

ISSN 1677-8464

## Uso de Algoritmos Genéticos em Problemas de Localização Capacitada para Alocação de Recursos no Campo e na Cidade

Luiz Antônio Nogueira Lorena<sup>1</sup>  
Marcelo Gonçalves Narciso<sup>2</sup>

Roteamento e transporte de cargas, passageiros, produção estão sempre presentes no nosso dia-a-dia, sejam na cidade ou no domínio rural. De forma análoga, problemas envolvendo localização de escolas, hospitais, armazéns, supermercados, silos são problemas estratégicos que levam em conta vias de acesso, facilidade do consumidor chegar até o local.

Um sistema de apoio a decisão (SAD) para auxiliar na solução destes problemas é de grande utilidade para a sociedade de modo geral. O ARSIG - Análise de Redes com Sistema de Informação Geográfica (2001) é um sistema de apoio à decisão que integra um SIG com algoritmos para resolver problemas de localização e roteamento. Os resultados das rotas e localizações são mostrados em mapas. Em princípio, o sistema pode ser usado com os SIGs ArcView ou Spring (mais detalhes, ver ARSIG... (2001).

Os problemas de localização podem ser capacitados (PLC) (Senne & Lorena, 1997b), isto é, as restrições de capacidade de cada facilidade (silos, hospitais) são consideradas no problema ou podem ser não capacitados (PLNC) (Senne & Lorena, 1997a), isto é, as

restrições de capacidade de cada facilidade não são consideradas na resolução do problema. Resolver um problema de localização levando-se em conta as restrições de capacidade são mais difíceis de resolver, pois o número de restrições do problema aumenta.

Alguns dos problemas de localização do projeto ARSIG foram resolvidos com o enfoque dos algoritmos genéticos construtivos (Furtado, 1998). Algoritmos genéticos estão em uma classe de algoritmos denominados de evolutivos.

Os algoritmos evolutivos são baseados num processo coletivo de aprendizagem dentro de uma população de indivíduos (estruturas), cada um dos quais representando um ponto no espaço de busca de soluções, para um dado problema. A população é aleatoriamente inicializada e evolui no espaço de busca através dos operadores seleção, recombinação e mutação. Durante o procedimento, informações da qualidade (valor da adaptação) dos pontos de busca são obtidos e são usados para direcionar a busca que favorece a escolha (no processo de seleção) de indivíduos mais adaptados, para que estes gerem novos

<sup>1</sup> Pós-Doutor em Computação Aplicada, LAC/INPE – Instituto Nacional de Pesquisas Espaciais, Av. dos Astronautas, 1758, Jardim da Granja, Caixa Postal 515 – 12201-970 – São José dos Campos, SP. (e-mail: [lorena@lac.inpe.br](mailto:lorena@lac.inpe.br))

<sup>2</sup> Doutor em Computação Aplicada, Pesquisador da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo – 13083-970 – Campinas, SP. (e-mail: [narciso@cnptia.embrapa.br](mailto:narciso@cnptia.embrapa.br))

indivíduos. O mecanismo de recombinação permite misturar informações de uma geração e passá-las aos seus descendentes, e a mutação introduz inovação na população.

A teoria tradicional do algoritmo genético (AGT) assume que ele funciona descobrindo, enfatizando e recombinando bons blocos construtivos nas soluções. Acredita-se que boas soluções são obtidas através da agregação de bons blocos construtivos, idéia que foi formalizada através da introdução da definição de esquema (Holland, 1995).

