ALGORITMOS VNS E GENÉTICO APLICADOS AO PROBLEMA DE AGRUPAMENTO COM SOMA MÍNIMA DE DISTÂNCIAS

José André de Moura Brito
IBGE – Instituto Brasileiro de Geografia e Estatística - Diretoria de Pesquisas - DPE
Coordenação de Métodos e Qualidade – COMEQ
Av. Chile , número 500, 10º Andar, Centro – Rio de Janeiro – RJ.
email: jose.m.brito@ibge.gov.br

Luciana Roque Brito
UFRJ – Universidade Federal do Rio de Janeiro
email: britom@terra.com.br

RESUMO

O presente trabalho trata da descrição de uma nova metodologia de resolução para o problema de agrupamento com soma mínima de distâncias. Neste problema, dado um conjunto de n objetos com p atributos, deve-se distribuir os n objetos em K grupos, considerando uma medida de distância, e de forma que a soma total das distâncias entre todos os pares objetos, que pertencem a cada um dos grupos, seja mínima. Com a finalidade de resolver tal problema, são propostos dois algoritmos e uma formulação de programação inteira (0-1). Sendo o primeiro algoritmo implementado a partir da metaheurística algoritmos genéticos e o segundo da metaheurística VNS. Com a finalidade de avaliar os algoritmos e a formulação, apresentamos no final deste trabalho um conjunto de resultados computacionais obtidos a partir dos dados do censo demográfico de 2000.

PALAVRAS CHAVE. Agrupamento, Soma Mínima, Distâncias.

ABSTRACT

This work deals of the description of new methodologie for the resolution of the clustering problem with minimal sum of distances. In this problem, given a set of n objects with p attributes, you must distribute these objects into groups, given a measure of distance, and so the sum total of all distances between pairs objects that belong to each group is minimal. In order to solve this problem, two algorithms are proposed and a formulation of integer programming (0-1). The first algorithm implemented from genetic algorithms metaheuristic and the second of VNS metaheuristic. In order to evaluate the algorithms and the formulation, we present the end of this work a set of computational results obtained from the population census data from 2000.

KEYWORDS. Clustering, Minimum Sum, Distances.

1. Introdução

A análise de agrupamento é atualmente uma técnica de grande aplicação prática. Há algumas décadas, biólogos e cientistas sociais já pensavam em formas sistemáticas de definir grupos considerando seus conjuntos de dados. Mas foi com a evolução dos computadores e com um crescente interesse pela aplicação desta técnica, por parte de outros estudiosos, que os métodos de análise de agrupamento começaram a ser aplicados em uma série de problemas associados a muitos domínios, incluindo a Estatística, a Inteligência Artificial, a Economia, etc.

Em particular, no presente trabalho, trataremos do problema de agrupamento com soma mínima de distâncias (Friedman e Meulman, 2004). Dado um conjunto de n objetos com p atributos (variáveis), deve-se distribuir estes n objetos em K grupos, considerando uma medida de distância baseada nos p atributos, de forma que a soma total das distâncias entre os objetos, tomados dois a dois dentro de cada um dos grupos, seja mínima.

Tal problema aparece inserido no âmbito das agências de estatística, mais especificamente, na fase de planejamento de pesquisas que fazem o uso de amostragem probabilística (Bolfarine e Bussab, 2005). Em algumas destas pesquisas, há a necessidade de se dividir a população que será investigada (Bolfarine e Bussab, 2005) em um certo número de grupos (ou estratos). A qualidade destes grupos, considerando uma particular função objetivo, é um dos fatores determinantes para a obtenção de estimativas (de total, de média, etc) mais precisas. Entende-se por estimativas mais precisas, aquelas com menor coeficiente de variação associado. Nestas pesquisas, os grupos são constituídos por municípios, setores censitários, domicílios, etc.

Para resolver tal problema, propomos neste trabalho dois algoritmos: o primeiro utilizando conceitos da metaheurística *Algoritmos Genéticos* (Holland, 1975) e o segundo da metaheurística *VNS* (Mladenovic e Hansen, 1997). Na seção 2 temos uma breve explanação sobre a análise de agrupamento e as medidas de distância que normalmente são utilizadas para formação dos grupos. Finalizando seção, tem-se a descrição do problema de agrupamento com soma mínima de distâncias. Na seção 3 é apresentada a metodologia proposta para a resolução do problema. Inicialmente temos a formulação de programação inteira para este problema, seguida de uma breve descrição de *Algoritmos Genéticos* e do *VNS*. E finalmente, apresentamos os algoritmos propostos para a resolução deste problema. Concluindo o trabalho, temos na seção 4 um conjunto de resultados computacionais e análises utilizando os dados da amostra do censo demográfico de 2000.

2. Análise de Agrupamento

A análise de agrupamento (Kaufman e Rousseeuw, 1989) é uma técnica através da qual os grupos são formados com base em medidas de similaridade ou dissimilaridade, que por sua vez são função dos atributos dos objetos que compõem o conjunto de dados analisado. Em geral, busca-se formar grupos de forma que os objetos em um mesmo grupo sejam similares entre si, considerando alguma medida de distância, enquanto que objetos em grupos distintos sejam bem diferentes.

A classificação de objetos similares dentro dos grupos é uma atividade importante da vida diária, fazendo parte de nosso processo de aprendizagem. Por exemplo: Uma criança aprende a distinguir entre o vermelho e o azul, entre mesas e cadeiras, entre homem e mulher, o que significa uma contínua melhoria nos esquemas de classificação do subconsciente.

