SPOLM 2008

ISSN 2175-6295 Rio de Janeiro- Brasil, 05 e 06 de agosto de 2008.

ALGORITMO ILS APLICADO AO PROBLEMA DAS *K*-MEDIANAS CAPACITADO

José André de Moura Brito
IBGE – Instituto Brasileiro de Geografia e Estatística – DPE / COMEQ.
Av.Chile, 500, 10º Andar, Centro – Rio de Janeiro – RJ.
jose.m.brito@ibge.gov.br

Flávio Marcelo Tavares Montenegro
IBGE – Instituto Brasileiro de Geografia e Estatística – DPE / COMEQ.
Av.Chile, 500, 10º Andar, Centro – Rio de Janeiro – RJ.
flavio.montenegro@ibge.gov.br

Luiz Satoru Ochi
UFF – Universidade Federal Fluminense – Instituto de Computação – IC.
Rua Passo da Pátria 156 - Bloco E - 3º andar, São Domingos Niterói – RJ.
satoru@dcc.ic.uff.br

Luciana Roque Brito
Universidade Federal do Rio de Janeiro – COPPE/UFRJ.
Cidade Universitária, Centro de Tecnologia, Bloco H, Sala 319 – Rio Janeiro – RJ.
britom@terra.com.br

RESUMO

Descrevemos um novo algoritmo para um problema clássico de agrupamento conhecido como Problema das k-Medianas. Este problema é similar ao Problema das k-Médias, cujas soluções encontram uso freqüente no processo de estratificação estatística em pesquisas amostrais. Entretanto, substituem-se os centróides do Problema das k-Médias por medóides (medianas), objetivando-se obter agrupamentos mais homogêneos e robustos.

Dados n objetos com p atributos (quantitativos ou qualitativos) e definido um número k de agrupamentos, deve-se selecionar k instâncias de cada um dos $(n-k)$ objetos restantes a sua mediana mais próxima. Restringindo o número máximo de objetos por agrupamento, obtém-se o chamado Problema das k-Medianas Capacitado.

Neste trabalho, propomos um algoritmo baseado em uma heurística de aplicação geral, ou metaheurística, chamada ILS (*Iterated Local Search*) para o Problema das k-Medianas Capacitado. Resultados computacionais com base em dados reais, obtidos dos censos demográfico e agropecuário do IBGE, são apresentados, e mostram a performance razoável do algoritmo em termos de consumo de tempo e qualidade das soluções.

Palavras Chave: Agrupamento; Medianas; Metaheurística; ILS; Capacitado.

Abstract

We describe a new algorithm for a classical clustering problem known as k-Medians Problem. This problem is similar to the k-Means Problem, whose solutions are commonly used for stratification in sample surveys. However, the centroids of the k-Means Problem are replaced by medoids (medians) in the k-Medians Problem, aiming to obtain more robust and homogeneous clusters.

Given n objects with p (quantitative or qualitative) attributes and fixed the number k of desired clusters, we must select k objects, called representatives or medians, in such a way to minimize the sum of distances from each remaining $(n-k)$ objects to their respective nearest median. Constraining the number of objects per cluster to be less or equal to a upper bound gives to the Capacitated k-Medians Problem.

In this paper, we propose an algorithm based on a metaheuristic (a general purpose heuristic) called Iterated Local Search (ILS) for the Capacitated k-Medians Problem. Computational results using real data from IBGE demographic and agricultural censuses are provided, showing the reasonable performance of the algorithm in terms of time consumption and quality of the solutions.
KeyWords: Clustering; Medians; Metaheuristic; ILS;Capacitated.

1. INTRODUÇÃO

Neste trabalho, descrevemos um novo algoritmo para o problema de agrupamento conhecido como Problema das k-Medianas Capacitado. Este problema é similar ao Problema das *k*-Médias (Hartigan e Wong, 1979), cuja solução encontra uso freqüente no processo de estratificação estatística em pesquisas amostrais. Entretanto, neste caso, substituem-se os centróides do Problema das *k*-Médias por medianas. Tal método é mais robusto (resistente a *outliers*) por usar medianas (Kaufman e Rousseeuw**,** 1989) e por minimizar uma soma de distâncias (dissimilaridades) ao invés do desvio quadrático médio (Späth, 1980).

Dado um conjunto de n objetos com p atributos (do tipo quantitativo e/ou qualitativo) e definido um número k de agrupamentos, deve-se selecionar k objetos, chamados de representativos ou medianas, de forma a minimizar a soma das distâncias (função dos atributos) de cada um dos $(n-k)$ objetos restantes a sua mediana mais próxima. Ao restringir-se o número máximo de objetos por agrupamento, obtém-se o chamado Problema das *k*-Medianas Capacitado.

A solução exata (ótimo global) do problema das k-medianas com ou sem a restrição de capacidade pode ser obtida através de uma formulação de programação matemática inteira. Todavia, mesmo para uma quantidade *n* de objetos apenas moderada, a resolução da formulação pode levar, dado o elevado número de variáveis inteiras do tipo 0-1, ao consumo expressivo de tempo computacional, ou até mesmo à não convergência, resultando em uma solução apenas viável (ótimo local).

Como possíveis alternativas para a resolução deste problema, têm sido propostos uma série de algoritmos heurísticos (não exatos), com maior ou menor capacidade de produzir boas soluções viáveis em tempo computacional razoável.

Neste trabalho, propomos, em particular, um algoritmo baseado na metaheurística ILS (*Iterated Local Search*), desenvolvida recentemente e aplicada com sucesso a diversos problemas de otimização. Esta metaheurística consiste, basicamente, de procedimentos de construção da solução inicial e melhoria (busca local), perturbação (diversificação) e aceitação iterativas da solução.