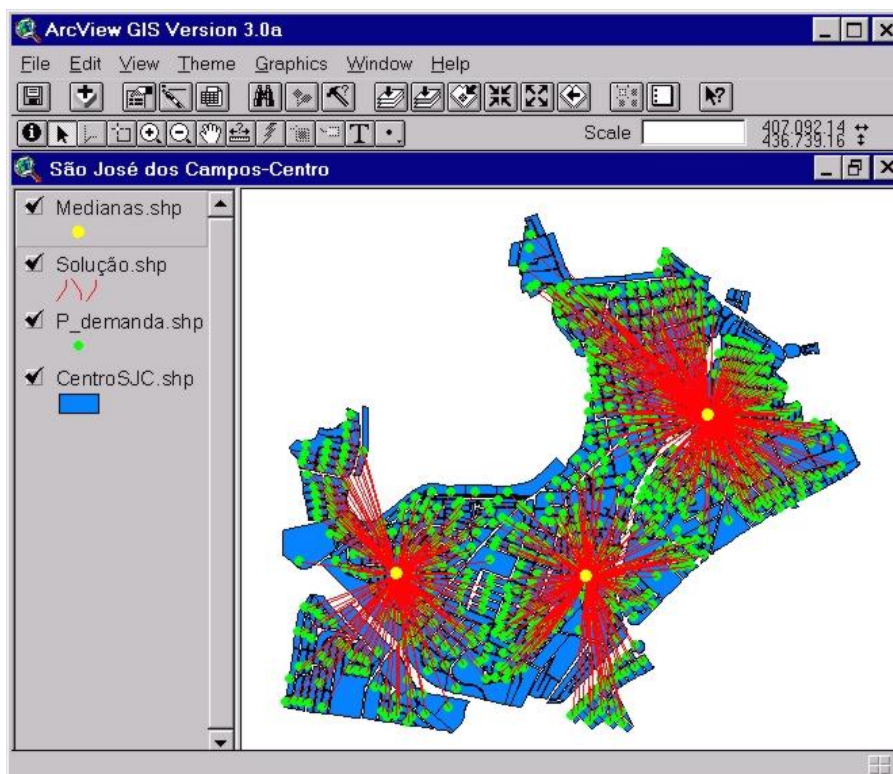
Neste trabalho, o algoritmo genético a ser usado para resolver os problemas de localização capacitada (PLC) é conhecido por "Construtivo". A idéia inicial do Algoritmo Genético Construtivo (AGC) surgiu com o artigo "A Dynamic List Heuristic for 2D-Cutting", desenvolvido por Lorena e Lopes (1996) para resolver um problema de cortes de estoques. A heurística também foi inspirada no algoritmo A\* (Pearl, 1985), conhecido em Inteligência Artificial e usado para direcionar procedimentos de busca. O AGC teve um bom desempenho quanto aos resultados obtidos nas instâncias do PLC. A abordagem de usar o AGC para o PLC é pioneira (inérita) e os resultados obtidos foram muito próximos do ótimo, considerando-se instâncias com solução ótima conhecida. Os detalhes sobre o AGC aplicado ao PLC bem como os resultados são o enfoque deste trabalho.

## Problemas de Localização

O ARSIG tem como um dos objetivos o desenvolvimento de algoritmos eficientes para problemas de localização e roteamento. Estes algoritmos devem ter bom desempenho (no mínimo, tão bom quanto os melhores publicados (Furtado, 1998; Narciso & Lorena, 2001; Senne & Lorena, 1997a; Senne & Lorena, 1997b) para que forneçam soluções de boa qualidade. Neste trabalho, o enfoque é o problema de localização capacitado (com restrição de capacidades). A seguir, tem-se uma descrição do problema.

### Problemas de localização capacitado

O problema de localização de facilidades (recursos) consiste em localizar um conjunto de facilidade dentre  $m$  possíveis para satisfazer, a um custo mínimo (o menor possível), todas as demandas de  $n$  clientes. Os custos envolvidos seriam a construção de facilidades (escola, silos, etc.) e também os custos com o transporte (considera-se proporcional à distância). Leva-se em conta também a capacidade de cada facilidade atender aos clientes. Este problema é de difícil resolução, considerado NP-Hard (Senne & Lorena, 1997a). Foi resolvido usando algoritmos genéticos construtivos, usando a nova mutação, descrita em Senne et al. (2000), e os resultados foram muito próximos do ótimo ou atingiram a solução ótima conhecida da instância. A Fig. 1 ilustra a solução de se localizar 3 escolas em uma dada faixa da região urbana de São José dos Campos, SP.



