros) para o atendimento de clientes (pontos de demanda), considerando-se que certos critérios de otimização devem ser atendidos. As facilidades são centros que prestam algum tipo de serviço, tais como fábricas, hospitais, bancos e escolas, que têm como clientes, respectivamente, depósitos, pacientes, correntistas e estudantes.

Muitos desses problemas são combinatoriais por natureza. Por isso, existe o inconveniente de nem sempre ser possível encontrar a solução em um tempo computacional viável, mesmo com a utilização de solvers comerciais, devido à sua classificação como *NP-hard* (Daskin, 1995) e ao porte do problema. Como alternativa buscaram-se métodos heurísticos, metaheurísticas e relaxações. Esta última possui a vantagem de definir limitantes para a solução ótima e pode apresentar uma informação dual de boa qualidade, o que permite avaliar o quão próximo a melhor solução encontrada para o problema está da ótima.

Uma estratégia usada para a solução de um problema de grande porte é a sua divisão em problemas menores, com a mesma característica do problema original, resolvendo-os com a ajuda de solvers comerciais. Essa divisão pode ser conseguida com o particionamento do grafo de conflitos, de forma a se obter aglomerados (*clusters*) de vértices e arestas. Busca-se com esse particionamento separar o grafo de conflitos em subgrafos, ao remover as arestas que ligam os agrupamentos, obtendo-se partes com as mesmas características do problema original.

O inconveniente dessa abordagem é que não há a garantia de, resolvendo-se os subproblemas, obter-se a solução para o problema completo, pois algumas arestas foram dele retiradas. Uma maneira de considerar essas arestas é relaxá-las no sentido lagrangeano e buscar encontrar um limitante de boa qualidade para o problema original. Essa é a idéia da relaxação lagrangeana baseada em *clusters* (LagClus), que incorpora a obtenção de um grafo de conflitos, o particionamento desse grafo em agrupamentos com a mesma característica do problema original e o uso da relaxação lagrangeana para incorporar à resolução do problema as arestas relaxadas, obtidas na separação dos referidos agrupamentos. A Figura 1 mostra as fases desse processo.

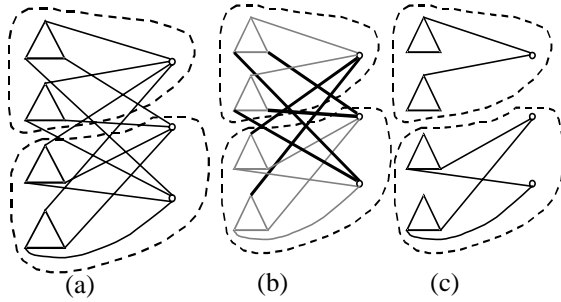


Figura 1 – Aplicação da LagClus. (a) Grafo de conflitos, (b) Arestas de ligação, (c) Separação dos subproblemas (*clusters*).

Um tipo de problema de localização bastante estudado na literatura é o denominado Problema de Localização de Facilidades Não-Capacitado, do inglês, *Uncapacitated Facility Location Problem* (UFLP), que envolve custos fixos para a localização de facilidades e custos de produção, de transporte e distribuição para satisfazer a demanda de determinadas regiões. O objetivo do UFLP é o de decidir onde localizar facilidades e as alocações dos seus clientes, de forma a minimizar o custo total.

Este trabalho trata de descrever e aplicar a LagClus ao UFLP. O restante deste trabalho é organizado como se segue. A Seção 2 descreve o UFLP. A LagClus é discutida na Seção 3. A Seção 4 apresenta os resultados computacionais. As conclusões são apresentadas na Seção 5.

2. UFLP

A modelagem matemática do problema de localização de facilidades não-capacitado (UFLP) é apresentada a seguir (Cornuéjols et al., 1990). Formalmente, as alocações são representadas pelas variáveis binárias x_{ij} , tal que $i \in I$ e $j \in J$, onde I é o conjunto dos pontos de demanda a serem alocados e J é o conjunto de localizações candidatas. $[x_{ij}]$ é a matriz de alocações, com $x_{ij} = 1$, se o ponto de demanda i for alocado à facilidade j , $x_{ij} = 0$, caso contrário. As localizações são representadas pelas variáveis binárias y_j , com $y_j = 1$, se o ponto j for selecionado, $y_j = 0$, caso contrário. O parâmetro c_{ij} define o custo de servir o cliente i pela facilidade j e o parâmetro f_j define o custo fixo de se abrir uma facilidade em j .