Os métodos de análise de agrupamento são aplicados em diferentes domínios, tais como: a Estatística, a Inteligência Artificial, a Economia, o Marketing, etc.

O desenvolvimento destes métodos (Kaufman e Rousseeuw, 1989; Späth, 1980) , também conhecidos como métodos de “clusterização”, constitui uma área de pesquisa que vem sendo bastante explorada, com contribuições recebidas de diversas áreas. Todas as pesquisas sobre os métodos de análise de agrupamento convergem para as limitações computacionais no que se refere à automatização dos processos de agrupamento, à qualidade das soluções encontradas e ao tempo necessário para a obtenção de tais soluções.

2.1 Medidas Utilizadas

Os algoritmos de agrupamento operam tipicamente em duas estruturas de entrada. A primeira representa os objetos por meio de p atributos (variáveis), tais como idade, renda, sexo, etc, através de uma matriz de ordem $n \times p$, onde as linhas correspondem aos objetos e as colunas aos valores de seus respectivos atributos. A segunda estrutura corresponde a uma matriz n x n que contém as distâncias entre todos os objetos. Estas distâncias representam o grau de dissimilaridade ou similaridade entre os objetos.

Distância é uma medida matemática de semelhança, que pode ser geográfica, temporal ou baseada em qualquer característica do objeto (Kaufman e Rousseeuw, 1989). No caso da análise de agrupamento, conforme já destacado, a distância pode ser utilizada para determinar a similaridade ou a dissimilaridade entre objetos que serão agrupados. Definida a distância utilizada, diversos métodos de agrupamento podem ser empregados, considerando sempre o objetivo de garantir que os objetos semelhantes fiquem em um mesmo grupo, e os objetos diferentes fiquem em grupos distintos.

A seguir, apresentamos as medidas de distância que foram utilizadas neste trabalho, considerando variáveis quantitativas, qualitativas ou mistas.

2.1.1 Variáveis Quantitativas

As variáveis quantitativas são variáveis que apresentam como possíveis realizações, números resultantes de uma contagem ou uma mensuração e podem ser divididas em:

a) Intervalares

Em uma variável intervalar as diferenças entre valores sucessivos são sempre iguais.

b) Percentuais ou Absolutas

As variáveis percentuais são as variáveis intervalares para as quais existe um zero absoluto, que representa a origem das medidas. Por exemplo, a variável altura de um indivíduo é percentual. Pode-se dizer que uma altura de 164 cm é o dobro de uma altura de 82 cm. Se houver mudança de medida para metros, continua-se ainda a dizer que a 1ª altura é o dobro da 2ª. No caso das variáveis quantitativas, pode-se utilizar a fórmula da distância euclidiana.

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

$$d_{ij} = \sqrt{\sum_{h=1}^{p} (x_i^h - x_j^h)^2}, \qquad (1)$$

Cabe observar, que antes de se proceder à aplicação da equação (1), é importante avaliar as magnitudes dos valores das variáveis quantitativas associadas aos objetos, tendo em vista que tais variáveis podem traduzir diferentes estruturas de agrupamento. Por exemplo, considerando as variáveis idade , em anos, e renda , em reais, para um conjunto de indivíduos (objetos), verifica-se, que neste caso, a variável renda teria mais importância do que a variável idade, por aquela ter uma magnitude maior.

Desta forma, antes de se construir a matriz de distâncias associadas às variáveis quantitativas, efetua-se, normalmente, uma padronização dos dados. Ou seja, para um conjunto de n objetos, representando por $X = \{ x_1, x_2, \ldots, x_j, \ldots, x_n \}$, com cada objeto x_j tendo p variáveis quantitativas $x_j = (x_j^1, x_j^2, \ldots, x_j^h, \ldots, x_j^p)$, os valores originais associados às variáveis são convertidos para valores adimensionais. O processo consiste em calcular a média μ e o desvio padrão σ dos valores associados a cada uma das variáveis, considerando os n objetos, e aplicar a fórmula apresentada em (2) para obter-se o j-ésimo valor associado à h-ésima variável.

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

sendo $j = 1, 2, \ldots, n$ *(Objetos)* , $h = 1, 2, \ldots, p$ *(Variáveis)* e $z_j = (z_j^1, z_j^2, \ldots, z_j^h, \ldots, z_j^p)$ o vetor de variáveis normalizadas para j-ésimo objeto;

2.1.2 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:

a) 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 no processo de agrupamento. Já as variáveis binárias assimétricas são variáveis cujos dois estados têm influência diferenciada no processo de agrupamento. Considerando os objetos *i* e *j* , medidos através de *p* variáveis binárias, constrói-se o quadro 1:

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 atributos presentes (valor 1) nos dois objetos;
b - O número de atributos presentes em i e ausentes em j
c - O número de atributos ausentes em i e presentes em j
d - O número de atributos ausentes (valor 0) nos dois objetos.