**Fig. 1.** Solução do processamento com localização de 3 escolas em uma dada região de São José dos Campos.

Em problemas de localização capacitada, a capacidade de cada facilidade é limitada, isto é, cada facilidade po-

derá atender aos clientes conforme a sua capacidade. A formulação matemática para o problema seria

$$v(\text{PLC}) = \min \sum_{i \in V} \sum_{j \in V} c_{ij} \cdot x_{ij} + \sum_{i \in V} F_i$$

(PLC) sujeito a  $\sum_{i \in V} x_{ij} = 1, \forall j \in V = \{1, \dots, n\}$  (1)

$$\sum_{j \in V} d_j \cdot x_{ij} \leq s_i \cdot y_i, \forall i, j \in V = \{1, \dots, n\}$$
 (2)

$$\sum_{i \in V} s_i \cdot y_i \leq \sum_{j \in V} d_j, \forall i, j \in V = \{1, \dots, n\}$$
 (3)

$$x_{ij} \leq y_i, i, j \in V$$
 (4)

$$0 \leq x_{ij} \leq 1 \text{ e } y_i \in \{0, 1\} \forall i \in V, j \in V$$
 (5)

As variáveis citadas significam:

$c_{ij}$  = matriz de custo (é simétrica)

$F_i$  = custo de se abrir uma facilidade (escola, solo, etc.)

$x_{ij}$  = 1 se a demanda do cliente  $j$  for satisfeita pela localidade  $i$ . 0 se não for

$d_j$  = demanda de cada cliente  $j$ .

$y_i$  = 1 se a facilidade  $i$  for aberta. 0 se não for.

$s_i$  = capacidade da  $i$ -ésima localidade.

Neste problema (PLC), observa-se que a restrição (1) significa que cada cliente deve ser atendido por uma ou mais localidades. A restrição (2) significa que a  $i$ -ésima facilidade, com uma capacidade  $s_i$ , tem que atender a um conjunto de demandas  $d_j$  tal que este conjunto não ultrapasse  $s_i$ . A restrição (3) significa que o conjunto total de todas as demandas  $d_j$  a serem atendidas deve ser menor ou igual a todas as facilidades  $s_i$  escolhidas. A restrição (4) significa que, se uma facilidade for aberta,  $x_{ij}$  deve ser no máximo igual a 1. Por fim, a restrição (5) significa que o domínio de  $y_i$  é 0 ou 1 e o de  $x_{ij}$  é  $0 \leq x_{ij} \leq 1$ , isto é, uma facilidade pode atender parcialmente a uma demanda.

Observa-se que o problema de localização capacitado, além de modelar melhor a realidade, é difícil de se resolver (tem muitas restrições, dificultando a resolução). No ARSIG este tipo de problema está sendo tratado e os testes realizados serão vistos no item Resultados deste trabalho.

### Descrição do Algoritmo Genético Construtivo

O AGC inicia com uma população de esquemas (blocos construtivos). Um esquema seria uma solução inicial proposta, podendo ser incompleta. Um exemplo de esquema será descrito mais adiante. Os esquemas carregam informações sobre propriedades estruturais do problema e são avaliados através de funções que determinam o quão promissor é cada um destes esquemas. Os melhores esquemas são incentivados a recombinarem com outros, de tal forma que através de sucessivas gerações novos esquemas ou estruturas são produzidos, os quais além de agregar mais informações sobre o problema, apresentam bom desempenho nas funções de avaliação. Os esquemas ou estruturas que não obtiverem boa avaliação serão eliminados da população através de um critério de poda. No final do processo, espera-se que estruturas de alta qualidade sejam obtidas, pois a agregação sucessiva de informações sobre o problema proporciona a obtenção de soluções (estruturas) de melhor qualidade para o problema de otimização.

No AGC os esquemas são avaliados diretamente através de duas funções de avaliação, e representa uma das diferenças em relação ao AGT, onde os esquemas são avaliados através das instâncias que produzem. A mutação de localização-alocação melhorou os resultados de aplicações anteriores do AGC. Mais

detalhes podem ser vistos em Furtado (1998). Neste trabalho está descrita a aplicação do AGC em problemas de localização capacitado, isto é, as facilidades (hospitais, escolas, etc.) têm limites de atendimento de demanda.

