

UM ALGORITMO HEURÍSTICO APLICADO AO PROBLEMA DE OFERTA DE PRODUTOS EM CAMPANHAS DE MARKETING DIRETO

T.A. Oliveira¹, V.N. Coelho³, M.J.F. Souza¹, D.L.T. Boava¹,
E.M.F.M. Boava¹, I.M. Coelho² e B.N. Coelho¹

¹ Universidade Federal de Ouro Preto, Ouro Preto, MG, 35400-000, Brasil

² Universidade Federal Fluminense, Niterói, RJ, 24210-240, Brasil

³ Universidade Federal de Minas Gerais, Belo Horizonte, MG, 31270-901, Brasil

{thaysoliveira7,vncoelho}@gmail.com, marccone@iceb.ufop.br

{profboava,profamacedo,igor.machado}@yahoo.com.br, brunonazario@gmail.com

Abstract. *This paper focuses on the targeted offers problem in direct marketing. The objective is to maximize the feedback of customers purchases, providing products for customers who have the highest probability of positively accepting the offer and, at the same time, minimizing the operating costs of the campaign. Given the combinatorial nature of the problem and the large volume of data, involving instances with up to one million customers, approaches solely based on mathematical programming methods, said exact, appear limited and infeasible. Thus, the solution of actual cases is typically made by heuristics. In this paper, the use of a heuristic algorithm, based on the Greedy Randomized Adaptive Search Procedures (GRASP), Iterated Local Search (ILS) and Variable Neighborhood Descent (VND) procedures, is proposed. Computational experiments performed on a set of test problems from the literature show that the proposed algorithm was able to produce competitive solutions. Additionally, the proposed algorithm generates final solutions with low variability, which shows its robustness.*

KEYWORDS: *Direct Marketing, Heuristics, Campaign, Customers.*

Resumo. *Este trabalho tem seu foco no problema de marketing direto com oferta de produtos. O objetivo é maximizar o feedback de compras dos clientes, oferecendo os produtos para os clientes que possuem a maior probabilidade de aceitarem a oferta de forma positiva e, ao mesmo tempo, minimizando os custos operacionais da campanha. Dada a natureza combinatória do problema e o grande volume de dados, envolvendo instâncias com até um milhão de clientes, abordagens exclusivamente embasadas em métodos de programação matemática, dito exatos, mostram-se limitadas e inviáveis. Desta forma, a solução de casos reais normalmente é feita por meio de métodos heurísticos. Neste trabalho, propõe-se a utilização de um algoritmo heurístico baseado nos conceitos dos procedimentos Greedy Randomized Adaptive Search Procedures (GRASP), Iterated Local Search (ILS) e Variable Neighborhood Descent (VND). Experimentos com um conjunto de problemas-teste da literatura mostram que o algoritmo desenvolvido produziu soluções competitivas. Adicionalmente, o algoritmo desenvolvido gera soluções finais com baixa variabilidade, o que mostra sua robustez.*

PALAVRAS-CHAVE: *Marketing Direto, Heurística, Campanha, Clientes*

1 Introdução

Este trabalho trata do Problema de Marketing Direto com Oferta de Produtos – PMDOP. Neste problema, busca-se determinar para quais clientes deve-se oferecer determinados produtos em uma campanha, ou seja, os clientes potenciais para receber uma determinada oferta, maximizando os lucros e respeitando as restrições operacionais.

A execução de campanhas de Marketing Direto – MD produzem, ao longo do tempo, importantes dados e informações que devem ser analisadas para o suporte à tomada de decisão.

Atualmente, com o surgimento das análises envolvendo grandes volumes de dados (*big-data analysis*) (Krishnan, 2013), essa tarefa mostra-se cada vez mais extensa e complexa (Moro *et al.*, 2012). Surge, assim, a necessidade de integração entre as áreas de MD e Pesquisa Operacional – PO. Neste âmbito, salienta-se a crescente preocupação ligada aos novos modelos de resposta ao marketing, do inglês “marketing response models”, no qual os administradores e pesquisadores tentam, ao máximo, compreender como os consumidores respondem às iniciativas do marketing (Hanssens *et al.*, 2005). Essa classe de problemas apresenta-se praticamente intratável sem auxílio de ferramentas adequadas. Modelos eficientes e inteligentes vêm sendo adotados na literatura desde os anos 80 (Guadagni e Little, 1983), e vem se tornando cada vez mais frequentes no cotidiano das empresas e empreendedores. Neste contexto, a PO possui um grande papel no atendimento e satisfação dos clientes, de forma que recentes trabalhos da literatura têm relacionado ambas as áreas, como pode ser visto em Ling e Li (1998); Cohen (2004); Bhaskar *et al.* (2009); Nobibon *et al.* (2011); Chun (2012); Olson e Chae (2012); Seret *et al.* (2012).

O atual cenário das organizações mostra-se bastante competitivo e marcado por profundas mudanças devido ao crescente processo de globalização. Isso favorece o surgimento de novas estratégias competitivas. O advento das novas tecnologias confere uma nova roupagem ao ambiente da organização, uma vez que facilita o processo de informação e comunicação. Desta forma, verificam-se cada vez mais empresas que procuram estar em consonância com as mudanças do ambiente. Na tentativa de se adaptar a este meio, as organizações desenvolvem estratégias que direcionam para o sucesso organizacional.