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

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

$$d_{ij} = \frac{b + c}{a+b+c+d} \tag{3}$$

b) Nominais

São aquelas cujos estados não se limitam a dois como nas variáveis binárias, mas podem assumir um determinado número de estados. A distância entre os dois objetos, considerando a dissimilaridade, cujos atributos são nominais pode ser medida através de:

$$d_{ij} = \frac{u - m}{u} \tag{4}$$

Na equação acima, u *é o* número total de variáveis nominais e m o número de variáveis nominais de mesmo estado nos dois objetos, ou seja, m é o número de coincidências entre as variáveis.

c) Ordinais

As variáveis que guardam uma relação de ordenação entre si são ditas ordinais. Desta forma, os valores (estados) 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. O cálculo da dissimilaridade envolvendo variáveis do tipo ordinal é similar ao que se faz no caso de variáveis do tipo intervalar. Entretanto, é importante que os valores de dissimilaridade associados com tais variáveis estejam no intervalo $[0,1]$, uma vez que será estabelecida previamente uma ordem de precedência para tais variáveis, ou seja, se cada um dos i $(i = 1,...,n)$ objetos têm uma variável do tipo ordinal, então tem-se M_h estados R_{ih} , com $R_{ih} \in \{1,2,..., M_h\}$.

Para mapearmos o valor de cada variável no intervalo $[0,1]$, substituímos todos os estados R_{ih} do valor do *i-ésimo* objeto na *h-ésima* variável (atributo) por:

$$x_i^h = \frac{R_{ih} - 1}{M_h - 1} \tag{5}$$

sendo R_{ih} o número (ordem) do estado do *h-ésimo* atributo do *i-ésimo* objeto e M_h o maior estado do *h-ésimo* atributo. E após a aplicação da equação (5), a dissimilaridade pode ser calculada através da equação abaixo:

$$d_{ij} = \sum_{h=1}^{p} \left| (x_i^h - x_j^h) \right| \tag{6}$$

2.1.3 Variáveis com Atributos Mistos

Comumente, os objetos de um determinado conjunto de dados a ser agrupado possuem variáveis de vários tipos. É novamente necessário que se possa calcular as distâncias entre esses objetos. Isto pode ser feito agrupando-se as distâncias associadas aos vários tipos de variáveis em uma única distância. Através da seguinte equação:

$$d_{ij} = \frac{\sum_{h=1}^{p} \delta_{ij}^h d_{ij}^h}{\sum_{h=1}^{p} \delta_{ij}^h} \tag{7}$$

sendo

δ_{ij}^h = 1; se os valores de x_i^h *e* x_j^h estão definidos, considerando o h-ésimo atributo;

δ_{ij}^h = 0; caso contrário;

d_{ij}^h = distâncias entre os objetos i e j considerando o *h*-ésimo atributo.

2.2 Problema Abordado

Conforme já observado, a análise de agrupamento é uma técnica usada para gerar uma estrutura de K grupos que ajusta um conjunto de n observações com p atributos . Os grupos formados caracterizam-se da seguinte forma: os objetos de um mesmo grupo têm um alto grau de associação (similaridade) e os objetos de grupos distintos têm um baixo grau de associação, considerando uma particular função objetivo (que agrega distâncias).

Definindo uma função $c(i)$ que associa cada objeto i a um particular grupo G_k $(1 \le k \le K)$, temos a seguinte relação:

$$c(i) = k \Rightarrow i \in G_k \tag{8}$$

No caso do problema que será abordado, podemos formalizar este objetivo como encontrar a função $c^*(i)$ ótima que minimiza um critério $D(c)$ que mede o grau para o qual o objetivo não é atingido, ou seja:

$$c^* = \arg\min_c \{D(c)\} \tag{9}$$

sendo,

$$D(c) = \sum_{k=1}^{K} \sum_{c(i)=k} \sum_{c(j)=k} d_{ij} \tag{10}$$

E d_{ij} é uma medida de distância que representa o grau de dissimilaridade entre cada par de objetos (i, j), dentro de cada grupo G_k $(1 \le k \le K)$.

A resolução do problema definido a partir das equações (8), (9) e (10), consistirá então, em alocar os n objetos aos K grupos de forma que a soma total das distâncias (dissimilaridades) entre os objetos, tomados dois a dois dentro de cada um dos grupos, seja mínima.

Cabe observar que esta não é uma uma tarefa trivial, tendo em vista a natureza combinatorial deste tipo de problema (entre outros, veja Hansen e Jaumard (1997) e Hubert et al. (2001)).

Caso seja aplicado algum processo de busca exaustiva para garantir a obtenção da solução ótima, será necessário enumerar todas as soluções, ou seja, todas as possibilidades de combinação dos n objetos em k grupos. O número de possibilidades, neste caso, está associado ao número de Stirling de segundo tipo (Johnson e Wichern, 2002), dado por $\frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j}\binom{k}{j}j^n$. Considerando-se, por exemplo, $n=16$ e $K=3$ temos 7.141.686 soluções possíveis. Observando que para um número n maior de objetos, estes valores crescem exponencialmente.

3. Metodologia Proposta

Em virtude da complexidade de obtenção de soluções exatas para o problema de agrupamento, há uma motivação natural pela utilização de métodos heurísticos, tendo em vista, que estes métodos produzem soluções viáveis em um tempo computacionalmente exeqüível, quando comparados aos tempos dos métodos exatos. Em Hartigan (1979), Späth (1980), Kaufman e Rousseeuw (1989) e Gordon (1999) encontra-se uma apresentação detalhada de alguns destes métodos, que diferem pelo tipo de função objetivo utilizada e pela forma de construção dos grupos. Tais métodos possuem, em geral, uma conhecida dificuldade em lidar com mínimos locais, o que pode implicar, por sua vez, na obtenção de soluções viáveis de baixa qualidade, ou seja, que tendem a estar muito distantes da solução ótima.