UFLP

$$\min \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{j \in J} f_j y_j \quad (1)$$

Sujeito a

$$\sum_{j \in J} x_{ij} = 1 \quad \forall i \in I \quad (2)$$

$$x_{ij} \leq y_j \quad \forall i \in I, \forall j \in J \quad (3)$$

$$x_{ij}, y_j \in \{0,1\} \quad \forall i \in I, \forall j \in J \quad (4)$$

A função objetivo (1) minimiza o custo total do sistema. As restrições (2) impõem que cada ponto de demanda seja alocado a um somente um centro. As restrições (3) definem que somente é possível alocar um ponto de demanda i a um centro j se houver um centro em j . As restrições (4) definem as condições de integralidade.

Vários métodos para a solução do UFLP têm sido elaborados por pesquisadores nas últimas décadas. Embora existam algoritmos exatos para a solução desse problema [Körkel, 1989], a busca por algoritmos alternativos (heurísticas, metaheurísticas e relaxações) é a escolha natural para a solução de instâncias de larga escala, devido à natureza *NP-Hard* do UFLP (Cornuéjols et al., 1990). Uma heurística efetiva e bastante usada é o método lagrangeano (Beasley, 1993) que é baseado na relaxação lagrangeana com o algoritmo de otimização por subgradientes (Daskin, 1995). Várias soluções utilizando metaheurísticas foram aplicadas, tais como *Simulated Annealing* (Alves & Almeida, 1992), Algoritmos Genéticos (Kratka et al., 2001) e Busca Tabu (Sun, 2005).

Recentemente, Resende & Werneck, 2005, adaptaram uma heurística híbrida, que tinha sido desenvolvida para a solução do problema das p -medianas, ao UFLP. Essa heurística é chamada híbrida, porque combina elementos de várias metaheurísticas.

3. Relaxação lagrangeana com divisão em *clusters* (LagClus)

A relaxação lagrangeana com divisão em *clusters* surgiu do trabalho de Ribeiro (2005), que adaptou os resultados de Hicks et al. (2004), no qual esses autores desenvolveram um algoritmo *Branch-and-Price* para o problema de Máximo Conjunto Independente de Vértices com Pesos (PMCIVP). O PMCIVP pertence à classe de problemas que podem ser decompostos em subproblemas do mesmo tipo do original. A abordagem dos autores considera o particionamento do conjunto de vértices do grafo de conflitos para obter subgrafos induzidos de mais fácil solução. O problema original é então reformulado usando a decomposição Dantzig-Wolfe (Bazaraa et al., 1990) e esses subproblemas passam a gerar colunas para a decomposição, aproximando as soluções do problema original. A solução apresentada pelos autores mostra sucesso na abordagem, porém com tempos computacionais altos em alguns casos. Na tentativa de se obter tempos mais adequados, surgiu a idéia de se aplicar relaxação lagrangeana a esse tipo de problema, desenvolvendo-se a LagClus (Ribeiro, 2005).

A idéia da LagClus é a de trabalhar com problemas que podem ser decompostos em subproblemas com a mesma característica do problema original, obtendo-se assim problemas

menores que podem resolvidos de forma exata com algum *solver* comercial, em tempos computacionais aceitáveis. Conforme escrito anteriormente, busca-se particionar o grafo de conflitos, separando-o em subgrafos de mesmas características do original e relaxar as arestas que estão entre os *clusters*. Ribeiro (2005) apresentou bons resultados dessa técnica para problemas de Máximo Conjunto Independente de Vértices (PMCIIV), Rotulação Cartográfica de Pontos (PRCP) e Carregamento de Paletes (PCP).