O presente trabalho está dividido em seis seções: Na seção dois, introduz-se em detalhes o problema das k-medianas capacitado. Na seção três, são descritos os tipos de atributos (variáveis) considerados neste problema, os quais são utilizados para o cálculo das distâncias entre as medianas e os objetos de cada grupo. Na seção quatro, apresentamos a formulação clássica de programação inteira aplicada ao problema. Na seção cinco, temos uma descrição detalhada da metaheurística ILS (*Iterated Local Search*) e do algoritmo ILS proposto para a resolução do problema das *k*-medianas capacitado. Na seção seis, apresentamos análises e conclusões referentes às aplicações da formulação e do algoritmo ILS a um conjunto

problemas (instâncias) gerados a partir da base de dados do censo demográfico de 2000 e do censo agropecuário de 2006.

2. PROBLEMA DAS *K*-MEDIANAS CAPACITADO

Dado um conjunto C formado por n objetos ($C = \{o_1, o_2, ..., o_n\}$) com p atributos (quantitativos e/ou qualitativos), deve-se selecionar, a partir de C, k objetos que definem um conjunto M de medianas, de forma a minimizar a soma das distâncias de cada um dos $(n-k)$ objetos restantes a sua mediana $med_i \in M, i \in \{1,...,k\}$, mais próxima. Ou seja, as k medianas $med_i \in M, i = 1,...,k$, definem k grupos, aos quais, por sua vez, serão alocados os $(n-k)$ objetos restantes, procurando-se minimizar a soma das distâncias d_{ij} de todos os objetos $o_j \in med_i, i = 1,...,k$, às suas respectivas medianas:

$$\text{Minimizar } f = \sum_{i=1}^{k} \sum_{\forall o_j \in med_i} d_{ij} \qquad (1)$$

As distâncias da equação (1) representam o grau dissimilaridade entre os objetos e suas respectivas medianas, sendo função de seus atributos (variáveis quantitativas ou qualitativas). Uma possível restrição para o problema das k-medianas é a definição de um número máximo de objetos por grupo igual a T, obtendo-se, desta forma, o chamado problema das k-medianas capacitado.

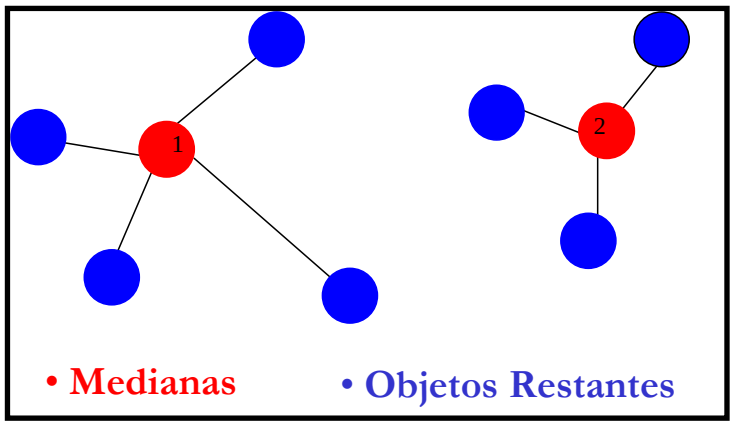

Figura 1 – Exemplo das *k*-medianas capacitado : *n=9*, *k=2* e *T=5*.

O problema das k-medianas e algumas de suas variantes aparecem em muitas situações do mundo real, tais como: localização de indústrias, escolas, redes, facilidades públicas, etc (veja Christofides, 1975). O problema também pode ser interpretado em termos de análise de agrupamentos (Kaufman e Rousseeuw**,** 1989) como: localizar n usuários (objetos) e então substituí-los por k pontos (ou objetos representativos) no espaço n-dimensional (veja Hansen e Jaumard, 1997).

Hakimi (1964) foi primeiro pesquisador a definir o problema e formulá-lo para a localização de uma mediana, generalizando-o, em seguida, para múltiplas medianas. Tal problema é reconhecidamente de alta complexidade computacional (Kariv e Hakimi, 1979 e Garey e Johnson, 1979).

A solução exata (ótimo global) do problema das k-medianas, e de suas variantes, pode ser obtida através de uma formulação de programação inteira binária. Todavia, mesmo para uma quantidade n de objetos apenas moderada, a resolução desta formulação pode levar, dado o elevado número de variáveis 0-1, ao consumo expressivo de tempo computacional (dias, meses, anos, ...) ou até mesmo à não convergência, resultando em uma solução apenas viável (ótimo local).

Além desta formulação e de outros métodos exatos, muitos algoritmos heurísticos têm sido propostos para resolvê-lo, sendo inúmeras referências sobre este problema e suas variantes encontradas, por exemplo, em Mirchandani e Francis (1990).

Mas mesmo com o desenvolvimento de tantos algoritmos heurísticos, muitos destes têm sua performance prejudicada à medida que se aumenta o número de objetos e de medianas. No caso dos algoritmo heurísticos, estes podem convergir para soluções de qualidade muito baixa, isto é, pontos de ótimo local que apresentam valores da função f (equação 1) muito distantes daquele do ótimo global (Campello e Maculan, 1994).

Tendo em vista tais dificuldades, tem-se buscado, no decorrer dos últimos anos, a aplicação de heurísticas de uso mais geral ao presente problema. Estas heurísticas, chamadas de metaheurísticas (Glover e Kochenberger, 2002), têm sido aplicadas com muito êxito em diversos problemas de otimização combinatória, fornecendo, com freqüência, soluções viáveis de qualidade superior àquelas fornecidas pelas heurísticas mais simples.