Ling e Li (1998) afirmam o interesse das empresas de varejo, bancos e companhias de seguro em utilizar técnicas de mineração de dados para identificar clientes promissores.

Nesse contexto, ressalta-se que o desenvolvimento de uma técnica eficiente para resolução do PMDOP é de grande importância. De fato, um bom planejamento de uma campanha de oferta de produtos implica em um melhor aproveitamento dos recursos e, conseqüentemente, um aumento dos lucros. A relevância desta pesquisa encontra-se em termos práticos e teóricos. Em termos práticos, o trabalho pode contribuir para grandes organizações de diversos ramos diferentes, incluindo empresas, indústrias, bancos, empreendedores. Em termos teóricos, serve de fundamento para trabalhos posteriores nessa área, uma vez que existem poucos estudos que abordam a concatenação de um problema com uma abordagem via PO.

Desta forma, propõe-se, neste presente trabalho, um algoritmo híbrido inspirado nas metaheurísticas *Greedy Randomized Adaptive Search Procedure* (GRASP) Feo e Resende (1995) e Busca Geral em Vizinhança Variável (GVNS), que utiliza como busca local o método Descida em Vizinhança Variável (VND). Do procedimento GRASP utiliza a fase

de construção para produzir soluções viáveis e de boa qualidade rapidamente. O GVNS foi escolhido devido a sua simplicidade, eficiência e capacidade natural de sua busca local (método VND) para lidar com diferentes vizinhanças.

O restante deste trabalho está organizado como segue. A Seção 2 apresenta o problema abordado em detalhes. A Seção 3 detalha o algoritmo proposto para resolver o PMDOP. A Seção 5 mostra os resultados dos experimentos computacionais e, finalmente, a Seção 6 conclui o trabalho.

2 O Problema de Marketing Direto com Oferta de Produtos

Um breve histórico do Problema de Marketing Direto com Oferta de Produtos – PMDOP é apresentado a seguir.

Cohen (2004) apresenta um estudo de caso do PMDOP aplicado a um banco internacional de grande porte, utilizando vários canais de ofertas. Foram consideradas onze tipos de ofertas, cinco de investimentos, três de empréstimos e três delas baseadas em ofertas dia a dia (*day to day offers*). Aproximadamente 2.500.000 milhões de clientes em potencial foram considerados na campanha. A partir de modelos de resposta conhecidos da literatura, foi determinado o benefício de cada oferta $j \in O$ para cada cliente $i \in C$. Esses valores foram estimados com base na probabilidade do cliente i aceitar positivamente a oferta j , bem como o possível lucro associado. Foram consideradas restrições de investimento máximo disponível para cada produto, investimento máximo disponível para a campanha, retorno mínimo esperado e limitações para cada canal de oferta. Os resultados e análises validaram a proposta, que se mostrou promissora para aumentar o sucesso das futuras campanhas.

Bhaskar *et al.* (2007, 2009) propõem um modelo utilizando lógica *fuzzy*, incluindo assim, certa incerteza ao resultado final. Essa abordagem mostrou-se aplicável e eficiente, dada a imprecisão na previsão da reação dos consumidores.

O modelo apresentado por Nobibon *et al.* (2011), estado da arte, engloba restrições similares às de Cohen (2004), incorporando novas restrições, como o número mínimo de unidades ofertadas de cada produto durante a campanha, conhecida como *minimum-quantity commitment* (MQC). Além disso, esse trabalho incluiu novas restrições de ofertas de produtos, nas quais cada cliente poderia receber mais de uma oferta em uma dada campanha. Todavia, ainda foram utilizados apenas custos fixos. Os autores desenvolveram oito algoritmos para resolver o problema, sete deles baseados em métodos exatos e um deles baseado no procedimento heurístico Busca Tabu – BT Glover e Laguna (1997). Dentre os métodos testados, o *Branch-and-Price* Peeters e Degraeve (2006) e a BT apresentaram os melhores resultados.

O modelo de Nobibon *et al.* (2011) pode ser reescrito da seguinte forma: Têm-se um conjunto de clientes $C = [c_1, c_2, \dots, c_m]$ e um conjunto de oferta de produtos $O = [o_1, o_2, \dots, o_n]$, associa-se um custo $d_{ij} > 0$ e um benefício $r_{ij} \geq 0$ caso a oferta do produto $j \in O$ seja direcionada para o cliente $i \in C$. Para cada cliente $i \in C$, existe um número M_i que indica o número máximo de produtos que podem ser recebidos pelo cliente i . Para cada oferta de produto $j \in O$, existe um número mínimo O_j^{min} de produtos que devem ser oferecidos durante a campanha, lucro mínimo esperado R (conhecido como *hurdle rate*), investimento B_j disponível para o produto j e, por fim, um custo fixo f_j se o produto j for utilizado na campanha de divulgação.