3.1 O processo de aplicação da LagClus

O processo de aplicação da LagClus pode ser sintetizado em:

- Montar a grafo de conflitos de um problema P_1 .
- Aplicar uma heurística de particionamento para dividir os vértices do grafo G em P *clusters* com aproximadamente a mesma cardinalidade, obtendo-se um P -particionamento. O problema P_1 passa a ser representado por meio da sua função objetivo, sujeita às restrições de adjacências (arestas) divididas em dois conjuntos: um com restrições de adjacências (arestas) intra *clusters* e outro, entre *clusters* ou arestas de ligação.
- Relaxar no sentido lagrangeano as restrições de adjacências entre *clusters*; e
- Decompor a relaxação lagrangeana resultante em P subproblemas e resolvê-la.

3.1.1 Grafo de conflitos

Um grafo de conflitos representa uma relação lógica entre variáveis binárias. Existe um vértice para cada variável binária e seu complemento, e uma aresta entre dois vértices quando no máximo uma das variáveis representadas pelos vértices podem ser iguais a um na solução ótima (Atamtürk et al., 2000).

Para montar o grafo de conflitos para os problemas de localização, deve-se trabalhar com o complemento das variáveis de localização, $\bar{y} = 1 - y$, obtendo-se \overline{UFLP} (Cornuéjols & Thizy, 1982). Os conflitos aparecem nas restrições 6, que formam *cliques*, e com essas transformações de variáveis passam a aparecer também nas restrições 7, que são as restrições de Balinski modificadas, onde apenas a variável de alocação ou a de localização pode ter o valor um.

\overline{UFLP}

$$\min \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} - \sum_{j \in J} f_j \bar{y}_j + \sum_{j \in J} f_j \quad (5)$$

Sujeito a

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

$$x_{ij} + \bar{y}_j \leq 1 \quad \forall i \in I, \forall j \in J \quad (7)$$

$$x_{ij}, \bar{y}_j \in \{0,1\} \quad \forall i \in I, \forall j \in J \quad (8)$$

3.1.2 Particionamento de grafos

Dado um número P e um grafo $G = (V, E)$, onde V um conjunto de vértices com pesos, e E , um conjunto de arestas também com pesos, o problema de particionamento de grafos é o de encontrar P subconjuntos V_1, V_2, \dots, V_p de V , tal que:

- $\bigcup_{i=1}^P V_i = V$ e $V_i \cap V_j = \emptyset$ para todo $i \neq j$;
- $W(i) \approx W/P$, $\forall i = 1, 2, \dots, P$, onde $W(i)$ e W representam, respectivamente, as somas dos pesos dos vértices pertencentes a V_i e V ;
- A soma dos pesos das arestas que conectam os subconjuntos deve ser mínima.

Qualquer conjunto $\{V_i \subseteq V : 1 \leq i \leq P\}$ é chamado de uma partição se satisfaz a condição a), ou seja, cada V_i é uma parte de um P -particionamento. O objetivo consiste então em dividir G em P partições, de forma a minimizar a quantidade de cortes nas arestas e de reduzir o desbalanceamento dos pesos nessas partições.

Referências, programas e *links* para particionamento de grafos podem ser encontrados em: http://www.ace.ual.es/~cgil/grafos/Graph_Partitioning.html.

3.1.3 Relaxação lagrangeana

Considere o seguinte problema de programação inteira, escrito em notação matricial, referido como problema (P), onde x é o vetor de variáveis de decisão, c é o vetor de custos, b e e são vetores de disponibilidades de recursos e A e D são matrizes de coeficientes.

$$(P) \quad v(P) = \min cx \quad (9)$$

Sujeito a

$$Ax \leq b \quad (10)$$

$$Dx \leq e \quad (11)$$

$$x \in \{0, 1\} \quad (12)$$

Define-se a relaxação lagrangeana do problema (P) com respeito ao conjunto de restrições $Ax \leq b$ acrescentando-se um vetor de multiplicadores de Lagrange λ com sinal apropriado a esse conjunto de restrições, incorporando o produto obtido à função objetivo. A relaxação lagrangeana ($L_\lambda P$) do problema (P) é dada por:

$$(L_\lambda P) \quad v(L_\lambda P) = \min_x cx + \lambda(Ax - b) \quad (13)$$

Sujeito a

$$Dx \leq e \quad (14)$$

$$x \in \{0, 1\} \quad (15)$$

As restrições escolhidas $Ax \leq b$ são as restrições que permitem que o problema (P) seja mais facilmente resolvido, caso sejam removidas.