3. VARIÁVEIS CONSIDERADAS PARA CÁLCULO DAS DISTÂNCIAS

Antes de se proceder à apresentação da formulação e do algoritmo heurístico propostos para a resolução do problema das k-medianas capacitado, faz-se necessária a descrição dos tipos de variáveis que foram consideradas no cálculo das distâncias da equação (1) (vide seção 2). É fato conhecido (Kaufman e Rousseeuw, 1989), que a cada tipo de variável (quantitativa ou qualitativa) corresponde uma particular medida de distância. Em análise de agrupamentos a distância é uma medida matemática que representa o grau de similaridade ou dissimilaridade (Kaufman e Rousseeuw, 1989) entre objetos que serão agrupados.

3.1 VARIÁVEIS QUANTITATIVAS

As variáveis quantitativas são variáveis que apresentam, como possíveis realizações, números resultantes de uma contagem ou uma mensuração.

Para calcular as distâncias entre os objetos com variáveis quantitativas, deve-se, inicialmente, eliminar a dependência na escolha das unidades de medida que serão utilizadas. Ou seja, deve-se efetuar uma padronização dos dados, de tal forma que, dado um conjunto de n objetos representado por $C = \{o_1, o_2, ..., o_n\}$, com cada objeto o_j possuindo observações de p variáveis, $o_j = (o_j^1, o_j^2, ..., o_j^p)$, os valores originais associados às variáveis sejam convertidos para valores adimensionais. O processo de padronização consiste em calcular a média μ e o desvio padrão σ dos valores associados a cada um dos atributos, considerando os n objetos, e, para obter-se o j-ésimo valor associado à h-ésima variável, aplicar a fórmula

$$z_j^h = \frac{o_j^h - \mu_h}{\sigma_h} \qquad (2)$$

onde $j = 1, 2, \ldots, n$ *(objetos)*, $h = 1, 2, \ldots, p$ *(variáveis)* e $z_j = (z_j^1, z_j^2, ..., z_j^h, ..., z_j^p)$ é o vetor de variáveis normalizadas para j-ésimo objeto.

No caso das variáveis quantitativas, uma vez efetuada a padronização, pode-se utilizar a fórmula da distância euclidiana:

Distância Euclidiana - A distância entre dois objetos o_i e o_j é a raiz quadrada do somatório dos quadrados das diferenças entre os valores de i e j para todas as p variáveis:

$$d_{ij} = \sqrt{\sum_{h=1}^{p} (z_i^h - z_j^h)^2}, \qquad (3)$$

3.2 VARIÁVEIS QUALITATIVAS

As variáveis qualitativas representam a informação que identifica alguma qualidade, categoria ou característica, não susceptível de medida, mas de classificação. Podem ser

divididas em binárias, nominais, ordinais e mistas. A seguir, apresentaremos alguns detalhes sobre esses diversos tipos e suas respectivas fórmulas de cálculo de distâncias, tais como usadas neste trabalho.

3.2.1 Variáveis Binárias

As variáveis binárias assumem valor 0 (não tem o atributo) ou 1 (tem o atributo) e podem ser de dois tipos: simétricas ou assimétricas. As variáveis binárias simétricas são aquelas cujos dois estados influenciam igualmente o processo de agrupamento. Já no caso das variáveis binárias assimétricas, os dois estados têm influência diferenciada no processo de agrupamento. Considerando os objetos o_i e o_j , medidos através de p variáveis binárias, constrói-se o seguinte quadro:

Quadro 1

Objetos i e j , Medidos Através de p Variáveis Binárias.

	Objeto j		***Totais***
Objeto i	1	0	
1	a	b	a+b
0	c	d	c+d
Totais	a+c	b+d	

onde
a - Corresponde ao número de variáveis presentes (valor 1) nos dois objetos;
b - O número de variáveis presentes em i e ausentes em j
c - O número de variáveis ausentes em i e presentes em j
d - O número de variáveis ausentes (valor 0) nos dois objetos.

Considerando os dados do quadro 1 (coeficientes a,b,c,d) é possível definir o seguinte tipo de distância para avaliar o grau de dissimilaridade entre variáveis binárias simétricas:

Distância de Emparelhamento - Igual peso às presenças e ausências simultâneas:

$$d_{ij} = \frac{b + c}{a+b+c+d} \tag{4}$$

3.2.2 Nominais

São variáveis cujos estados não se limitam a dois, como nas variáveis binárias, mas podem assumir um determinado número de estados. Neste caso, a distância, ou dissimilaridade, entre os dois objetos o_i e o_j pode ser medida através de:

$$d_{ij} = \frac{p - m}{p} \tag{5}$$

Na equação (5), p é o número total de variáveis nominais e m o número de variáveis nominais de mesmo estado nos dois objetos, ou seja, o número de coincidências entre as variáveis.

3.2.2 Ordinais

Como são variáveis que guardam uma relação de ordenação entre seus possíveis estados, os valores de cada variável do tipo ordinal (que variam entre 1 e M) não são atribuídos aleatoriamente.

Estes tipos de variáveis são muitas vezes empregados para definir a aceitação acerca de algum domínio. No cálculo da dissimilaridade envolvendo variáveis do tipo ordinal, é

importante que os valores das dissimilaridades estejam no intervalo [0,1], como uma forma de padronização dos valores, mas considerando também que será estabelecida previamente uma ordem de precedência para os estados de tais variáveis. Assim, se os objetos têm uma variável do tipo ordinal, então podemos definir M_h estados R_{jh}, com $R_{jh} \in \{1,2,...,M_h\}$, e, para mapearmos o valor de cada variável no intervalo [0,1], substituir o estado R_{jh} do *j-ésimo* objeto na *h-ésima* variável por:

$$x_j^h = \frac{R_{jh} - 1}{M_h - 1} \tag{6}$$

onde R_{jh} é o número (ordem) do estado do h-ésimo atributo do *j*-ésimo objeto e M_h o maior estado da *h*-ésima variável.