Nos trabalhos de Furtado (1998) e Lorena & Narciso (2001) têm-se uma descrição detalhada do AGC. Nes-

te item, vamos mostrar uma breve descrição dos algoritmos do AGC e do AGC com mutação modificada (AGCM) para resolver o problema de localização capacitada.

Os passos da forma básica de um algoritmo genético construtivo (AGC) podem ser resumidos, como a seguir.

### AGC() {Algoritmo Genético Construtivo}

$\alpha = 0$ ;

$\varepsilon = 0.05$ ; {intervalo de tempo}

**Inicializar**  $P_\alpha$ ; {população inicial}

**Avaliar**  $P_\alpha$ ; {usar as funções de avaliação em cada estrutura}

**Para todo**  $s_k \in P_\alpha$  **calcule**  $\delta(s_k)$  {cálculo do rank}

**Fim\_Para**

**Enquanto não** (condição de parada) **fazer**

**Para todo**  $s_k \in P_\alpha$  satisfazendo  $\alpha < \delta(s_k)$  **fazer** {teste de evolução}

$\alpha := \alpha + \varepsilon$ ;

**Selecionar**  $P_{\alpha+\varepsilon}$  de  $P_\alpha$ ; {operador reprodução}

**Recombinar**  $P_\alpha$ ;

**Avaliar**  $P_\alpha$ ; {calcula adaptação proporcional, usando f e g}

**Fim\_Para**

**Para todo novo**  $s_k \in P_\alpha$  **calcule**  $\delta(s_k)$  {cálculo do rank}

**Fim\_Para**

**Fim\_enquanto**

Onde, para um problema de localização com 10 vértices, valores típicos podem ser:

$s_i = (0, 0, 1, 0, \#, 0, \#, 1, 1, \#)$  ← esquema genérico

1 ← vértice semente,

0 ← vértice alocado a um vértice semente, e

# ← vértice que não participa temporariamente do problema (este símbolo não está presente em estruturas).

Este é um exemplo de representação usado para o problema das p-medias (Furtado, 1998), um outro tipo de problema de localização, onde os pontos de demandas também são considerados como pontos de localização. No caso do problema de localização capacitado tem-se um número m de pontos de localização, distintos dos pontos de demanda, e portanto uma representação adequada seria:

$s_i = (\#, \#, 1, \#, \#, 1, 1, \# \mid 0, 0, \#, 0, \#, 0, \#, 0, 0, \#)$ ,

onde os primeiros símbolos se referem a pontos de localização e os últimos aos pontos de demanda.

O vértice semente e os vértices alocados a este formam um grupo de vértices  $C_k(s_i)$ .

Portanto o processo de avaliação dupla (funções f e g) pode ser definido por:

Seja  $\{\zeta_1, \dots, \zeta_p\}$  o conjunto de índices dos vértices sementes, então

$$g(s_i) = \sum_{k=1}^p \sum_{j \in C_k(s_i)} \mu_{\zeta_k j} \quad \text{e} \quad f(s_i) = \sum_{k=1}^p \lambda_k \cdot [C_k(s_i) - 1],$$

onde  $\lambda_k = \min_{j \in C_k(s_i)} \mu_{\zeta_k j}$  é a menor distância que foi atribuída ao grupo i, e  $|C_k(s_i)|$  é a cardinalidade do conjunto  $C_k(s_i)$ .

O processo evolutivo pára quando algum dos seguintes critérios é satisfeito:

1. a solução ótima do problema é obtida (quando esta é conhecida);
2. a população torna-se vazia, pois todas as estruturas são rejeitadas (obtido através de um parâmetro  $\alpha$  suficientemente grande);
3. um número pré-estabelecido de iterações é atingido.

Um algoritmo intensivo de mutação de trocas de sementes por não-sementes está previsto no algoritmo apresentado. Estuda-se neste trabalho o efeito de

uma nova mutação, que leva em conta os grupos de sementes e vértices alocados às sementes. Este algoritmo de mutação está descrito a seguir.