O problema $(L_\lambda P)$ é um problema em x , resolvido para um dado vetor fixo $\lambda \geq 0$, escolhido para garantir que $v(L_\lambda P) \leq v(P)$. Se as restrições forem do tipo $Ax \geq b$, os multiplicadores devem ser não positivos para garantir $v(L_\lambda P) \leq v(P)$. O sinal de λ será irrestrito se $Ax = b$. A qualidade do limite gerado depende do valor de λ e encontrar bons limites por meio dessa relaxação depende de se encontrar um conjunto de multiplicadores λ que resolve o chamado dual lagrangeano (DL), dado por:

$$\max_{\lambda \geq 0} \left\{ \begin{array}{ll} \min & cx + \lambda(Ax - b) \\ \text{sujeito a} & Dx \leq e \\ & x \in \{0,1\} \end{array} \right\} \quad (16)$$

O ideal é encontrar o valor ótimo do dual lagrangeano igual ao valor ótimo do problema original (P). Se esses valores não são iguais, diz-se, então, que existe um *gap* de dualidade, cujo valor é dado pela diferença entre os dois valores ótimos.

A função $v(L_\lambda P)$ é linear por partes, contínua e côncava (Parker & Rardin, 1998). Por ser côncava, o ótimo local é ótimo global, o que faz com que a relaxação lagrangeana seja uma estratégia atraente para se obter limites da solução de (P). Entretanto, por ser linear por partes, não se pode garantir que seja diferenciável no ponto ótimo. Com isso, não se pode usar gradientes, que são substituídos por subgradientes, que aproximam, no sentido lagrangeano as soluções do problema relaxado à solução ótima. (Espejo & Galvão, 2002), (Reeves, 1995), (Ribeiro, 2005).

3.1.4 Aplicação da LagClus

A Figura 2 mostra o grafo de conflitos referente ao \overline{UFLP} com $m = n = 4$.

Depois de definido o grafo de conflitos, deve-se particioná-lo em P subgrafos, de forma que as restrições de adjacências sejam divididas em intra e entre *clusters*. Antes desse particionamento, é feito o colapso das *cliques* para garantir que todos os seus vértices permaneçam no mesmo *cluster*. Supondo-se a divisão em dois *clusters*, com os vértices 1 e 2 no *cluster* 1 e os vértices 3 e 4 no *cluster* 2, obtém-se a Figura 3 (a).

Após o particionamento, deve-se relaxar no sentido lagrangeano as restrições de adjacências entre *clusters*, que são incorporadas à formulação do problema.

As restrições relaxadas permitem dividir o problema em P subproblemas distintos (Figura 3(b)). Aplicando-se a relaxação lagrangeana sobre esse conjunto de restrições, sendo $\lambda \in R_+^K$ e K a

quantidade de arestas relaxadas, e resolvendo os subproblemas, o valor da LagClus será dado por:

$$v(L_\lambda \overline{UFLP}) = \sum_{p=1}^P L_\lambda (\overline{UFLP})_p + \sum_{k=1}^K \lambda_k$$