Na Figura 1 pode ser observado um exemplo de uma simples solução para o PM-DOP com $m=3$ e $n=2$, ou seja, com três clientes e duas diferentes ofertas de produtos. Para exemplificar os valores obtidos, expressaremos os valores em dólares (US\$). Verifica-se que o produto 1 foi oferecido para os clientes 1, 2 e 3, agregando um custo igual a 7\$ e 10\$ de benefício, já o produto 2 foi oferecido apenas para os clientes 1 e 3, com um custo, também, igual a 7\$ e benefício total de 12\$. Logo, é possível perceber que todas as restrições do problema são atendidas, dando um retorno total igual 22\$ e investimento de campanha igual a 19\$ = 7\$ + 7\$ + 3\$ + 2\$ (3\$ e 2\$ relativos aos custos fixos dos produtos 1 e 2, respectivamente), ou seja, foi obtido um lucro de aproximadamente 15,78%. Por fim, ressalta-se que as restrições individuais de cada produto foram atendidas, os investimentos máximos disponíveis para cada produto foram respeitados e um número mínimo de ofertas foi realizado.

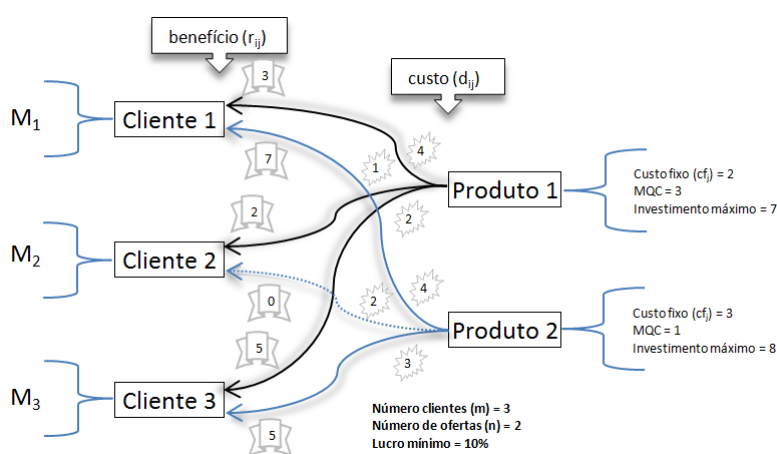


Figura 1. Exemplo PMDOP.

O PMDOP pode ser reduzido a um caso especial do Problema da Mochila 0-1 Múltipla – PMM. Um estudo geral sobre essa classe de problema pode ser encontrado em Kellerer *et al.* (2004). De fato, essa analogia pode ser vista se cada cliente for considerado como uma mochila $m \in C$, cada objeto da mochila pode ser visto como um produto único $j \in I$, onde $I = [i_1, i_2, \dots, i_n, \dots]$, de forma que pode $\exists i_x = i_y | x \neq y$, ou seja, podem existir produtos iguais dentro da mochila. Assim sendo, como o PMM pertence à classe NP-difícil Papadimitriou e Steiglitz (1998), o PMDOP também o é.

3 Metodologia

3.1 Representação de uma solução

Uma solução é representada por uma matriz de booleanos $R_{|C| \times |O|}$, em que: C representa o conjunto de clientes disponíveis para a campanha de produtos e O representa os produtos cotados para a campanha. Caso o valor da célula $s_{i,j} | i \in P, j \in O$ seja igual a “1” o produto j será ofertado para o cliente 1, caso contrário, o valor da célula será “0”.

A Figura 2 mostra a representação de uma solução para o PMDOP, levando como a base o exemplo apresentado na Figura 1.

3.2 Avaliação de uma Solução

Uma solução s é avaliada por uma função f , a ser maximizada, dada pela Eq. 1. Esta função é baseada em penalidades, sendo composta de duas parcelas: f^{obj} e f^{inv} .

	Cliente	Produto 1	Produto 2
$s =$	1	1	1
	2	1	0
	3	1	1

Figura 2. Representação de uma solução para o PMDOP

$$f(s) = f^{obj}(s) + f^{inv}(s) \quad (1)$$

A parcela f^{obj} , dada pela Eq. 2, refere-se à função objetivo propriamente dita do problema, sendo C o conjunto de clientes e P o conjunto de produtos. Associa-se um custo $d_{ij} > 0$ e um benefício $r_{ij} \geq 0$ caso a oferta do produto $j \in O$ seja direcionada para o cliente $i \in C$. Acrescenta-se um custo fixo f_j se o produto j for utilizado na campanha de divulgação ($y_j = 1$).

$$f^{obj}(s) = \sum_{i \in C} \sum_{j \in P} (r_{i,j} - d_{i,j}) s_{i,j} - \sum_{j \in P} f_j y_j \quad (2)$$

Já a segunda parcela f^{inv} , dada pela Eq. 3, refere-se ao atendimento das restrições operacionais do problema, descritas na Seção 2. Caso alguma dessas restrições seja violada, é atribuído um alto valor associado ao valor da função objetivo dessa solução s , pois esta solução é inviável na prática.

$$f^{inv}(s) = \sum_{i \in C} (f_i^M(s)) + \sum_{j \in P} (f_j^O(s) + f_j^B(s)) + f^R(s) \quad (3)$$

