

Metaheurísticas aplicadas à Geração de Carrossel no Sistema Brasileiro de TV Digital

Bruno Jefferson de S.Pessoa
LAVID/DI
Universidade Federal da Paraíba
CCEN - Cidade Universitária
+55 83 3216 7093
bruno@lavid.ufpb.br

Guido Lemos de Souza Filho
LAVID/DI
Universidade Federal da Paraíba
CCEN - Cidade Universitária
+55 83 3216 7093
guido@lavid.ufpb.br

Lucídio dos Anjos F. Cabral
Departamento de Estatística
Universidade Federal da Paraíba
CCEN – Cidade Universitária
+55-83 3216-7075
lucidio@de.ufpb.br

ABSTRACT

In Digital TV environments, interactive applications are transmitted via broadcast along with audio and video streams in a transmission mechanism that reminds the cyclical turning of a "carousel". While defining the allocation method of this applications in data streams, data broadcast protocols do not consider the applications' demands as for the download time. Furthermore, applications with strong time restrictions will be damaged, since they do not have priority during the allocation of the applications in the data streams, and they will have a wait time that would represent exorbitant fines according to the current business model. To solve this problem, it were proposed a business model that defines rules to establish the priority of each application and, principally, a scheduling algorithm, based on the GRASP and VND metaheuristics, which aimed to minimize the delay in the execution of applications in a computational reasonable time.

Categories and Subject Descriptors

H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems; C.2.2 [Computer-Communication Networks]: Network Protocols; G.2.1 [Discrete Mathematics]: Combinatorics – *combinatorial algorithms*; I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search – *heuristic methods*.

General Terms

Algorithms, Experimentation.

Keywords

Digital TV, DSM-CC, Carousel, Business Model, Metaheuristics, Combinatorial Optimization.

1. INTRODUÇÃO

A interatividade, em um Sistema de TV Digital, é a principal responsável pela mudança na forma com a qual os telespectadores assistem televisão. Por meio dela, eles poderão interagir diretamente com a programação das emissoras de TV ao enviarem informações através de um canal de retorno, o qual pode ser a telefonia convencional ou mesmo a transmissão terrestre.

Para dar suporte à interatividade, os padrões de TV Digital existentes desenvolveram uma camada de software capaz de abstrair os detalhes de hardware dos terminais de acesso, proporcionando aos programadores de aplicações de TV Digital ou aplicações interativas (como são chamadas as aplicações que serão executadas nos *set-top boxes* ou terminais de acesso) uma

independência quanto as suas especificidades. Com isso, uma vez implementadas, as aplicações serão executadas em terminais de acesso de diferentes fabricantes sem necessidade de adaptação. A camada de software referida, denominada *middleware*, é responsável por todo o tratamento das aplicações citadas, desde sua recepção até o controle de seu ciclo de vida. Os principais middlewares de TV Digital existentes são o MHP [5] (europeu), o ACAP [1] (americano), o ARIB (japonês) e o GINGA [11][12] (brasileiro).

As aplicações de TV Digital, também conhecidas como *Xlets* em um *middleware* procedural, são transmitidas via *broadcast* juntamente com os fluxos de áudio e vídeo, uma após a outra, de uma forma que lembra um "carrossel de dados", visto que há a retransmissão das aplicações logo após a transmissão da última aplicação no carrossel.

A retransmissão das aplicações é de todo imprescindível, uma vez que nada se pode garantir quanto à recepção das aplicações, isto é, o telespectador pode estar com o *set-top box* desligado no momento da primeira transmissão de uma aplicação, fazendo-se indispensável o uso das retransmissões para que este tenha condições de recepção-a a posteriori.

Embora indispensável, o modelo de transmissão mencionado apresenta problemas que conflitam com o modelo de negócios utilizado no sistema de televisão atual. Como as aplicações estão organizadas em forma de uma fila circular, para serem executadas cada uma delas terá que esperar o tratamento da aplicação em sua frente, com um atraso máximo para recepção igual ao período do carrossel, ou seja, o período de tempo necessário para a transmissão de todas as aplicações, respeitando-se a taxa reservada para a transmissão de dados e considerando que haja sincronização instantânea nos terminais de acesso. Hoje, atrasos como o elucidado, usuais em ambientes de TV Digital, representariam multas exorbitantes segundo o modelo de negócios atual, e praticamente inviabilizariam a transmissão em carrossel.

Sendo assim, este trabalho apresenta um modelo matemático e um algoritmo que, baseados em um modelo de negócio próprio, objetivam minimizar o atraso na execução de suas aplicações em um tempo computacional razoável. Para isso, foi utilizada uma metaheurística híbrida, composta das metaheurísticas GRASP e VND [2] [6][13][15].

2. SISTEMAS DE TV DIGITAL

Um sistema de TV Digital pode ser definido como um conjunto de especificações que visam a transmissão e recepção de sinais digitais específicos ao ambiente de TV Digital [11]. Basicamente,

ele é composto de uma estação transmissora, um meio físico sobre o qual o sinal de vídeo é transmitido, e um receptor desses sinais, o qual é popularmente conhecido como *set-top box* ou terminal de acesso, conforme mencionado na seção anterior.