Levando-se em conta esta última observação, tem se buscado, em particular, a utilização das chamadas metaheurísticas para uma série de problemas de agrupamento. As metaheurísticas são heurísticas de uso geral que podem ser aplicadas em diversos problemas de otimização, fornecendo, em geral, soluções viáveis (mínimos locais) de qualidade superior àquelas advindas das heurísticas.

No caso do problema de agrupamento abordado neste trabalho, apresentamos nesta seção, dois algoritmos: o primeiro utiliza conceitos da metaheurística *Algoritmos Genéticos* (Holland, 1975; Glover e Kochenberger, 2002) e o segundo da metaheurística *VNS* (Mladenovic e Hansen, 1997).

Além destes dois algoritmos, apresentamos a formulação programação de inteira associada ao problema de agrupamento. A partir das soluções produzidas pela formulação, será possível avaliar a performance dos dois algoritmos, ou seja, calcular o *gap* entre as soluções dos algoritmos e a solução da formulação.

3.1 Formulação

$$\text{Minimizar} \quad \sum_{g=1}^{K}\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} d_{ij}x_{ij}^{g}$$

Sujeito a

$$\sum_{g=1}^{k} y_i^g = 1, i = 1,..,n \tag{11}$$

$$\sum_{i=1}^{n} y_i^g \geq 1, g = 1,...,K \tag{12}$$

$$x_{ij}^g \leq y_i^g, \ g = 1,...,K, i = 1,...,n, j = 1,...,n, i \neq j \tag{13}$$

$$x_{ij}^g \leq y_j^g, \ g = 1,...,K, i = 1,...,n, j = 1,...,n, i \neq j \tag{14}$$

$$y_i^g + y_j^g - 1 \leq x_{ij}^g, \ g = 1,...,K, i = 1,...,n, j = 1,...,n, i \neq j \tag{15}$$

A função objetivo desta formulação minimiza a soma das distâncias entre todos os objetos (tomados dois a dois) que estiverem em cada um dos K grupos. A variável binária x_{ij}^g

assume valor 1 quando os objetos i e j estão em um mesmo grupo g e 0 caso contrário e a variável binária y_i^g assume valor 1 quando o objeto i é alocado ao grupo g. A restrição (11) garante que cada objeto i deve ser alocado a exatamente um dos g grupos $(g = 1,...,K)$ e a restrição (12) garante que cada grupo g tem pelo menos 1 objeto. As restrições 13, 14 e 15 asseguram que se quaisquer objetos i e j estiverem em um mesmo grupo, então a variável x_{ij}^g assume valor 1.

Pode-se observar que o número de variáveis binárias desta formulação é da ordem $O(n^2.K) + O(n.K) = O(n^2)$, o que possibilita a sua efetiva aplicação para problemas com um número moderado de objetos.

3.2 Algoritmos Genéticos

Os algoritmos genéticos (AG), criados por Holland em 1975, são uma classe particular de algoritmos evolutivos, que utilizam os operadores genéticos de seleção, cruzamento e mutação. Um AG trabalha com o princípio de que os indivíduos mais capacitados sobrevivem e geram descendentes com suas características hereditárias.

Desta forma, um algoritmo genético parte de uma população inicial de indivíduos (cromossomos), faz a avaliação de cada um, seleciona os melhores e aplica os operadores genéticos, com a finalidade de criar uma nova população. Este processo pode ser usado para resolver qualquer problema de otimização.

Como primeiro passo do algoritmo, deve-se determinar como os cromossomos serão representados, o que comumente é feito através de *strings* ou vetores, sendo estes preenchidos com valores binários, inteiros ou reais. A população inicial pode ser gerada aleatoriamente ou utilizando alguma heurística de construção (Glover e Kochenberger, 2002), produzindo *m* cromossomos $(C_1, C_2, ..., C_m)$, onde *m* é o tamanho da população, que representa um pequeno subconjunto do espaço total de soluções viáveis do problema.

Para cada cromossomo da população deve-se obter $f_i = f(C_i), \forall i = 1,...,m$, que é a avaliação da função objetivo. E finalmente, operadores genéticos devem ser aplicados à população na seguinte ordem: (1) Seleção: Baseado nos valores de f_i, os melhores cromossomos são selecionados e duplicados em substituição aos piores. (2) Cruzamento: Este operador possibilita a diversificação no espaço das soluções. O operador de cruzamento escolhe aleatoriamente dois cromossomos e troca partes do seu padrão genético. (3) Mutação: Operador que permite a regeneração de algum gen que possa ter sido eliminado da população, de forma inesperada, em alguma etapa do processo.

Após a aplicação destes operadores, obtém-se uma nova população, e sobre esta, deve-se repetir a avaliação da função objetivo para cada cromossomo e a aplicação dos operadores genéticos e assim sucessivamente, em um processo iterativo de geração de novas populações. Normalmente, são considerados os seguintes critérios de parada neste algoritmo:

- Número máximo de gerações - Repetir os passos (1, 2, 3) por Q vezes.
- Tempo máximo de processamento.
- Número máximo de iterações em que não há melhoria no valor da função objetivo.

A solução obtida ao final da execução do algoritmo corresponde a última população gerada, da qual o cromossomo que tem associado o menor (maior) valor da função objetivo no problema de minimização (maximização) é usualmente escolhido.