Na Eq. 3, a parcela $f_i^M(s)$ avalia s quanto à inviabilidade do cliente i receber mais ofertas do que o limite M_i , ou seja, o máximo de produtos solicitado por este cliente. Por outro lado, as parcelas $f_j^O(s)$, $f_j^B(s)$ e $f_j^{HR}(s)$ referem-se às restrições relativas aos produtos ativos e cotados para a campanha. $f_j^O(s)$ avalia s quanto ao fato do produto j ter sido ofertado para uma quantidade menor que O_j^{min} clientes; $f_j^B(s)$ avalia s quanto à situação da carteira de dinheiro (*budget* B) ter sido estourada ou não para o produto j . Finalmente, a parcela $f_j^R(s)$ verifica se o lucro *hurdle rate* (R) esperado foi alcançado. As inviabilidades não foram ponderadas em relação ao desvio de uma solução viável, mas sim com valores fixos que, em geral, inviabilizam a solução e elevam uma barreira no espaço de soluções.

3.3 Geração de uma Solução Inicial

O Algoritmo 1 apresenta a adaptação proposta neste trabalho para o procedimento guloso de Praag (2010), e, também, já utilizado por Nobibon *et al.* (2011). Tal procedimento foi transportado em um parcialmente guloso e possui a seguinte notação e dados de entrada: p_{ij} é o preço de venda em oferecer o produto j para o cliente i ; c_{ij} é o custo em oferecer o produto j para o cliente i ; f_j é o custo fixo do produto j , caso ele seja selecionado para participar da campanha; R é a taxa de lucro mínima esperada na campanha; rt

é a receita total; ct é o custo total; S é o conjunto de produtos selecionados para campanha; LC é a lista de candidatos, clientes, selecionados de forma gulosa; LRC é a lista de candidatos restrita, selecionada de forma parcialmente gulosa, de acordo com o parâmetro $\gamma \in [0, 1]$; CS é o conjunto de clientes selecionados; O_j é o número mínimo de ofertas.

As linhas 1 e 2 do Algoritmo 1 iniciam as variáveis rt , ct , V e S . Essas variáveis serão utilizadas como variáveis auxiliares durante a geração de uma solução inicial. Das linhas 3 até 7 realiza-se o cálculo dos valores de NPP para cada produto no conjunto P e clientes M . Desta forma, a variável NPP_{ij} recebe cada valor de lucro de cada oferta j para um cliente i .

Entre as linhas 8 e 23, o cálculo do “melhor” conjunto de clientes para receber as ofertas de produtos de acordo com o parâmetro de gulosidade γ é realizado. Na linha 9 é gerada a lista com os melhores clientes em ordem decrescente dos valores NPP calculados anteriormente. Tais clientes são os mais desejados pois possuem o maior valor de NPP, sendo alocados nas primeiras posições do vetor. Na linha 10 os conjuntos de clientes selecionados para um dado produto corrente j (CS_j) inicia-se vazio, já as variáveis de custos e benefícios de C_j e P_j , respectivamente, iniciam-se em zero. Da linha 11 até a linha 23 é realizado a seleção dos O_j (número mínimo de clientes para que um produto j seja utilizado na campanha). Na linha 14, um número ($\gamma * |LC_j|$) de clientes é adicionado a lista de candidatos restritos LRC_j , o tamanho dessa lista varia de acordo com os clientes remanescentes da lista LC_j . As linhas 11, 17, 22 selecionam, de forma aleatória, um cliente i pertencente a lista LRC_j e atualiza os conjuntos CS_j (adicionando o cliente i) e LC_j (removendo o cliente i). A linha 22 calcula o valor PR_j que é a soma de todos os benefícios menos os custos envolvidos na oferta do produto j para os clientes selecionados na lista CS_j .

O melhor produto j^* , que possui o maior valor PR_j e que satisfaz as restrições operacionais do problema ($[C_{j^*} \leq B_{j^*}] \& [(rt + P_{j^*}) \geq (1 + R)(ct + C_{j^*} + f_{j^*})]$), é selecionado na linha 24 do Algoritmo 1. Se este produto existe (linha 25) ele é selecionado na linha 26, na qual os valores de rt e ct são atualizados. No intervalo entre as linhas 27 e 35 os clientes selecionados para a campanha (CS_j), são inseridos na solução corrente que está sendo gerada (linha 28), em seguida o número máximo de ofertas ainda disponível para o cliente, MO_i , é atualizado. Se não existem mais ofertas disponíveis (linha 30), o cliente é retirado da lista de clientes disponíveis, removido do conjunto de clientes ativos V . Finalmente, retorna-se à linha 8.

Por outro lado, caso não existam mais clientes disponíveis (linha 25 não satisfeita), finalmente, entre as linhas 36 e 40 a solução s é refinada. Neste caso, os produtos que possuem benefícios maiores do que os custos e respeitam as restrições operacionais do problema são adicionados. Essa etapa ocorre, principalmente, porque o método de construção utilizado aloca, inicialmente, apenas os O_j mínimos clientes em cada produto selecionado para a campanha.

4 Estruturas de Vizinhança

Para explorar o espaço de soluções do problema foram utilizadas três estruturas de vizinhança, todas propostas por Nobibon *et al.* (2011). Faz-se necessário ressaltar que todos os movimentos descritos e utilizados neste trabalho mantêm a viabilidade da solução. Uma breve descrição dos movimentos é apresentado a seguir.