Após a aplicação da equação (6), a dissimilaridade pode ser calculada através da equação

$$d_{ij} = \sum_{h=1}^{p} \left| (x_i^h - x_j^h) \right| \tag{7}$$

ou, então, da fórmula de distância euclidiana (equação (3)).

3.3 VARIÁVEIS MISTAS

Comumente, os objetos de um determinado conjunto de dados a serem agrupados possuem variáveis mistas. Neste caso, o cálculo das distâncias entre esses objetos pode ser feito agrupando-se as distâncias associadas aos vários tipos de variáveis em uma única distância d_{ij}, através da equação abaixo:

$$d_{ij} = \frac{\sum_{h=1}^{p} \delta_{ij}^h d_{ij}^h}{\sum_{h=1}^{p} \delta_{ij}^h} \tag{8}$$

onde d_{ij}^h é a distância (dissimilaridade) entre os objetos o_i e o_j considerando o *h*-ésimo tipo de variável. Observamos, ainda, que δ_{ij}^h assume valor 1 se os valores de x_i^h *e* x_j^h estão definidos para h-ésima variável. Em caso contrário, δ_{ij}^h assume valor zero e a distância d_{ij}^h não é calculada.

4. FORMULAÇÃO DO PROBLEMA

A solução do problema das *k*-medianas capacitado é determinada por dois tipos de decisão: (i) a seleção dos k objetos que serão as medianas dos grupos e (ii) a alocação de cada um dos $(n-k)$ objetos restantes a sua mediana mais próxima sem ultrapassar um número máximo de objetos por grupo.

Levando em conta tais decisões, pode-se associar a este problema a seguinte formulação de programação inteira:

	Minimizar $\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij} x_{ij}$	
	$\sum_{i=1}^{n} x_{ij} = 1,\ j = 1,...,n$	(9)
	$x_{ij} \le y_i, i = 1,...,n,\ j = 1,...,n$	(10)
	$\sum_{i=1}^{n} y_i = k$	(11)

	$\sum_{j=1}^{n} x_{ij} \leq T, i = 1,...,n$	(12)
	$y_i, x_{ij} \in \{0,1\}, i, j = 1,...,n$	(13)

Nesta formulação, y_i é uma variável binária que assume valor um se o objeto o_i é definido como mediana, e zero em caso contrário, e x_{ij} é uma variável binária que assume valor um se o objeto o_j é alocado à mediana i (med_i). A restrição (9) garante que cada objeto j será alocado a exatamente uma mediana. A restrição (10) garante que um objeto o_j será associado a um objeto o_i somente se este for uma mediana. A restrição (11) garante que serão escolhidos exatamente k objetos como medianas. A restrição (12) garante que o número de objetos alocados a cada mediana i não ultrapassará um valor T, definido previamente. Finalmente, a função objetivo minimiza a soma das distâncias d_{ij} (dissimilaridades) entre os objetos o_j e as suas medianas med_i $(i = 1,...,k)$.

Analisando x_{ij} e y_i, pode-se observar que o número de variáveis binárias desta formulação é de ordem quadrática, o que possibilita a sua efetiva aplicação somente para problemas com um número não mais que moderado de objetos. Apenas para que se tenha uma idéia de complexidade de resolver tal formulação, se considerarmos o número de objetos $n = 1000$, já teremos por volta de 1.000.000 de variáveis binárias.

5. METODOLOGIA PROPOSTA

Nesta seção, descrevemos um novo algoritmo para o problema das k-medianas capacitado. Tal algoritmo utiliza, basicamente, conceitos da metaheurística ILS (*Iterated Local Search*). Inicialmente, com a finalidade de facilitar o entendimento do novo algoritmo, serão apresentados alguns conceitos básicos associados a heurísticas e a metaheurísticas e, em particular, à metaheurística ILS. Concluindo a seção, apresentamos uma descrição detalhada do algoritmo.

5.1 HEURÍSTICAS E METAHEURÍSTICAS

Uma técnica que encontra boas soluções para um determinado tipo de problema, com custo computacional razoável, sem ser capaz de garantir o ótimo global é chamada de heurística. As heurísticas são ferramentas úteis e atrativas para a resolução de diversos problemas de otimização, especialmente aqueles de alta complexidade computacional. As heurísticas de uso geral, conhecidas como metaheurísticas, podem ser adaptadas para serem utilizadas em qualquer tipo de problema. As metaheurísticas vêm sendo desenvolvidas pelo menos desde os anos 80. Enquanto as heurísticas tradicionais param, em geral, no primeiro ótimo local encontrado, as heurísticas que utilizam os paradigmas das metaheurísticas possuem mecanismos que as possibilitam escapar de tais ótimos, que usualmente são ótimos locais "pobres", isto é, os valores associados a tais ótimos tendem a estar muito distantes do valor do ótimo global. As propriedades que garantem o interesse prático e teórico de uma metaheurística são:

- Simplicidade: Ser baseada em um claro e simples princípio, que deve ser largamente aplicável.
- Coerência: Todos os passos da metaheurística para um problema em particular devem seguir naturalmente os princípios das metaheurísticas.
- Eficiência: Para problemas particulares, podem fornecer soluções ótimas ou próximas do ótimo para todas, ou quase todas, as instâncias consideradas.
- Eficácia: Para problemas particulares, devem levar um tempo computacional razoável para fornecer ótimos globais ou próximas do ótimo.

- Robustez – O desempenho das metaheurísticas deve ser consistente sobre uma variedade de instâncias (problemas de variadas dimensões).