### AGCM() {Algoritmo Genético Construtivo com Mutação Modificada}

$\alpha=0$ ;

$\epsilon = 0.05$ ; {intervalo de tempo}

**Inicializar**  $P_\alpha$ ; {população inicial}

**Avaliar**  $P_\alpha$ ; {usar as funções de avaliação em cada estrutura}

**Para todo**  $s_k \in P_\alpha$  **calcule**  $\delta(s_k)$  {cálculo do rank}

**Fim\_Para**

**Enquanto não** (condição de parada) **fazer**

**Para todo**  $s_k \in P_\alpha$  satisfazendo  $\alpha < \delta(s_k)$  **fazer** {teste de evolução}

$\alpha := \alpha + \epsilon$ ;

**Selecionar**  $P_{\alpha+\epsilon}$  de  $P_\alpha$ ; {operador reprodução}

**Recombinar**  $P_\alpha$ ;

**Avaliar**  $P_\alpha$ ; {calcula adaptação proporcional, usando f e g}

**Selecionar** um dado  $s_k \in P_\alpha$  e **implementar mutação**

**Fim\_Para**

**Para todo novo**  $s_k \in P_\alpha$  **calcule**  $\delta(s_k)$  {cálculo do rank}

**Fim\_Para**

**Fim\_enquanto**

Portanto, no algoritmo AGCM tem-se a inserção do seguinte comando no algoritmo AGC descrito acima:

**Selecionar** um dado  $s_k \in P_\alpha$  e **implementar mutação**.

Este comando na verdade é o seguinte algoritmo:

sol2 = 10000000 (atribuir valor muito grande para sol2)

Escolher uma estrutura  $s_k$  e ver quais são os grupos formados

**Enquanto não fim repita**

**Enquanto** existir grupo para examinar

Mudar semente de lugar no grupo (todos os casos)

**Se** melhorar a solução associado ao novo grupo

**então** sol1 = solução modificada

**Fim Se**

**Fim enquanto**



**Se**  $sol1 < sol2$  (sementes mudaram e resultado melhorou) **então**

Abre os grupos e calcular nova solução ( $sol2$ ) com o novo  $s_k$

**Se**  $sol2 > sol1$

**então** parar

**senão** ver quais são os grupos formados na nova estrutura  $s_k$

**Fim Se**

**Fim Se**

**Fim\_Enquanto**

No caso do problema de localização capacitado deve observar a capacidade dos grupos na realocação dos pontos de demanda.

## Resultados

O problema de localização capacitado se aproxima mais da realidade pois geralmente as facilidades (escolas, silos, etc.) tem um limite de atendimento ao público. Assim, quando se descreve um problema de localização qualquer usando o enfoque do PLC, tem-se resultados mais adequados à realidade.

Com respeito aos problemas de localização capacitado, tem-se alguns testes com instâncias pequenas e médias (até 130 vértices). Cada instância usada tinha os seguintes dados:

1. número de facilidades (medianas) a serem implementadas (escola, hospital, silos, etc.);
2. número de centros consumidores (ou demandas);
3. distância entre os centros de demanda e as facilidades;
4. custo de se instalar/manter uma facilidade;
5. capacidade máxima de cada facilidade em atender a demandas;
6. solução ótima do problema ou melhor solução conhecida.

Os resultados estão colocados nas Tabelas 1 e 2. Nestas tabelas, tem-se os seguintes campos:

Facilidades = é o número de facilidades que poderão ser utilizados

Vértices = representam os clientes (centros de demanda)

Solução ótima = representa a melhor solução possível do problema

AGC = representa a solução encontrada, a qual pode ser ótima ou não

Diferença = idem aos das Tabelas 1 e 2

Tempo = idem aos das Tabelas 1 e 2

Na coluna Problemas, tem-se o nome das instâncias utilizadas. Por exemplo, CAP41 é o nome de uma das instâncias para o problema de localização capacitada. Esta instância tem 50 vértices (centros de demanda) e 16 facilidades (escola, silo, hospital, etc.). O conjunto de instâncias para este problema pode ser visto no site <<http://www.ms.ic.ac.uk/jeb/pub/>>. Neste site, existem os arquivos do tipo CAPXX.txt, onde XX é um número, os quais foram usados para os testes do algoritmo genético para o PLC.