Algoritmo 1: ConstroiSolucaoParcialmenteGulosaNPP

Entrada: $\gamma \in [0, 1]$, Função $f(\cdot)$
Entrada: Conjunto de produtos $P \leftarrow \{1, \dots, n\}$ e conjunto de clientes $M \leftarrow \{1, \dots, m\}$

- 1 Inicia as variáveis de receita total (rt), custo total (ct), conjunto de clientes disponíveis V e produtos selecionados para campanha S .
- 2 $rt \leftarrow 0, ct \leftarrow 0, V \leftarrow M$ e $S \leftarrow \emptyset$
- 3 **para cada** $j \in P$ **faça**
- 4 **para cada** $i \in M$ **faça**
- 5 $NPP_{ij} \leftarrow \frac{p_{ij} - c_{ij}}{c_{ij}}$
- 6 **fim**
- 7 **fim**
- 8 **para todo** $j \notin S$ **faça**
- 9 $LC_j \leftarrow$ conjunto de clientes $i \in V$ em ordem decrescente em relação ao NPP_{ij}
- 10 $CS_j \leftarrow \emptyset, C_j \leftarrow 0$ e $P_j \leftarrow 0$
- 11 **para** $|CS_j| \leq O_j$ **faça**
- 12 $LRC_j \leftarrow \emptyset$
- 13 **para** $c = 0 \leq (\gamma * |LC_j|)$ **faça**
- 14 $LRC_j \leftarrow LRC_j \cup \{LC_j(c)\}$
- 15 **fim**
- 16 Selecione um cliente $i \in LRC_j$ de forma aleatória
- 17 $CS_j \leftarrow CS_j \cup \{i\}$
- 18 $LC_j \leftarrow LC_j \setminus \{i\}$
- 19 **fim**
- 20 $C_j \leftarrow$ soma dos custos c_{ij} de todos os clientes $i \in CS_j$
- 21 $P_j \leftarrow$ soma dos benefícios p_{ij} de todos os clientes $i \in CS_j$
- 22 $PR_j \leftarrow P_j - C_j - f_j$
- 23 **fim**
- 24 Selecione j^* com o maior valor $PR_{j^*} > 0$ tal que $[C_{j^*} \leq B_{j^*}] \& [(rt + P_{j^*}) \geq (1 + R)(ct + C_{j^*} + f_{j^*})]$
- 25 **se** $\exists j^*$ **então**
- 26 $rt \leftarrow rt + P_{j^*}, ct \leftarrow ct + C_{j^*} + f_{j^*}$ e $S \leftarrow S \cup \{j^*\}$
- 27 **para cada** cliente $i \in CS_{j^*}$ **faça**
- 28 $s_{ij^*} \leftarrow 1$
- 29 $MO_i \leftarrow MO_i - 1$
- 30 **se** $MO_i = 0$ **então**
- 31 $V \leftarrow V \setminus \{i\}$
- 32 **fim**
- 33 Volte para linha 8
- 34 **fim**
- 35 **fim**
- 36 **para cada** cliente ativo $i \in V$ **faça**
- 37 **para cada** $j \in S$ **faça**
- 38 Se $p_{ij} > c_{ij}$ e a oferta do produto j ao cliente i gera uma solução factível, então, $s_{ij} \leftarrow 1$
- 39 **fim**
- 40 **fim**
- 41 **retorna** s

Movimento Troca Clientes Intra - $N^{TC-Intra}(s)$: Este movimento consiste em trocar duas posições $l, m \in C$ referentes a um produto $j \in P$, tal que $s_{l,j} = s_{m,j}$ e $s_{m,j} = s_{l,j}$ de forma que as restrições operacionais do problema sejam mantidas.

Movimento Troca Clientes Inter - $N^{TC-Inter}(s)$: Similar ao movimento $N^{TC-Intra}(s)$, porém, efetuam-se apenas trocas envolvendo produtos distintos $i, j \in P$, ou seja, $s_{l,i} = s_{m,j}$ e $s_{m,j} = s_{l,i}$.

Movimento Troca Produtos - $N^{TP}(s)$: Consiste em trocar duas colunas distintas $i, j \in P$ tais que $y_i = 1$ e $y_j = 0$ ou vice e versa. Neste movimento um produto que não era utilizado na campanha passa a ser utilizado e um outro produto ativo torna-se inativo. O produto a ser inserido na campanha deve atender as restrições operacionais, financeiras e respeitar a disponibilidade dos clientes.

4.1 Algoritmo Proposto

O algoritmo proposto, denominado *GGVNS*, combina os procedimentos GRASP e GVNS. Do procedimento GRASP utiliza a fase de construção para produzir soluções viáveis e de boa qualidade rapidamente. O GVNS foi escolhido devido a sua simplicidade, eficiência e capacidade natural de sua busca local, feita pelo método *Variable Neighborhood Descent* – VND, para lidar com diferentes vizinhanças, como relatado em Souza *et al.* (2010).

O pseudocódigo do algoritmo *GGVNS* está esquematizado no Algoritmo 2. Neste algoritmo, *GRASPStartmax* representa a quantidade de iterações em que a fase de construção do procedimento GRASP é aplicada e *IterMax* indica o número máximo de iterações realizadas em um dado nível de perturbação.