Dentre as inúmeras referências que tratam de metaheurísticas, destacamos as seguintes: Hansen e Ribeiro (2001) e Glover e Kochenberger (2002).

5.2 METAHEURÍSTICA ILS

A metaheuristica ILS (*Iterated Local Search*), proposta por Lourenço, Martin e Stützle (Lourenço et al., 2002), consiste, essencialmente, na aplicação iterativa de um procedimento de busca local em uma solução inicial s_o, que é previamente obtida a partir da utilização de um procedimento aleatório de construção ou considerando uma heurística de construção (Campello e Maculan, 1994). O procedimento de busca local tem por finalidade melhorar a solução inicial e aquelas produzidas após perturbações de soluções ótimas locais.

O êxito desta metaheurística está diretamente associado à escolha do **procedimento de busca local**, ao **procedimento de perturbação** aplicado sobre a solução corrente e ao critério **de aceitação das soluções**. Observamos que a implementação destes procedimentos, bem como do critério de aceitação, está intrinsecamente associada ao problema que será resolvido.

Para descrever a meta-heurística ILS (figura 2), deve-se, primeiramente, fazer as seguintes considerações:

Seja f a função objetivo associada ao problema de otimização combinatória em questão. Rotulamos as soluções viáveis (ótimos locais), ou simplesmente soluções, por s e denotamos por S o conjunto de todas as soluções s. Define-se, também, um subconjunto S^* de S $(S^* \subset S)$, com soluções s^* que sejam também ótimos locais para $\boldsymbol{P}$. Todavia, trabalha-se com S^*, na expectativa de se obter soluções de melhor qualidade que aquelas advindas de S. Ou seja, S^* é um conjunto de soluções mais restrito.

Deseja-se explorar S^*, considerando uma "trajetória" que possibilite a passagem de uma solução atual s^* para uma nova solução, independente desta solução ser uma solução próxima da atual (considerando algum tipo de vizinhança). Para tanto, primeiramente aplica-se uma mudança ou perturbação em s^* (passo 3) , a qual conduz a uma solução intermediária s' pertencente a S. Então, o procedimento de busca local é aplicado em s' e encontra-se uma solução s'' em S^* (passo 4) . Se s'' passa no teste de aceitação (passo 5), ela se torna o próximo elemento na trajetória descrita em S^*, isto é, definimos $s^* = s''$, senão, retorna-se a s^*.

O procedimento de busca local iterativo conduz a uma boa amostragem sobre o conjunto de soluções, ou seja, produz novas soluções, em geral, de boa qualidade. Este fato decorre de as perturbações não poderem ser nem tão pequenas (pouca alteração na estrutura da solução) e nem tão grandes (muita alteração na estrutura da solução). Se elas forem muito pequenas, s' freqüentemente pertencerá à região de atração de s^* e, com isso, poucas novas soluções de S^* serão exploradas. Ao contrário, se as perturbações forem muito grandes, s' pertencerá a uma região aleatória e será necessário reiniciar o algoritmo.

Procedimento Iterated Local Search

1. s^o=Gerar_Solução_Inicial;
2. s^*=Busca_Local(s^o);

Repita

3. s'=Perturbação(s*);
4. s"=Busca_Local(s') ;
5. s*=Critério_Aceitação(s*,s");

Até (Sejam efetuadas *m* iterações);

Fim-Procedimento

Figura 2 - Pseudo-Código da Metaheurística ILS

5.3 ALGORITMO ILS PARA O PROBLEMA DAS *K*-MEDIANAS CAPACITADO

Apresenta-se, a seguir, uma descrição dos quatro procedimentos do algoritmo ILS proposto para a solução do problema descrito neste trabalho.

No procedimento de geração da solução inicial, construímos q soluções, cada uma com k objetos que representam as k-medianas do problema capacitado. Em seguida, dentre estas q soluções, selecionamos a melhor solução de acordo com o valor da função objetivo. Algumas das soluções restantes, as de melhor qualidade segundo o valor da função objetivo, são armazenadas em um conjunto E que é atualizado em cada iteração do algoritmo, substituindo a pior solução de E pela solução obtida mediante a aplicação da busca local.

Conforme observado, ao aplicar-se o procedimento de perturbação em uma solução deve-se buscar o equilíbrio, evitando-se perturbações pequenas ou grandes demais na solução intermediária s^* que será modificada e utilizada na busca local.

Tendo em conta esse equilíbrio, dada uma solução s^* advinda da geração da solução inicial (passo 1) ou após a aplicação do critério de aceitação (passo 5), o procedimento de perturbação que adotamos neste algoritmo consiste em: selecionar, de forma aleatória, uma mediana de s^*, dentre as k medianas, e também um objeto o_j, dentre os $(n-k)$ objetos que não são medianas na solução atual, trocando-o em seguida pela mediana selecionada.

A busca local implementada no algoritmo ILS consiste de um procedimento denominado **T*rocas*** e de um procedimento denominado ***Reconexão por Caminhos***. No primeiro procedimento, aplicado em todas as iterações do algoritmo, selecionam-se duas medianas, med_r e med_s, que compõem a solução obtida após a aplicação do procedimento de perturbação. A partir destas medianas, efetuam-se l trocas de objetos entre os grupos definidos por med_r e med_s, com o intuito de reduzir o valor da função objetivo. Além deste tipo de movimento, tenta-se substituir cada uma das k medianas pelos $(n-k)$ objetos restantes de forma a reduzir o valor da função objetivo.