3.3 Algoritmo Genético Proposto

Seguindo as observações acima, para implementação do algoritmo genético considerando problema de agrupamento proposto, optou-se por representar os cromossomos através de um conjunto de m vetores numéricos, sendo cada um destes vetores compostos por n posições que correspondem, por sua vez, ao número de objetos do problema. Cada uma das n posições destes vetores conterá um valor inteiro entre 1 e K, sendo K o número de

agrupamentos escolhido previamente. Definindo-se por exemplo $n=8$ e $K=2$, podemos ter o seguinte cromossomo:

1	1	2	1	2	2	2	1

Observamos que na primeira geração deste algoritmo os m vetores (cromossomos) são preenchidos aleatoriamente, ou seja, são sorteados valores entre 1 e K para cada uma nas n posições dos m cromossomos.

Uma vez gerada a população, no passo seguinte será efetuada a avaliação da função objetivo. Ou seja, para cada cromossomo C_i da população, deve-se calcular o valor de $f_i = f(C_i), \forall i = 1,...,m$ considerando a equação 10 e aplicar os operadores descritos abaixo.

(i) Seleção – Para realizar esta operação foi utilizado o método do torneio. Em tal método um conjunto de τ $(\tau < m)$ cromossomos é selecionado da população atual e comparado, considerando a função objetivo, e o melhor dentre estes é selecionado. Tal método é aplicado m vezes de forma a obter a nova população.

(ii) Cruzamento – Foi aplicado o cruzamento uniforme, que é capaz de recombinar quaisquer posições entre dois cromossomos. Considerando que o cromossomo tem n posições, utilizou-se uma "máscara" (vetor auxiliar) com n posições com valores preenchidos com 0's e 1's estocasticamente, usando a distribuição de *Bernoulli* (Bussab e Morretin, 2003). Em seguida, são selecionados dois cromossomos pais da população atual, e partir destes, produz-se dois novos cromossomos filhos, considerando tal máscara. As posições com 0 permanecerão inalteradas nos cromossomos filhos e as posições com 1 serão alteradas. Para ilustrar o procedimento, no quadro 2 temos um exemplo com $n=7$ e $K=3$

Quadro 2 – Ilustração do Cruzamento Uniforme

1	0	1	1	0	1	0	Máscara
1	**2**	**2**	**3**	**1**	**3**	**2**	**Pai-1**
2	**2**	**3**	**3**	**2**	**1**	**1**	**Pai-2**
2	**2**	**3**	**3**	**1**	**1**	**2**	**Filho-1**
1	**2**	**2**	**3**	**2**	**3**	**1**	**Filho-2**

(iii) Mutação – Correspondeu a uma pequena perturbação, realizada em alguns cromossomos C_i, com a perspectiva de regenerar valores entre 1 e K que porventura tivessem sido eliminados na reprodução ou no cruzamento. Tal operação foi efetuada em um pequeno percentual dos elementos (gens) associados a um cromossomo C_i, sorteando-se de forma aleatória uma de suas n posições e para esta posição substitui-se o valor atual por um novo valor entre 1 e K.

1	3	2	1	**2**	2	2	3

1	3	2	1	**1**	2	2	3

Além de utilizar os operadores de mutação e cruzamento, a partir do trabalho de Buriol et al. (2005), desenvolvemos um procedimento baseado em *path-relinking* , que foi aplicado a cada T gerações do algoritmo. O *path-relinking* é uma estratégia de intensificação originalmente proposta por Glover (Glover e Kochenberger, 2002), para explorar trajetórias entre soluções elite obtidas a partir da *busca tabu* ou *scatter search*. A idéia principal por trás do *path-relinking* é: dado um conjunto de soluções de melhor qualidade, denominadas soluções elite, deve-se construir caminhos entre duas delas de forma que seja possível avaliar as soluções intermediárias pertencentes a tais caminhos. Com isto, o *path-relinking* procura funcionar como uma fase de intensificação sobre as soluções obtidas na fase de busca local, ou seja, procura melhorar a qualidade das soluções.

No caso do presente algoritmo, a cada T gerações a população atual é particionada em três conjuntos (A, B e C); Sendo o conjunto A formado pelos 30% melhores cromossomos (soluções considerando a função objetivo), enquanto o conjunto C é formado pelos 20% piores cromossomos e o conjunto B é constituído pelos cromossomos de qualidade média (50%). Em seguida, são considerados três tipos de atualização nas soluções, ou seja, cromossomos de cada

um dos grupos: (1) Todos os cromossomos de A são promovidos para próxima geração (2) Todos os cromossomos de C são substituídos por novas soluções iniciais, geradas aleatoriamente, como na primeira geração e (3) Antes de efetuar as atualizações (1) e (2), cada cromossomo de B é substituído por um cromossomo obtido a partir da "combinação" de um cromossomo de A (solução base) com um cromossomo de C (solução guia). Ou seja, são efetuados movimentos na solução guia (C) de forma a transformá-la na solução base (A). As soluções obtidas no decorrer deste processo são chamadas de soluções intermediárias. A melhor solução intermediária (cromossomo avaliado) obtida substituirá o pior cromossomo do conjunto B. No quadro 3 a seguir é ilustrado o processo de obtenção das soluções intermediárias:

Quadro 3 – Soluções do Path-Relinking

Solução Base (A)	**2**	**3**	**2**	**3**	**1**	**1**
Solução Guia (C)	**1**	**2**	**2**	**3**	**1**	**2**
Intermediária (1)	**2**	**2**	**2**	**3**	**1**	**2**
Intermediária (2)	**2**	**3**	**2**	**3**	**1**	**2**
Intermediária (3)	**2**	**3**	**2**	**3**	**1**	**2**
Intermediária (4)	**2**	**3**	**2**	**3**	**1**	**2**
Intermediária (5)	**2**	**3**	**2**	**3**	**1**	**2**
Intermediária (6)	**2**	**3**	**2**	**3**	**1**	**1**