Algoritmo 2: *GGVNS*

Entrada: Solução s , *GRASPStartmax*, *IterMax*, Função $f(\cdot)$

Saída: Solução s^* de qualidade possivelmente superior à s de acordo com a função f

```

1  $s_0 \leftarrow s$ 
2  $s_0 \leftarrow$  Melhor Solução em GRASPStartmax iterações do procedimento
   ConstroiSolucaoParcialmenteGulosaNPP()
3  $s^* \leftarrow \text{VND}(s_0, f)$ 
4  $p \leftarrow 0$ 
5 enquanto critério de parada não satisfeito faça
6    $iter \leftarrow 0$ 
7   enquanto  $iter < \text{IterMax}$  e critério de parada não satisfeito faça
8      $s' \leftarrow \text{Refinamento}(s^*, p, f)$ 
9     se  $s'$  for melhor que  $s^*$  de acordo com a função  $f$  então
10       $s^* \leftarrow s'$ ;
11       $p \leftarrow 0$ ;
12       $iter \leftarrow 0$ 
13     fim
14     senão
15        $iter \leftarrow iter + 1$ 
16     fim
17   fim
18    $p \leftarrow p + 1$ 
19 fim
20 retorna  $s^*$ 

```

A solução inicial s_0 (linha 2 do Algoritmo 2) é gerada aplicando-se *GRASP* a fase

de construção do procedimento GRASP, isto é, o procedimento *ConstroiSolucaoParcialmenteGulosaNPP* descrito na seção 3.3.

A busca local é feita pelo procedimento VND, Algoritmo 4, usando-se os dois movimentos descritos na seção 4, ou seja, os movimentos das vizinhanças $N^{TC-Intra}$, $N^{TC-Inter}$. Já na fase de perturbação, além desses dois movimentos, o movimento N^{TP} também foi utilizado. Esses movimentos são aplicados de forma aleatória de acordo com o procedimento *SelecionaVizinhança* (linha 2 do Algoritmo 3).

Algoritmo 3: Refinamento

Entrada: r vizinhanças: N^{AR} , N^T

Entrada: Solução Inicial s , Nível p e Função de Avaliação f

Saída: Solução s

```
1 para  $i \leftarrow 1$  até  $p + 2$  faça
2    $k \leftarrow$  SelecionaVizinhança( $r$ )
3    $s' \leftarrow$  Perturbação( $s, k$ )
4 fim
5  $s \leftarrow$  VND( $s', f$ )
6 retorna  $s$ 
```

Algoritmo 4: VND

Entrada: r vizinhanças na ordem aleatória: N^{AR} , N^T

Entrada: Solução Inicial s e Função de Avaliação f

Saída: Solução s

```
1  $k \leftarrow 1$ 
2 enquanto  $k \leq r$  faça
3   Encontre o melhor vizinho  $s' \in N^{(k)}(s)$ 
4   se  $f(s') < f(s)$  então
5      $s \leftarrow s'$ ;
6      $k \leftarrow 1$ 
7   fim
8   senão
9      $k \leftarrow k + 1$ 
10  fim
11 fim
12 retorna  $s$ 
```

5 Resultados Computacionais

O algoritmo GGVNS proposto foi implementado na linguagem C++ e testado em um computador DELL XPS 8300 Intel Core i7-2600, 8MB Cache, 3.4GHz, 16GB RAM, sob o sistema operacional Ubuntu 12.04 Foram realizadas três baterias de experimentos. Os experimentos foram realizados de acordo com os padrões já apresentados na literatura.

Para testar o algoritmo proposto foram considerados 4 grupos de instâncias da literatura de pequeno (S3), médio (M1 e M2) e grande porte (L). Cada grupo contém 3 subgrupos (5, 10 e 15 produtos) com 18 instâncias cada, que diferem entre si com relação aos custos e limites das restrições operacionais. Desta forma, 216 execuções foram feitas em cada bateria de testes, exatamente, uma execução para cada instância.

A primeira bateria de experimentos foi realizada de forma a verificar, puramente, o poderio do método de construção de uma solução inicial parcialmente gulosa proposto

(descrito no Algoritmo 1). A segunda bateria utilizou os mesmos parâmetros da primeira, porém, o método de busca-local VND (Algoritmo 4) foi utilizado após cada geração de uma solução do GRASP. Finalmente, a última bateria de experimentos aplicou os mesmos parâmetros da segunda; todavia, uma última etapa de refinamento (com base no procedimento descrito no Algoritmo 2) foi utilizada.

Os melhores resultados conhecidos, f_i^* , de acordo com a função objetivo apresentada na Seção 3.2, foram fornecidos pelos autores do trabalho Nobibon *et al.* (2011). A métrica utilizada para avaliar o desempenho do algoritmo foi o gap, $gap_i^n = \frac{f_i^* - f_i^n}{f_i^*}$, considerando f_i^* como o melhor valor conhecido para o problema-teste i e f_i^n o valor obtido pelo algoritmo/método n durante sua i -ésima execução.