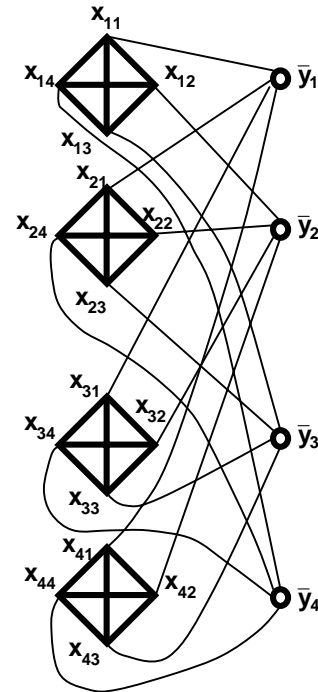


Figura 2 - Grafo de conflitos referente ao problema com $m = n = 4$

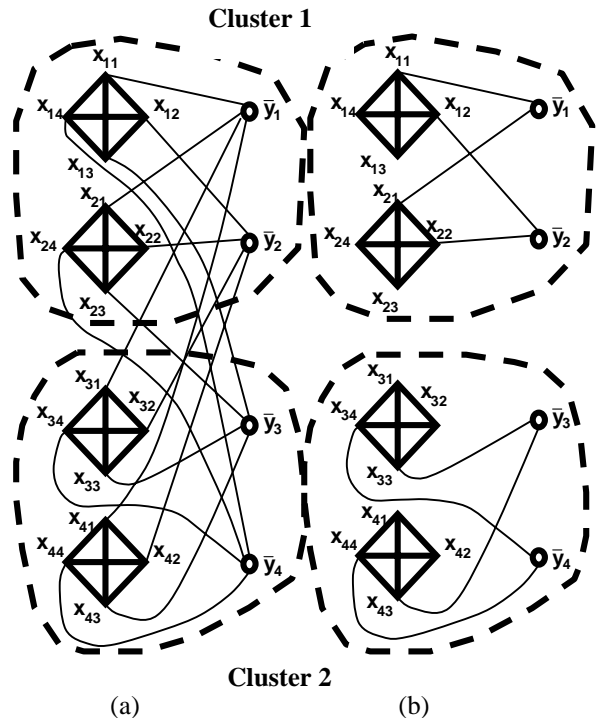


Figura 3 - (a) Particionamento em dois *clusters*, (b) *Clusters* finais

4. Experiência computacional

Esses conceitos foram testados no UFLP em instâncias com grande *gap* de dualidade, obtidas em http://www.math.nsc.ru/AP/benchmarks/UFLP/Engl/uflp_dg_eng.html. Essas são instâncias de difícil solução para métodos baseados em relaxação linear. São apresentadas três classes de instâncias: Gap-A, Gap-B, Gap-C. Os algoritmos de busca local e *Branch-and-Bound* (B&B) gastam considerável esforço para obterem as suas soluções. Todas as instâncias têm os valores $m=n=100$. O custo fixo de abrir uma facilidade é 3000.

A Tabela 1 mostra o desempenho médio do algoritmo *Branch-and-Bound*. Esse algoritmo encontrou a solução ótima em todas as instâncias. A coluna Tempo apresenta o tempo gasto [hh:mm:ss] para executar o algoritmo em um computador PC Pentium 1200 MHz, com 128Mb de RAM. A coluna Iterações B&B mostra a quantidade de iterações executadas com esse algoritmo. Os valores da coluna *Gap* são calculados por $100 * (\text{Solução ótima} - \text{Solução da relaxação de programação linear}) / (\text{Solução ótima})$, e fornece, em porcentagem, o *gap* entre esses valores (Kochetov & Ivanenko, 2003).

Tabela 1 - Desempenho médio do algoritmo *Branch-and-Bound* (Kochetov & Ivanenko, 2003).

Classes	<i>n</i>	<i>Gap</i>	Iterações B&B	Tempo (s)
Gap-A	100	25,6	10 105 775	00:04:52
Gap_B	100	21,2	30 202 621	00:12:24
Gap-C	100	28,4	541 320 830	01:42:51

As instâncias do grupo Gap-A são mais fáceis que as do Gap-B e Gap-C. Cada cliente pode ser atribuído a dez potenciais facilidades, escolhidas aleatoriamente com uma distribuição uniforme entre todas as facilidades.

As instâncias do grupo Gap-B são mais difíceis que as do grupo Gap-A e mais fáceis que as do Gap-C. Cada facilidade pode ser atribuída a dez potenciais clientes, escolhidos aleatoriamente com uma distribuição uniforme entre todos os clientes.

As instâncias do grupo Gap-C são as mais difíceis. Cada facilidade tem dez potenciais clientes e cada cliente tem dez potenciais facilidades escolhidos aleatoriamente com uma distribuição uniforme.