3.4 VNS (Variable Neighborhood Search)

O VNS, proposto por Mladenovic e Hansen (1997), é uma metaheurística que busca uma exploração eficiente do espaço de soluções viáveis através de uma troca sistemática de estruturas de vizinhança. O VNS se diferencia de outras meta-heurísticas baseadas em métodos de busca local por não seguir uma trajetória, mas sim explorar vizinhanças gradativamente mais distantes da solução atual, focando a busca em uma região em torno de uma nova solução somente se um movimento de melhora é realizado.

O algoritmo de busca local implementado nesta metaheurística deve realizar uma seqüência de modificações em uma vizinhança de uma solução, procurando melhorar o valor da função objetivo até que um ótimo local seja encontrado. Contudo, em muitos casos, o ótimo local está distante do ótimo global, sendo necessário analisar o valor da função objetivo em outras vizinhanças a fim de encontrar soluções melhores.

No algoritmo VNS básico (Glover e Kochenberger, 2002), seleciona-se aleatoriamente uma solução $x_0' \in V_s(x_0)$ e a submete a uma busca local. Se a busca local fracassar na obtenção de uma solução de valor melhor do que a solução atual, incrementa-se a ordem da vizinhança corrente e, dessa forma, passamos a explorar a solução em outra vizinhança V_{s+1}. Em caso contrário, se a busca local encontrar uma solução de valor melhor do que a solução corrente na vizinhança V_s, atualizamos a solução e a ordem da vizinhança volta a ser igual a 1 (vizinhança V_1). Este processo se repete até que algum critério de parada seja satisfeito.

3.5 Algoritmo VNS Proposto

O algoritmo VNS proposto neste trabalho, funciona, essencialmente, da seguinte forma: Em cada uma de suas w iterações, o algoritmo constrói uma solução inicial x_0 de forma aleatória (como na 1ª geração do algoritmo genético), onde x_0 é um vetor com n posições, sendo cada posição de x_0 preenchida com um valor entre 1 e K (número de grupos).

A exploração das vizinhanças é efetuada considerando todas as possíveis combinações dos K grupos tomados dois a dois e a seleção aleatória de s objetos em cada grupo para efetuar a troca. Na vizinhança 1 de x_0 ($V_1(x_0)$) seleciona-se um objeto de cada grupo (G_i e G_j), na vizinhança dois de x_0 ($V_2(x_0)$) selecionam-se dois objetos em cada grupo. E de forma análoga, na vizinhança s de x_0 ($V_s(x_0)$), são selecionados s objetos em cada grupo. Uma vez

definidos os grupos e selecionados os objetos, são aplicados dois tipos de trocas (movimentos). A seguir apresentamos na figura 1 o pseudo-código simplificado do algoritmo VNS.

(1) **Faça** $x_{best} \leftarrow +\infty$
(2) **Para** w←1 Até *Maximo_Iteracoes* **Faça**
(3) s←1
(4) Gerar uma solução x_0
(5) **Enquanto** (s < *Maximo_Vizinhancas*) **Faça**
(6) **Para** cada grupo G_i e grupo G_j $(i = 1,...K-1,\ j = i+1,..,K)$ **Faça** (em x_0)
(7) **Para** q←1 Até *Maximo_Trocas* **Faça** (na solução x_0)
(8) Selecionar s objetos o_r de G_i e s objetos o_t de G_j
(9) Substituir cada objeto o_t em G_j por o_r e cada objeto o_r em G_i por o_s
(10) Substituir cada objeto o_t em G_j por o_r ou substituir o_r em G_i por o_t
(11) **Se** $f(x_0^{'}) < f(x_0)$ **Então** $x_0 = x_0^{'}$ Senão $x_0^{'} = x_0$
(12) **Se** $f(x_0) < f(x_{best})$
Então Faça $s \leftarrow 1$, $x_{best} = x_0$ **Senão** Faça $x_0 = x_{best}$ $s \leftarrow s+1$

As trocas definidas em (8) e (9) diferem quanto à cardinalidade dos grupos, ou seja, em (8) o número de objetos em cada grupo permanece inalterado e em (9) há alteração do número de objetos em cada grupo. Os passos (8)-(11) são repetidos por q vezes, o que corresponde a um número máximo de trocas. E para cada troca efetuada de (8) a (10), uma nova solução $x_0^{'}$ é obtida. No passo (11), se $f(x_0^{'}) < f(x_0)$ (considerando a equação 10) substituímos x_0 por $x_0^{'}$, caso contrário continuamos com a solução atual. Uma vez efetuados os passos de (6) até (11), é verificado se a solução x_0 é melhor do que x_{best}, sendo x_{best} a melhor solução obtida, considerando as w iterações (passo 2). Caso isto ocorra, define-se $s = 1$ e $x_{best} = x_0$, caso contrário, define-se $s = s+1$ e define-se $x_0 = x_{best}$ e a busca continua até que seja atingida a vizinhança máxima. E uma vez atingida tal vizinhança, gera-se uma nova solução x_0 (passo 4) e novamente aplicam-se os passos de (3) até (12), sendo estes passos repetidos por *Maximo_Iteracoes*.