A Tabela 1 apresenta os resultados obtidos em cada uma das três baterias de experimentos. A Coluna “n” indica o método utilizado na resolução. “Init.” e “H8” referem-se aos métodos de construção guloso e a heurística Busca Tabu de Nobibon *et al.* (2011). As colunas “S3”, “M1”, “M2” e “L” indicam o grupo de problemas-teste utilizado e, logo abaixo, indicam-se os subgrupos formados com diferentes números de produtos a serem ofertados em cada um desses grupos.

Tabela 1. Resultados computacionais – GAP (%)

n	S3			M1			M2			L		
	5	10	15	5	10	15	5	10	15	5	10	15
RESULTADOS DA LITERATURA Nobibon <i>et al.</i> (2011)												
Init.	19.20	17.85	18.34	17.90	16.20	18.36	18.92	19.23	17.84	26.22	24.86	25.65
H8	6.86	6.52	7.76	7.22	8.54	7.60	9.75	9.58	9.11	10.86	11.04	10.23
γ	BATERIA DE EXPERIMENTOS I – GRASP PURO											
0	21.69	23.53	25.77	21.32	27.77	28.02	23.26	29.55	31.45	29.94	31.78	33.15
0.2	21.72	23.52	25.82	21.39	27.70	27.99	23.28	29.52	31.43	31.22	31.77	33.17
0.4	21.69	23.52	25.78	21.37	27.71	27.99	23.21	29.55	31.44	31.42	31.77	33.15
0.6	21.69	23.54	25.77	21.32	27.67	27.97	23.25	29.55	31.44	32.09	31.78	33.15
BATERIA DE EXPERIMENTOS II – GRASP + VND												
0	18.15	17.23	18.74	19.22	17.57	18.02	18.55	19.16	16.15	25.95	24.04	25.75
0.2	17.72	18.22	18.82	18.19	16.15	17.29	17.95	20.15	15.14	26.01	24.18	23.98
0.4	17.99	17.24	17.98	17.58	16.23	17.56	17.41	18.44	16.54	27.92	23.76	23.79
0.6	17.82	18.88	17.78	18.15	15.19	15.97	16.93	17.89	16.11	24.19	23.12	23.69
BATERIA DE EXPERIMENTOS III – GGVNS												
0	10.68	6.98	7.77	9.72	9.12	12.15	11.29	9.90	11.15	12.99	12.29	12.15
0.2	6.90	7.14	7.92	7.22	8.67	7.69	10.26	9.59	9.34	11.24	10.94	10.74
0.4	6.79	6.45	7.58	8.47	9.10	7.89	9.68	9.76	9.91	10.86	11.22	10.11
0.6	6.77	6.50	7.49	7.24	8.42	7.63	9.85	9.78	9.44	11.51	11.93	10.22

Analisando-se a Tabela 1, verifica-se que o algoritmo proposto, GGVNS, foi capaz de obter um GAP abaixo ou igual àquele apresentado por Nobibon *et al.* (2011) em até oito grupos de problemas-teste, obtendo um menor GAP em seis grupos de instâncias. Apesar do método guloso ($\gamma = 0$) ter sido reproduzido de forma idêntica à literatura, os resultados obtidos não foram iguais. O método parcialmente guloso proposto neste trabalho foi validado quando utilizado em conjunto com um procedimento de busca local,

como é o caso apresentado na bateria de experimentos II. Esse resultado é inteiramente consistente, em vista do potencial do GRASP em gerar soluções iniciais em diferentes bacias de atração.

6 Conclusões

Neste trabalho foi discutido um Problema de Marketing Direto com suposições realistas. Em particular, este trabalho teve seu foco no Problema de Marketing Direto com Oferta de Produtos (PMDOP). Com o estudo envolvendo as áreas de MD e PO, percebeu-se que as áreas de Marketing e PO se relacionam de maneira que juntas conseguem dar suporte uma a outra. Enquanto o MD fica responsável pelo relacionamento com os clientes, construindo o banco de dados para interagir e potencializar os clientes, a PO é voltada para resolução de problemas reais, encontrando a melhor alternativa para solução de tais problemas.

Apesar da sua relevância prática, essa variante tem recebido pouca atenção da literatura. Em virtude de sua dificuldade de solução na otimalidade, foi proposto um algoritmo heurístico de busca em trajetória, denominado *GGVNS*, que combina o poderio do *GRASP* e o método *GVNS*, percorrendo um espaço de busca de soluções inteiras viáveis.

Para validar o algoritmo proposto, foi utilizado um conjunto de instâncias reais da literatura composto de 324 instâncias gerado por Nobibon *et al.* (2011). Porém, apenas 216 foram utilizadas nesse trabalho, pois o restante se apresentaram como instâncias muito pequenas e de fácil resolução. Três baterias de experimentos foram realizadas. A primeira e a segunda baterias buscaram validar a adaptação do método parcialmente guloso proposto neste trabalho. Na primeira, o procedimento parcialmente guloso foi executado puramente, enquanto na segunda bateria de experimentos, aplica-se também o procedimento de busca local VND às soluções geradas. A última bateria buscou validar o método *GGVNS* por completo, incluindo a geração e o refinamento de uma solução. Os resultados computacionais indicam a superioridade do algoritmo heurístico proposto, o qual foi capaz de apresentar melhores soluções do que a literatura. Desta forma, a presente proposta mostra-se validada e promissora.