Os valores da solução ótima e do *gap* de dualidade são fornecidos com as respectivas instâncias no *site* acima mencionado e são comparados com os resultados da relaxação lagrangeana com divisão em *clusters* e com os resultados da relaxação lagrangeana tradicional. Esses valores são apresentados nas Tabelas 2, 3 e 4. Com os dados das colunas Solução ótima e *Gap* dualidade obtidos na literatura, foram calculados os valores da coluna Solução PL = (Solução ótima) * (1 - *Gap* dualidade/100), da Tabela 2. A coluna LagClus da Tabela 3 mostra os valores obtidos com a relaxação

lagrangeana com divisão em *clusters*. Os valores de *Gap* LagClus são calculados por: $100 * (\text{Solução ótima} - \text{LagClus}) / (\text{Solução ótima})$, e fornece, em porcentagem, o *gap* entre esses valores. Da mesma forma, na Tabela 4, os valores de *Gap* Lag fornecem o *gap* entre os valores das colunas Lag e Solução ótima. Os valores das colunas Tempo foram obtidos em um computador Pentium IV 3 GHz, com 1Gb de RAM. Para o desenvolvimento da relaxação lagrangeana do UFLP, foram relaxadas as restrições (2), com a otimização do dual lagrangeano obtida por meio do algoritmo de subgradientes. Os subproblemas da LagClus foram resolvidos com o uso do *solver* comercial CPLEX, versão 7.5 (ILOG, 2001).

Os valores em negrito da Tabela 3 mostram que a LagClus obteve melhores valores de *gap* para uma divisão do problema original em dois *clusters*. Isso ocorre porque a quantidade de arestas entre *clusters* é reduzida, melhorando os limitantes obtidos. A perda com essa melhora são os tempos para se obter a solução.

Não foi implementada uma heurística primal, sendo utilizados no método subgradientes os valores da solução ótima como limitantes superiores fixos (*upper bounds*).

O particionamento dos grafos foi feito com programas da família Metis (Karypis & Kumar, 1998) encontrados em <http://www.cs.umn.edu/~metis>, que é uma heurística bastante conhecida.

Tabela 2 – Resultados da aplicação da relaxação de programação linear para o UFLP em instâncias com grande *gap* de dualidade.

Problema	Solução ótima	Relaxação de programação linear		
		Solução PL	<i>Gap</i> PL	Tempo (s)
332GapA	36154	26960,03	25,43	0,25
432GapA	36155	27221,10	24,71	0,24
532GapA	36150	26143,68	27,68	0,28
331GapB	45123	34176,16	24,26	0,25
431GapB	45132	34968,27	22,52	0,23
531GapB	45135	36446,51	19,25	0,23
333GapC	42147	30181,47	28,39	0,25
433GapC	42145	30167,39	28,42	0,25
533GapC	39177	30170,21	22,99	0,25

Tabela 3 Resultados da aplicação da LagClus para o UFLP em instâncias com grande *gap* de dualidade.

Problema	Solução ótima	Relaxação lagrangeana com divisão em <i>clusters</i>		
		Lag Clus	<i>Gap</i> LagClus	Tempo (s)
332GapA_2	36154	27289,76	24,52	953
332GapA_4	36154	26396,21	26,99	85
432GapA_2	36155	27695,29	23,40	1004
432GapA_4	36155	26787,33	25,91	110
532GapA_2	36150	26506,28	26,68	987

532GapA_4	36150	25628,88	29,10	106
331GapB_2	45123	34763,40	22,96	1070
331GapB_4	45123	33942,52	24,78	99
431GapB_2	45132	35229,87	21,94	757
431GapB_4	45132	34721,16	23,07	102
531GapB_2	45135	36802,64	18,46	631
531GapB_4	45135	36219,24	19,75	77
333GapC_2	42147	31182,79	26,01	1833
333GapC_4	42147	29984,59	28,86	181
433GapC_2	42145	31096,01	26,22	1604
433GapC_4	42145	30085,64	28,61	200
533GapC_2	39177	31095,70	20,63	1940
533GapC_4	39177	30061,16	23,27	220

Tabela 4 Resultados da aplicação da relaxação lagrangeana para o UFLP em instâncias com grande *gap* de dualidade

Problema	Solução ótima	Relaxação lagrangeana		
		Lag	Gap Lag	Tempo (s)
332GapA	36154	26635,67	26,33	73
432GapA	36155	26985,32	25,36	61
532GapA	36150	25776,22	28,70	60
331GapB	45123	33933,98	24,80	64
431GapB	45132	34669,69	23,18	61
531GapB	45135	36161,20	19,88	61
333GapC	42147	30171,01	28,41	68
433GapC	42145	30164,78	28,43	79
533GapC	39177	30153,76	23,03	75