4. Resultados Computacionais

A presente seção contém os resultados computacionais, obtidos a partir da aplicação da formulação e dos dois algoritmos apresentados na seção 3. A formulação foi implementada utilizando o software LINGO 7.0 e os dois algoritmos foram implementados 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.

4.1 Dados Utilizados

Com a finalidade de avaliar os dois algoritmos e a formulação, utilizou-se um conjunto de instâncias, gerado a partir dos dados de domicílios da amostra do censo demográfico de 2000 do estado de Pernambuco. Mais especificamente, para esta unidade da federação foram sorteadas 30 áreas de ponderação. 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. Em uma fase posterior, dando continuidade à geração das trinta instâncias, selecionou-se em cada uma das áreas de ponderação um conjunto de registros de domicílios (variando entre 20 e 400), definindo, desta forma, as 30 instâncias utilizadas neste trabalho. Tais registros de domicílios correspondem aos objetos.

Observamos ainda, que em cada um desses registros, foram escolhidas 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 (quantitativa). Tais variáveis foram definidas previamente e utilizadas no cálculo das distâncias d_{ij} utilizadas no processo de agrupamento (obtidas considerando as equações da seção 2.1). Maiores informações sobre conceitos de área e sobre as variáveis consideradas podem ser obtidas consultando: *Metodologia do Censo Demográfico 2000* / IBGE – Rio de Janeiro : IBGE, 2003, Relatórios Metodológicos.

4.2 Resultados e Análise

Na tabela 1 apresentamos os resultados obtidos a partir da aplicação da formulação e dos algoritmos genéticos e VNS. Na primeira coluna desta tabela temos a identificação da instância descrita por **Area_*x*_*y*_*z***, sendo ***x*** o número da área de ponderação, ***y*** o número de domicílios selecionados (objetos) e ***z*** o número de grupos formados. Nas colunas dois e três, temos, respectivamente, a melhor solução viável (***Sfo***) obtida a partir da formulação (considerando tempo máximo de processamento de dez horas) e o número variáveis da formulação. Na coluna quatro temos o valor da solução obtida (***SfAG***) a partir do algoritmo genético e na coluna cinco e o seu tempo de processamento em segundos. Nas colunas seis e sete, temos respectivamente, a solução obtida (***SfVNS***) a partir do algoritmo VNS e seu tempo de processamento em segundos. E finalmente, nas colunas oito, nove e dez, os valores percentuais representam o ***gap*** entre as soluções advindas dos algoritmos e as soluções advindas da formulação, sendo o ***gap*** definido da seguinte forma: $gap_{VNS}^{AG} = (|\, Sf_{AG} - Sf_{VNS} \,| / Min(Sf_{AG}, Sf_{VNS}))$, $gap_{AG}^{FO} = (|\, Sf_{Fo} - Sf_{AG} \,| / Sf_{FO}$ e $gap_{VNS}^{FO} = (|\, Sf_{Fo} - Sf_{VNS} \,| / Sf_{FO}$

Os parâmetros definidos para estes algoritmos foram os seguintes: (***AG***) *Probabilidade_Cruzamento*=0.50, P*robabilidade_Mutação*=0.03, P*opulação*=100, *Gerações*=500 e *Periodicidade_Path_Relinking*=20 e (***VNS***) *Maximo_Trocas*=100 *Maximo_Iterações*=500, *Maximo_Vizinhanças*=3.

Pelos resultados apresentados na tabela 1, podemos fazer as seguintes considerações:

- No que concerne à performance dos dois algoritmos, é possível observar uma vantagem do algoritmo genético em relação ao VNS. Em 22 das 30 instâncias consideradas (73,3%), o algoritmo genético produziu soluções melhores que o VNS. E apenas na primeira instância os dois algoritmos produziram a solução ótima.
- A melhor solução produzida pelo algoritmo VNS em relação ao algoritmo genético foi bservada na instância de número 16 com um *gap* de 2,65%. No caso do algoritmo genético, a melhor solução produzida em relação ao VNS foi observada na instância de número 22, com *gap* de 3,92%.
- *gap* entre as soluções advindas dos dois algoritmos foi em média de 1,34%, observando-se, que em 28 das 30 instâncias, este *gap* foi inferior a 3%. Tal fato reforça o desempenho similar dos dois algoritmos, que produziram soluções próximas.
- Tanto o AG, quanto o VNS, produziram soluções de qualidade razoavelmente superior àquelas advindas da formulação exata. Os "gaps" médios entre as soluções da formulação e os algoritmos genético e VNS foram respectivamente de 26,55% e 26,03%. Ademais, as soluções dos algoritmos foram obtidas às expensas de um tempo computacional bem inferior quando comparado ao tempo máximo de **dez horas**, fixado previamente para a formulação.
- A dificuldade de obtenção de soluções exatas ou soluções viáveis de boa qualidade, por parte da formulação, foi possivelmente decorrente da característica de simetria associada às variáveis binárias y_i^g (determinam se o i-ésimo objeto está associado ao g-ésimo grupo) e da não-linearidade associada ao problema. Ou seja, inicialmente, sem considerar a linearização introduzida através da variáveis x_{ij}^g , a função objetivo seria definida por: $\sum_{g=1}^{K}\sum_{i=1}^{n}\sum_{j=1, j\neq i}^{n} d_{ij} y_i^g . y_j^g$