Neste momento, antes da explanação do segundo procedimento de busca local, é pertinente observar que no presente algoritmo foi utilizado o conceito de ***multstart***. Tal conceito tem sido aplicado em diversos algoritmos que utilizam outras metaheurísticas. No caso do algoritmo ILS implementado neste trabalho, ao invés de efetuar os passos um e dois apenas uma vez e efetuar os passos três, quatro e cinco em um certo número m de iterações, obtendo ao final destas a melhor solução, efetuamos uma chamada principal do algoritmo ILS por w iterações. Assim, em cada uma das w iterações, são executados os passos um e dois (geração e busca local) e, em seguida, os passos três, quatro e cinco (perturbação, busca local e critério de aceitação).

A vantagem do procedimento ***multstart*** sobre o procedimento tradicional é que aquele possibilita a obtenção de soluções de melhor qualidade.

Ao final de cada uma das w iterações, aplica-se o segundo procedimento de busca local (***Reconexão por Caminhos***) considerando a melhor solução s^* obtida deste algoritmo e as soluções $s^e \in E$.

Com base nestas soluções, o procedimento de ***Reconexão por Caminhos*** consistirá em "transformar" cada solução $s^e \in E$ em s^*, através de movimentos (incrementos) em cada uma das medianas que compõem s^e. Em cada movimento aplicado em s^e produz-se uma solução intermediária s^i, até que o conjunto de medianas de s^i fique igual ao de medianas de s^*.

Com aplicação de tal procedimento, tem-se a expectativa gerar soluções intermediárias s^i que tenham custo inferior ao da solução s^*, isto é, $f(s^*) > f(s^i)$.

Em Glover (1996) e Glover e Kochenberger (2002) são fornecidos maiores detalhes sobre este procedimento, que tem sido utilizado, conjuntamente com outras metaheurísticas, para diversos problemas de otimização.

A seguir, apresentamos os passos principais dos procedimentos básicos envolvidos no algoritmo ILS para o problema das *k*-medianas.

Procedimento de Geração (Obter s^0)

- Gerar q soluções iniciais s^0, cada uma com k-medianas, escolhidas aleatoriamente a partir dos n objetos.
- Para cada solução s^0, alocar os $(n-k)$ objetos restantes a sua mediana $med_i \in M, i \in \{1,...,k\}$, mais próxima, considerando a função de distância da equação (1) e não ultrapassando o número máximo (T) de objetos por grupo.
- Escolher, dentre as q soluções, a solução s^0 de menor custo, considerando a função da equação (1).
- Guardar as q^* ($q^* < q$) melhores soluções restantes (com menores valores de acordo com a equação (1)) em um conjunto *E*.

Procedimento de Perturbação (Obter s')

- Considerando a solução s^*, selecionar aleatoriamente, dentre os $(n-k)$ objetos restantes, um objeto o_i e selecionar aleatoriamente uma mediana med_i dentre as k medianas que compõe s^*.
- Substituir a mediana med_i pelo objeto o_j para obter s'.
- Recalcular o valor da função (equação (1)) considerando a nova mediana.

Procedimento de Busca Local (Obter s") – *Trocas*

- Selecionar em s^0 ou em s' duas medianas, med_r e med_s, $(1 \le r \le k, 1 \le s \le k, r \ne s)$.
- Selecionar, aleatoriamente, um objeto $o_a \in med_r$ e um objeto $o_b \in med_s$ e trocá-los entre si. Efetuar tal procedimento de troca l vezes, de forma a reduzir o valor da função dada pela equação (1).
- Realizar tentativas de substituir cada mediana med_i $(1 \le i \le k)$ por outra mediana (considerando como possíveis novas medianas apenas os $(n-k)$ objetos que não são medianas na solução atual) de forma a reduzir o valor da função dada na equação (1).
-

Procedimento de Busca Local (Obter s'') – *Reconexão por Caminhos*

- Selecionar a solução s^* obtida após m iterações do algoritmo ILS.
- Para cada solução $s^e \in E$, aplicar o procedimento de ***Reconexão por Caminhos*** (RC) e obter a melhor solução s^i.
- Para cada med_j de s^* e cada mediana med_i de s^e, efetuar movimentos intermediários em med_i de forma a obter med_j, avaliando, em cada movimento intermediário, se $f(s^i) < f(s^*)$. Na tabela abaixo, apresentamos um exemplo deste procedimento.

Tabela 1 - Ilustração do procedimento (RC)

Solução	*Mediana 1*	*Mediana 2*	*Mediana 3*
s*	**80**	**12**	**24**
s^e	77	14	23
s^i	**78**	14	23
s^i	**79**	14	23
s^i	**80**	**14**	23
s^i	**80**	**13**	23
s^i	**80**	**12**	**23**
s^i =s*	**80**	**12**	**24**

Critério de Aceitação (Obter s^*)

Se $f(s'') < f(s^*)$, então Substitua s^* por s'' e Atualize E e recalcule μ e σ

Senão, Se $f(s'') \in [\mu - 2\sigma, \mu + 2\sigma]$ Substitua s^* por s'', Atualize E e recalcule μ e σ

Senão, Matenha s^*

Obs$_1$: f é o valor da função objetivo (equação 1) considerando a solução s'' ou s^*, μ = média das soluções que estão em E e σ = desvio padrão das soluções que estão em *E*.
Obs$_2$: A atualização da lista *E* consistirá em substituir-se a pior solução desta lista, considerando o valor função f, pela solução s''.

6. RESULTADOS COMPUTACIONAIS

A presente seção contém um conjunto de resultados computacionais obtidos a partir da aplicação da formulação de programação inteira, apresentada na seção quatro, e do algoritmo ILS descrito na seção cinco. A formulação foi implementada usando-se o software LINGO 7.0 e o algoritmo foi implementado em Delphi (versão 7.0). Todos os testes foram efetuados em um computador Pentium IV com 1Gb de memória RAM e dotado de um processador de 1.73 Ghz (*Dual Core*). Antes da apresentação dos resultados, faremos uma breve descrição dos dados utilizados neste trabalho.