Cada algoritmo foi codificado na linguagem de programação C e executado numa estação SUN ULTRA 30, com processador de 200 MHz. Pode-se ainda observar nas Tabelas 1 e 2 que o tempo computacional aumenta para um número maior de medianas (facilidades). Isto pode ser facilmente explicado pela busca local (mutação) que é mais demorada nestes casos.

A média da diferença para o AGC em relação à solução ótima foi de 1,74% para a versão sem mutação modificada e 0,97% para a versão com mutação modificada (AGCM). A pequena diferença (em %) da solução obtida pelo AGC em relação à solução ótima mostra que o AGC foi eficiente na otimização dos problemas testados. O mesmo pode ser dito sobre o AGCM, com uma diferença relativa à solução ótima menor e tempo um pouco maior em algumas instâncias, por causa da mutação mais elaborada.

**Tabela 1.** Resultados obtidos com o AGC com instâncias do Projeto ARSIG para problema de localização capacitado.

Problema	Vértice	Facilidade	Solução ótima	AGC	Diferença (%)	Tempo (s)
Cap41	50	16	1040444.3	1060112.4	1,8904	21
Cap51	50	16	1025208.2	1059237.5	3,3193	35
Cap61	50	16	932615.75	951743.73	2,0510	42
Cap71	50	16	977799.40	991120.70	1,3624	67
Cap81	50	25	838499.28	851091.53	1,5018	95
Cap91	50	25	796648.43	803972.14	0,9193	135
Cap101	50	25	796648.43	803091.31	0,8087	131
Cap111	50	50	826124.71	850971.43	3,0076	410
Cap121	50	50	793439.56	803910.32	1,3197	507
Cap131	50	50	793439.56	803121.15	1,2202	593

**Tabela 2.** Resultados obtidos com o AGC com instâncias do Projeto ARSIG para problema de localização capacitado com a mutação mais elaborada.

Problema	Vértice	Facilidade	Solução ótima	AGC	Diferença (%)	Tempo (s)
Cap41	50	16	1040444.3	1047215.3	0,1992	23
Cap51	50	16	1025208.2	1030902.2	0,5176	39
Cap61	50	16	932615.75	949423.23	0,3764	49
Cap71	50	16	977799.40	989749.20	0,8312	72
Cap81	50	25	838499.28	849322.13	1,0946	103
Cap91	50	25	796648.43	802243.13	1,0880	141
Cap101	50	25	796648.43	801857.21	0,9755	167
Cap111	50	50	826124.71	831231.28	0,7574	489
Cap121	50	50	793439.56	801971.31	1,2834	560
Cap131	50	50	793439.56	802751.91	1,2669	650

## Conclusões

Algoritmos genéticos construtivos (AGC) surgem como uma alternativa para resolver problemas de difícil abordagem, tais como os problemas de localização. O algoritmo genético é mais simples de se implementar e pode ser aplicado para vários tipos diferentes de problemas. Os resultados são soluções de muito bom nível (quase ótimas ou ótimas). Esta abordagem é inovadora e se aplica a vários tipos de problema, tais como o problema de Atribuição Generalizada (localização) (Senne et al., 2000), P-medianas (localização) (Furtado, 1998), Cobertura de Conjuntos (localização) (Lorena & Lopes, 1996), etc. Mais detalhes sobre aplicações podem ser vistos em Narciso & Lorena (2001), Senne & Lorena (1997a), Senne et al. (2000) e Senne & Lorena (1997b).

Observou-se que a mutação trouxe uma melhora nos resultados, embora compromettesse um pouco o tempo de execução, o que já era esperado. Porém, este

tempo a mais gasto foi compensado pela melhora da solução. Assim, para problemas em que o tempo de resposta não for importante, mas sim a qualidade da solução melhor será usar o enfoque da mutação mais elaborada.