Como extensões deste trabalho, propõe-se o desenvolvimento de um modelo mais abrangente para o PMDOP. Como exemplo, pode-se incluir restrições de não-linearidade envolvendo os custos quando oferta-se uma grande quantidade de produtos, ou seja, o custo de oferecer o mínimo de ofertas O_j do produto j poderia variar de forma não-linear quando o número de clientes atendidos aumenta. Da mesma forma, clientes que recebem mais que uma oferta poderiam ter uma parcela de redução nos custos envolvidos. Tais restrições são inteiramente consistentes e representam casos reais que ocorrem no dia a dia das campanhas de ofertas de produtos. Finalmente, no aspecto computacional, propõe-se desenvolver novas estruturas de vizinhanças e perturbação de uma solução. Propõe-se, também, a adaptação do presente algoritmo para resolver outros problemas da área de Marketing e MD, tais como aqueles referenciados no decorrer deste trabalho. Isso contribuiria para fortalecer e incentivar a utilização de técnicas da Pesquisa Operacional aplicada a problemas de Marketing, em especial, aos problemas de MD.

Agradecimentos

Os autores agradecem às agências FAPEMIG e CNPq, assim como à Universidade Federal de Ouro Preto pelo apoio ao desenvolvimento deste trabalho.

Referências

- Bhaskar, Tarun; Sundararajan, Ramasubramanian e Krishnan, Puthu G. (2007). A fuzzy mathematical programming approach for cross-sell optimization in retail banking. *Proceedings of the 25th IASTED International Multi-Conference: artificial intelligence and applications*, AIAP'07, p. 554–560, Anaheim, CA, USA. ACTA Press.
- Bhaskar, Tarun; Sundararajan, Ramasubramanian e Krishnan, Puthu G. (2009). A fuzzy mathematical programming approach for cross-sell optimization in retail banking. *Journal of the Operational Research Society*, v. 60, n. 5, p. 717–727.
- Chun, Young H. (2012). Monte carlo analysis of estimation methods for the prediction of customer response patterns in direct marketing. *European Journal of Operational Research*, v. 217, n. 3, p. 673 – 678.
- Cohen, M.-D. (2004). Exploiting response models: optimizing cross-sell and up-sell opportunities in banking. *Informations Systems*, v. 29, n. 4, p. 327–341.
- Feo, T. A. e Resende, M. G. C. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, v. 6, p. 109–133.
- Glover, F. e Laguna, M. (1997). *Tabu Search*. Kluwer Academic Publishers, Boston.
- Guadagni, Peter M. e Little, John D. C. January(1983). A logit model of brand choice calibrated on scanner data. *Marketing Science*, v. 3, n. 2, p. 203–238.
- Hanssens, Dominique M.; Leeflang, Peter S. H. e Wittink, Dick R. July(2005). Market response models and marketing practice: Research articles. *Appl. Stoch. Model. Bus. Ind.*, v. 21, n. 4-5, p. 423–434.
- Kellerer, Hans; Pferschy, Ulrich e Pisinger, David. (2004). *Knapsack problems*. Springer. ISBN 978-3-540-40286-2.
- Krishnan, Krish. (2013). Chapter 1 - introduction to big data. *Data Warehousing in the Age of Big Data*, p. 3 – 14. Morgan Kaufmann, Boston.
- Ling, Charles X. e Li, Chenghui. (1998). Data Mining for Direct Marketing: Problems and Solutions. *Knowledge Discovery and Data Mining*, p. 73–79, (1998).
- Moro, Sérgio; Laureano, M. S. Raul e Cortez, Paulo. (2012). Enhancing bank direct marketing through data mining. *CAI*, p. 1–9. European Marketing Academy, (2012).
- Nobibon, Fabrice Talla; Leus, Roel e Spieksma, Frits C.R. (2011). Optimization models for targeted offers in direct marketing: Exact and heuristic algorithms. *European Journal of Operational Research*, v. 210, n. 3, p. 670 – 683.
- Olson, David L. e Chae, Bongsug(Kevin). (2012). Direct marketing decision support through predictive customer response modeling. *Decision Support Systems*, v. 54, n. 1, p. 443 – 451.
- Papadimitriou, C. H. e Steiglitz, K. (1998). *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications, Inc., New York.
- Peeters, Marc e Degraeve, Zeger. (2006). Branch-and-price algorithms for the dual bin packing and maximum cardinality bin packing problem. *European Journal of Operational Research*, v. 170, n. 2, p. 416 – 439.
- Praag, N. Van. (2010). Optimization of promotion campaigns using tabu search. Master's thesis, Faculty of Business and Economics, KULeuven.
- Seret, Alex; Verbraken, Thomas; Versailles, Sébastien e Baesens, Bart. (2012). A new som-based method for profile generation: Theory and an application in direct marketing. *European Journal of Operational Research*, v. 220, n. 1, p. 199 – 209.
- Souza, M.J.F.; Coelho, I.M.; Ribas, S.; Santos, H.G. e Merschmann, L.H.C. (2010). A hybrid heuristic algorithm for the open-pit-mining operational planning problem. *European Journal of Operational Research*, v. 207, n. 2, p. 1041–1051.