Tabela 1 – Resultados da Formulação e dos Algoritmos VNS e Genético

Instância	Sfo	Variáveis	SfAG	Tempo	SfVNS	Tempo	Gap $_{AG\_VNS}$	Gap $_{Fo\_AG}$	Gap $_{Fo\_VNS}$
Area_1_20_ 2	**32,08**	420	**32,08**	48	**32,08**	9	0,00%	0,00	0,00
Area_2_30_ 3	**41,93**	1395	**38,40**	168	**38,28**	43	0,33%	8,42	8,72
Area_3_30_ 3	**44,36**	1395	**38,11**	147	**38,03**	44	0,20%	14,09	14,26
Area_4_40_ 3	**86,76**	2460	**64,84**	169	**65,89**	45	1,62%	25,27	24,06
Area_5_40_ 4	**60,65**	3280	**51,79**	153	**52,60**	105	1,57%	14,60	13,26
Area_6_50_ 4	**92,00**	5100	**75,63**	158	**75,46**	87	0,22%	17,80	17,98
Area_7_50_ 3	**175,42**	3825	**141,45**	177	**143,39**	44	1,37%	19,36	18,26
Area_8_50_2	**265,38**	2550	**231,80**	162	**232,68**	16	0,38%	12,65	12,32
Area_9_50_4	**229,09**	5100	**85,96**	160	**86,49**	88	0,62%	62,48	62,24
Area_10_60_3	**393,39**	5490	**162,30**	202	**162,94**	46	0,39%	58,74	58,58
Area_11_60_4	**272,44**	7320	**120,43**	200	**120,69**	100	0,21%	55,79	55,70
Area_12_60_5	**207,01**	9150	**97,84**	161	**98,86**	161	1,04%	52,73	52,24
Area_13_70_3	**275,76**	7455	**219,68**	149	**220,70**	48	0,47%	20,34	19,97
Area_14_70_4	**292,52**	9940	**163,32**	156	**167,47**	92	2,54%	44,17	42,75
Area_15_70_4	**233,44**	9940	**160,63**	147	**162,64**	90	1,25%	31,19	30,33
Area_16_70_5	**189,91**	12425	**129,67**	166	**126,32**	148	2,65%	31,72	33,49
Area_17_80_3	**390,31**	9720	**329,52**	159	**334,89**	48	1,63%	15,58	14,20
Area_18_80_3	**399,95**	9720	**341,00**	157	**343,34**	47	0,68%	14,74	14,15
Area_19_80_4	**344,68**	12960	**227,61**	138	**232,13**	93	1,99%	33,97	32,65
Area_20_90_2	**763,14**	8190	**735,14**	151	**742,47**	17	1,00%	3,67	2,71
Area_21_90_3	**565,81**	12285	**458,25**	167	**460,61**	50	0,51%	19,01	18,59
Area_22_90_4	**367,30**	16380	**263,19**	142	**273,52**	98	3,92%	28,34	25,53
Area_23_100_2	**905,00**	10100	**868,93**	157	**873,60**	18	0,54%	3,99	3,47
Area_24_100_3	**636,71**	15150	**531,03**	157	**537,24**	51	1,17%	16,60	15,62
Area_25_100_4	**611,90**	20200	**368,12**	165	**376,45**	97	2,26%	39,84	38,48
Area_26_150_3	**2121,60**	33975	**1267,89**	172	**1302,27**	61	2,71%	40,24	38,62
Area_27_200_2	**3825,93**	40200	**3694,10**	169	**3660,77**	30	0,91%	3,45	4,32
Area_28_200_3	**3052,91**	60300	**2320,31**	164	**2261,81**	77	2,59%	24,00	25,91
Area_29_300_3	**8420,57**	135450	**4979,24**	178	**4898,92**	125	1,64%	40,87	41,82
Area_30_200_5	**1970,09**	100500	**1127,71**	175	**1171,77**	180	3,91%	42,76	40,52

Bibliografia

Bolfarine, H. e Bussab, Wilton O. (2005). Elementos de Amostragem. ABE – Projeto Fisher. Editora Edgard Blücher.

Buriol, L.S., Resende M.G.C., Ribeiro, C. C e Thorup M. (2005). A hybrid genetic algorithm for the weight setting problem in OSPF/IS-IS routing. Networks, **46**, 1, 36-56.

Bussab, W.O e Morettin P.A. (2003). Estatística Básica. 5a Edição. Editora Saraiva.

Friedman, J. H. e Meulman J.J (2004). Clustering objects on subsets of attributes. *Journal Royal Statistics Society* B, Part 4, **66**, 815-849.

Glover, F. e Kochenberger, G. A. (2002). *"Handbook of Metaheuristic"*, First Edition Norwell: K1uwer Academic Publishers.

Gordon, A. (1999). *Classification*, 2nd edn. London: Chapman and Hall-CRC.

Hansen, P. and Jaumard, B. (1997). Cluster Analysis and Mathematical Programming. *Mathematical Programming*, **79**, 191-215.

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

Holland, J.H. (1975). *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor.

Hubert, L. J., Arabie, P. e Meulman, J.J. (2001). *Combinatorial Data Analysis: Optimization by Dynamic Programming*. Philadelphia: Society for Industrial and Applied Mathematics.

Johnson A.R. e Wichern D.W. (2002). *Applied Multivariate Statistical Analysis*. Prentice Hall. Fifth Edition.

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

Mladenovic, N. e Hansen, P. (1997). Variable neighborhood search. *Computers Operations Research*, **24**, 1097-1100.

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