6.1 DADOS UTILIZADOS

Com a finalidade de avaliar o novo algoritmo e a formulação, foram utilizados 20 problemas teste obtidos a partir de duas bases de dados do IBGE – Instituto Brasileiro de Geografia e Estatística: (1) domicílios da amostra do Censo Demográfico de 2000 e (2) estabelecimentos do Censo Agropecuário de 2006.

No caso da amostra do Censo Demográfico, foram sorteadas nove áreas de ponderação no estado do Paraná e seis áreas de ponderação no estado de Pernambuco. Em seguida, selecionou-se em cada uma destas áreas um conjunto de registros de domicílios (com tamanhos variando entre 80 e 150 domicílios), definindo, desta forma, uma parte dos problemas utilizados nos experimentos. Os registros de domicílios destas áreas correspondem aos objetos a serem agrupados.

Uma área de ponderação é formada por um agrupamento mutuamente exclusivo de subáreas chamadas setores censitários, os quais englobam, cada um, um conjunto de domicílios. O tamanho das áreas de ponderação, em termos de número de domicílios e de população, não pode ser muito reduzido, sob pena de perda de precisão de suas estimativas. As áreas de ponderação foram definidas considerando essa condição e, também, que são os níveis geográficos mais detalhados da base operacional, concebidos como forma de atender a demandas por informações em níveis geográficos menores que os municípios.

Observamos, ainda, que no arquivo de domicílios foram considerados apenas os registros completos (sem nenhuma observação faltante). Para cada um desses registros, foram

selecionadas sete variáveis (atributos), quais sejam: Situação do Domicílio (ordinal), Número de Pessoas no Domicílio (quantitativa), Número de Banheiros no Domicílio (quantitativa), Domicílio Possui Iluminação (binária), Domicílio Possui Computador (binária), Tipo de Domicílio (nominal) e Total de Rendimentos do Domicílio em Salários Mínimos (quantitativa). Tais variáveis foram definidas previamente e usadas no cálculo das distâncias d_{ij} (vide seção 2) tanto no algoritmo quanto na formulação.

Maiores informações sobre conceitos de área de ponderação, setor censitário e sobre as variáveis consideradas podem ser obtidas consultando: *Metodologia do Censo Demográfico 2000* / IBGE – Rio de Janeiro: IBGE, 2003, Relatórios Metodológicos.

Da base de dados do Censo Agropecuário de 2006, foram selecionados os estados do Paraná, Rio Grande do Sul, Santa Catarina, Bahia e Ceará e, para cada um destes estados, em nível municipal, foram escolhidas as variáveis número de estabelecimentos agropecuários e área dos estabelecimentos agropecuários. De forma similar ao caso do Censo Demografico, tais variáveis foram utilizadas para o cálculo das distâncias d_{ij}.

6.2 RESULTADOS COMPUTACIONAIS

Nas tabela 1 e 2, apresentamos um conjunto de resultados obtidos a partir da aplicação da formulação e do algoritmo ILS às bases de dados descritas acima. Na primeira coluna destas tabelas, temos a identificação dos problemas utilizados nesta simulação.

Na tabela 1, os problemas são identificados por rótulos do tipo **Area_*x*_*y*** (onde x = o número da área de ponderação e y = o número de domicílios selecionados). Na tabela 2, os rótulos são do tipo **Agro_*x*_*y*** (onde x = código da UF e y = número de estabelecimentos agropecuários). Em seguida, nas colunas dois e três destas tabelas, tem-se o número de medianas consideradas e a capacidade de cada grupo, ou seja, o número máximo de objetos associados a cada mediana. No caso do Censo Demográfico, esta capacidade corresponde ao número máximo de domicílios por grupo, enquanto que no Censo Agropecuário a capacidade refere-se ao número máximo de municípios por grupo.

Nas colunas quatro, cinco e seis, temos, respectivamente, o valor da solução obtida a partir da formulação (***Sfo***), o número de variáveis da formulação e o tempo de processamento da formulação em horas. Finalmente, nas colunas sete e oito, são apresentadas as soluções (***SfILS***) obtidas a partir da aplicação do algoritmo ILS e os respectivos tempos de processamento do algoritmo em segundos.

Na execução do algoritmo ILS, o número de iterações principais associadas ao procedimento multstart foi de $w = 25$, e em cada uma destas iterações o algoritmo foi executado $m = 20$ vezes. Além disso, em cada uma das m iterações, o procedimento de geração produziu $q = 30$ soluções iniciais, sendo o conjunto E formado pelas 20 melhores soluções dentre as 30 soluções iniciais.

Para a execução da formulação, considerando cada um dos problemas utilizados, foi estabelecido um tempo limite de 6 horas. Ao final deste tempo obteve-se o ótimo global ou a melhor solução viável (ótimo local) do problema.

A partir dos resultados apresentados nas tabelas 1 e 2, podemos fazer as seguintes observações:

- As soluções produzidas pelo algoritmo ILS, para a maioria dos problemas teste (16 dentre os 20), foram melhores ou iguais àquelas advindas da formulação. Ademais, o algoritmo ILS consumiu um tempo de processamento bem inferior ao da formulação para produzir tais soluções.
- Dentre os vinte problemas considerados, apenas no terceiro e décimo problemas do censo demográfico foi possível obter o ótimo global através da formulação (e também do algoritmo ILS).
- A formulação foi incapaz de convergir mesmo para algumas das menores intâncias consideradas (com cerca de 6000 a 8000 variáveis).
- Analisando os problemas da tabela 2, é possível observar que, mesmo fixando um tempo limite de processamento razoável para a formulação (seis horas), à medida em

que se aumentou o tamanho (número de objetos) dos problemas, as diferenças entre os valores das soluções da formulação e aquelas do algoritmo aumentaram razoavelmente. Ou seja, as soluções viáveis (ótimos locais) produzidas pela formulação são de qualidade baixa, quando comparadas às do algoritmo, especialmente para as maiores instâncias.