5. Conclusões

Os resultados sugerem que a relaxação lagrangeana baseada em *clusters* deve ser mais bem investigada, pois apresenta indícios de ser uma boa alternativa para a solução de problemas de localização. Especificamente para UFLP, Resende e Werneck (2005) apresentam uma heurística para esse problema, aplicando-a a várias instâncias preparadas para *benchmarks*, encontradas na literatura. Esse trabalho é uma boa referência para que, utilizando a LagClus, novos testes e comparações possam ser feitos para esse problema.

6. Referências

- [1] Alves, M. L.; Almeida, M. T. Simulated annealing algorithm for the simple plant location problem: A computational study. *Revista Investigação Operacional*, v. 12, 1992.
- [2] Atamtürk, A.; Nemhauser, G. L.; Savelsbergh, M. W. P. Conflict graphs in solving integer programming problems. *European Journal of Operational Research*, v. 121, p. 40-55, 2000.
- [3] Bazaara, M. S.; Jarvis, J. J.; Serali, H. D. **Linear programming and network flows**. New York, NY: John Wiley & Sons, Inc., 1990.
- [4] Beasley, J. E. Lagrangean heuristics for location problems. *European Journal of Operational Research*, v. 65, p. 383-399, 1993.
- [5] Cornuéjols, G.; Nemhauser, G. L.; Wolsey, L. A. The uncapacitated facility location problem. In: Mirchandani, P. B., Francis R. L. (ed.) **Discrete Location Theory**. New York: Wiley-Interscience, cap. 3, p. 119-171, 1990.
- [6] Cornuéjols, G.; Thizy, J. M. Some facets of the simple plant location polytope. *Mathematical Programming*, v. 23, p. 50-74, 1982.
- [7] Daskin, M. S. **Network and discrete location: models, algorithms and applications**. New York, NY: John Wiley & Sons, Inc., 1995.
- [8] Espejo, . G. A.; Galvão, R. D. O uso das relaxações Lagrangeana e Surrogate em problemas de programação inteira. *Pesquisa Operacional*, v. 22, n. 3, p. 387-402, 2002.
- [9] Hicks, I. V.; Warren, J. S.; Warriar, D.; Wilhenlm, W. E. A Brach-and-Price approach for the maximum weight independent set problem. Texas A & M University: Department of Industrial Engineering, 2004.
- [10] ILOG CPLEX 7.5 Reference Manual 7.5v. 610p. Copyright by ILOG, France, 2001.
- [11] Karypis, G.; Kumar, V. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, v.20, p. 359-392, 1998.
- [12] Kochetov, Y.; Ivanenko, D. Computationally difficult instances for the uncapacitated facility location problem. In: MIC2003: **The Fifth Metaheuristics International Conference**, 2003. Kyoto, Japan. Anais... p. 41-1 – 41-5.
- [13] Körkel, M. On the exact solution of large-scale simple plant location problem. *European Journal of Operational Research*, v. 39, p. 157-173, 1989.
- [14] Kratica, J., Tasic, D., Filipovic, V.; Ljubic, I. Solving the simple plant location problem by genetic algorithm. *RAIRO Operations Research*, v. 39, p. 127-142, 2001.
- [15] Parker, R. G.; Rardin, R. L. **Discrete optimization**. New York, NY: Academic Press, Inc., 1998.
- [16] Reeves, C. R. **Modern heuristic techniques for combinatorial problems**. London: McGraw-Hill Book Company, 1995.
- [17] Resende, M. G. C.; Werneck, R. F. A hybrid multistart heuristic for the uncapacitated facility location problem. *European Journal of Operational Research*, *In Press, Corrected Proof*. Disponível online em 23 mai 2005.
- [18] Ribeiro, G. M. **Relaxação lagrangeana com divisão em clusters para alguns problemas de otimização modelados em grafos de conflitos**. 2005. 99 p. Proposta de Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2005.
- [19] Sun, M. Solving the uncapacitated facility location problem using tabu search. *Computers & Operations Research*, *In Press*. Disponível online em www.sciencedirect.com, 2005.