Graças a esta abordagem, o sistema de apoio à decisão ARSIG pode fornecer resultados de qualidade para o usuário, visto que o PLC modela com razoável precisão a realidade dos centros consumidores em geral e o AGC fornece soluções de boa qualidade a estes problemas.

## Agradecimentos

Os autores agradecem à Fundação de Amparo à Pesquisa do Estado de São Paulo – FAPESP (processo 99/06954-7) pelo suporte financeiro. O segundo autor agradece também ao Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq (processo 300837/89-5).

## Referências Bibliográficas

ARSIG - Análise de Redes com Sistemas de Informações Geográficas. Disponível em: <<http://www.lac.inpe.br/~lorena/ArsigIndex.html>>. Acesso em: 04 abr. 2001.

FURTADO, J. C. **Algoritmo genético construtivo na otimização de problemas combinatoriais de agrupamentos**. 1998. Tese (Doutorado em Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos.

HOLLAND, J. H. **Adaptation in natural and artificial systems**: an introductory analysis with applications to biology, control and artificial intelligence. Cambridge: MIT, 1995. 211 p.

LORENA, L. A. N.; LOPES, F. B. A dynamic list heuristic for 2D-cutting. In: DOLEZAL, J.; FIDLER, J. (Ed.). **System modelling and optimization**. New York: Chapman-Hall, 1996. p. 481-488.

NARCISO, M. G.; LORENA, L. A. N. Algoritmos genéticos construtivos e uma nova proposta de mutação aplicados ao problema de localização capacitado. In:

CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL, 25., 2001, Belo Horizonte. **Resumo das comunicações**. Belo Horizonte: [s.n.], 2001. pt 1, p. 235.

PEARL, J. **Heuristics - intelligent search strategies for computer problem solving**. Massachussets: Addison-Wesley, 1985. 350 p.

SENNE, E. L. F.; LORENA, L. N. Improving traditional subgradient: a lagrangean/surrogate approach to facility location problems. In: EURO INFORMS, 1997, Barcelona. **Proceedings**. [Barcelona: s.n., 1997a].

SENNE, E. L. F.; LORENA, L. N. Lagrangean/surrogate approach to facility for location problems. In: EURO INFORMS, 1997, Barcelona. **Proceedings**. [Barcelona: s.n., 1997b].

SENNE, E. L. F.; LORENA, L. A. N.; FURTADO, J. C.; NARCISO, M. G. **Uma nova proposta de solução para problemas de localização usando algoritmos genéticos e relaxação lagrangeana/surrogate**. Campinas: Embrapa Informática Agropecuária, 2000. 8 p. (Comunicado Técnico da Embrapa Informática Agropecuária, 8).

### Comunicado Técnico, 10

MINISTÉRIO DA AGRICULTURA,  
PECUÁRIA E ABASTECIMENTO



### Embrapa Informática Agropecuária Área de Comunicação e Negócios

Av. Dr. André Tosello s/nº  
Cidade Universitária - "Zeferino Vaz"  
Barão Geraldo - Caixa Postal 6041  
13083-970 - Campinas, SP  
Telefone/Fax: (19) 3789-5743  
E-mail: sac@cnptia.embrapa.br

1ª edição

© Embrapa 2001

### Comitê de Publicações

**Presidente:** Francisco Xavier Hemerly  
**Membros efetivos:** Amarindo Fausto Soares, Ivanilde Dispatto, Marcia Izabel Fugisawa Souza, José Ruy Porto de Carvalho, Suzilei Almeida Carneiro  
**Suplentes:** Fábio Cesar da Silva, João Francisco Gonçalves Antunes, Luciana Alvim Santos Romani, Maria Angélica de Andrade Leite, Moacir Pedroso Júnior

### Expediente

**Supervisor editorial:** Ivanilde Dispatto  
**Normalização bibliográfica:** Marcia Izabel Fugisawa Souza  
**Capa:** Intermídia Publicações Científicas  
**Editoração Eletrônica:** Intermídia Publicações Científicas