Tabela 1 – Resultados da Formulação e do Algoritmo ILS
Dados do Censo Demográfico

Problema	*Medianas*	*Capacidade*	*Sfo*	*Variáveis*	*Tempo*	*SflLS*	*Tempo*
Area_1_80	5	25	17,386	6400	6 horas	17,386	48 seg
Area_2_80	4	55	17,187	6400	6 horas	17,187	33 seg
Area_3_80	3	40	16,248	6400	1 hora *	16,248	20 seg
Area_4_90	3	50	18,288	8100	6 horas	18,309	21 seg
Area_5_90	4	40	18,763	8100	6 horas	18,772	25 seg
Area_6_100	4	40	18,791	10000	6 horas	18,777	41 seg
Area_7_100	5	30	19,039	10000	6 horas	19,045	67 seg
Area_8_100	5	40	19,835	10000	6 horas	19,809	57 seg
Area_9_150	4	60	31,211	22500	6 horas	31,197	75 seg
Area_10_90*	3	60	22,587	8100	2 horas *	22,587	25 seg
Area_11_90	6	25	16,604	8100	6 horas	16,653	66 seg
Area_12_100	5	50	19,624	10000	6 horas	19,594	54 seg
Area_13_100	5	60	19,834	10000	6 horas	19,778	64 seg
Area_14_100	5	40	20,728	10000	6 horas	20,664	72 seg
Area_15_150	6	40	28,852	22500	6 horas	28,118	176 seg

***Ótimo Global**

Tabela 2– Resultados da Formulação e do Algoritmo ILS
Dados do Censo Agropecuário

Problema	*Medianas*	*Capacidade*	*Sfo*	*Variáveis*	*Tempo*	*SflLS*	*Tempo*
Agro_SC_293	3	120	61,562	85849	6 horas	59,428	132 seg
Agro_PR_399	4	150	126,237	159201	6 horas	79,756	390 seg
Agro_RS_496	4	180	238,971	246016	6 horas	116,967	529 seg
Agro_CE_184	3	80	58,427	33856	6 horas	56,247	50 seg
Agro_BA_417	4	140	134,405	173889	6 horas	115,075	415 seg

Com base nas observações acima, concluímos que o algoritmo ILS proposto neste trabalho pode servir como uma boa alternativa para a resolução do problema das *k*-medianas capacitado, principalmente quando houver a necessidade de se trabalhar com problemas de dimensão mais elevada, ou seja, com mais objetos.

Os procedimentos de busca local e perturbação e o critério de aceitação definidos para o algoritmo ILS são apenas tentativas iniciais no sentido de produzir soluções viáveis de boa qualidade para o problema das *k*-medianas, em um tempo de processamento razoável. Em trabalhos futuros, podem ser implementados e incorporados, a este algoritmo ILS, procedimentos e critérios de aceitação mais sofisticados, com o objetivo de obter soluções iguais ou melhores que as soluções obtidas neste trabalho.

7. BIBLIOGRAFIA

[1] Campello, R. E. e Maculan, N. (1994). Algoritmos e Heurísticas. Desenvolvimento e Avaliação de Performance. Editora da Universidade Federal Fluminense.

[2] Christofides, N. (1975). Graph Theory: An Algorithmic Approach. Academic Press, New York.

[3] Garey, M. R. e Johnson, D. S. (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness. San Francisco: Freeman.

[4] Glover, F. (1996). Tabu Search and Adaptive Memory Programming – Advances, Applications and Challenges. In: R. S. Barr, R. V. Helgason and J. L. Kennington (eds.), Interfaces in Computer Science and Operations Research. Kluwer, pp. 1-75.

[5] Glover, F. e Kochenberger, G. A. (2002). Handbook of Metaheuristics, 1ª ed., Norwell: K1uwer Academic Publishers.

[6] Hakimi, S. L. (1964). Optimum distribution of switching centers and absolute centers and the medians of a graph. Operations Research, **12**, pp. 450-459.

[7]Hansen, P. e Jaumard, B. (1997). Cluster Analysis and Mathematical Programming. Math. Programming, **79**, pp. 191-215.

[8]Hansen, P. e Ribeiro, C. (2001). Essays and Surveys in Metaheuristics. Boston: Springer.

[9]Hartigan, J. A. e Wong, M. A. (1979). A k-means clustering algorithm, *Applied Statistics*, **28**, pp. 100-108.

[10] Kariv, O. e Hakimi, S. L. (1979). An Algorithmic approach to network location problems: part 2. The p-medians. *SIAM, Journal on Applied Mathematics*, **37**, pp. 539-560.

[11] Kaufman, L. e Rousseeuw, P. J. (1989). *Finding Groups in Data – An Introduction to Cluster Analysis*. Wiley-Interscience Publication.

[12] Lourenço H.R. , Martin, O. and Stützle T. **(2002).** Iterated local search. In F. Glover and G. Kochenberger, editors, Handbook of Metaheuristics, volume **57** of International Series in Operations Research & Management Science, pages 321-353. Kluwer Academic Publishers, Norwell, MA.

[13]Mirchandani, P. e Francis, R. (1990). Discrete Location Theory. Wiley-Interscience, New York.

[14] Späth, H. (1980). Cluster Analysis Algorithms for Data Reduction and Classification of Objetcs. John Wiley & Sons.