Com o objetivo de assegurar a interoperabilidade entre os elementos que compõem um Sistema de TV Digital, foram criados padrões, cujas especificações regulamentam todo o processo de captura, compressão, modulação e transmissão dos sinais de vídeo, bem como todas as interfaces físicas entre os equipamentos envolvidos no processo. Dentre os existentes podem-se destacar o Europeu (DVB – *Digital Video Broadcasting*), o Americano (ATSC – *Advanced Television Systems Committee*), e o Japonês (ISDB – *Integrated Services Digital Broadcasting*) [12][14].

Por ser o responsável pela transmissão das aplicações interativas, o Gerador de Carrossel, um dos elementos básicos em um Sistema de TV Digital, adquire notável importância para o presente trabalho, sendo, portanto, o principal destino das soluções propostas nas seções posteriores. Em geral, sua implementação segue o DSM-CC (*Digital Storage Media - Command and Control*), protocolo utilizado para a transmissão de dados multiplexados com conteúdo audiovisual na maioria dos padrões de TV Digital e definido na parte 6 do padrão MPEG-2 (extensões para DSM-CC) [10].

2.1 Sistema Brasileiro de TV Digital

No ano de 2005, o governo brasileiro instituiu o projeto SBTVD (Sistema Brasileiro de TV Digital), do qual participaram milhares de pesquisadores de várias universidades e centros de pesquisa. As atividades do SBTVD abordaram todas as partes que compõem um sistema de TV Digital, além do desenvolvimento de inovações que objetivaram ajustar a implantação da TV Digital ao contexto social brasileiro.

Devido a particularidades que são compartilhadas pela maioria dos países em desenvolvimento, o governo determinou alguns requisitos básicos para o Sistema Brasileiro de TV Digital. Entre eles tem-se: baixo custo e robustez na recepção, flexibilidade e capacidade de evolução, interatividade e novos serviços. Este visa promover a inclusão digital e é tratado como um requisito fundamental.

Percebe-se claramente que para alcançar suas pretensões e cumprir com os requisitos estabelecidos, o governo preconizará custos baixos no preço final dos terminais de acesso, a fim de que eles possam ser acessíveis à população desde a classe menos favorecida. Logo, os *set-top boxes* terão seus recursos computacionais limitados e um cenário propício à otimização de sistemas será estabelecido.

3. APRESENTAÇÃO DO PROBLEMA

Com a digitalização das informações transmitidas pelas emissoras de televisão, a largura de banda de 6 Mhz, concedida às TVs abertas pelo governo brasileiro, poderá transmitir aproximadamente 20 Mbps de áudio, vídeo e dados [17]. Após a análise de Fluxos de Transporte (protocolo de comunicação para áudio, vídeo e dados frequentemente utilizado em Sistemas de TV Digital) das principais emissoras do país, as quais seguem o modelo de transmissão adotado pelo SBTVD, foi constatado que no mínimo 14 Mbps será reservado para a transmissão de vídeo no padrão H.264, também conhecido como MPEG-4 Parte 10 ou AVC (*Advanced Video Coding*), e definido como padrão de

compressão no Sistema de TV Digital nacional. Logo, a taxa máxima pra transmissão de dados será limitada a 6 Mbps.

Tanto as aplicações procedurais (*Xlets*) quanto as aplicações declarativas possuem tamanho irrisório ao compará-las com a taxa limite mencionada acima. Entretanto, não apenas de código executável elas são formadas. Frequentemente, arquivos de áudio, vídeo, imagens e até mesmo de banco de dados são necessários a uma enorme gama de aplicações, fazendo com que o tamanho do carrossel aumente consideravelmente. Em consequência disso, a tendência é que o carrossel possua um tamanho bem maior que taxa reservada para a transmissão de dados.

Outro aspecto importante a ser ressaltado diz respeito à capacidade de armazenamento dos *set-top boxes*. Um dos requisitos do governo brasileiro que vem guiando a digitalização de seu sistema televisivo é que o resultado dessa transformação sirva como ferramenta de inclusão digital. Essa preocupação se justifica pelo fato de apenas 32,1 milhões de brasileiros possuírem acesso à internet, o que representa 21% da população, enquanto que 91% dos domicílios no Brasil possuem televisão [3]. Por conseguinte, quanto mais recursos que possam encarecer o preço final dos *set-top boxes* puderem ser removidos, mais pessoas terão condições de comprá-los e maior será o alcance da inclusão digital tão preconizada pelo governo.

Conclui-se, então, que pelo menos durante a transição da TV analógica para a digital, não existirá espaço de armazenamento reservado para as aplicações oriundas do carrossel.

Diante das limitações impostas, infere-se que essas aplicações terão um certo atraso para serem executadas, visto que terão que esperar o tratamento de todas as aplicações a sua frente, de acordo com a seqüência com a qual elas giram no carrossel, para que possam enfim ser recepcionadas. A ilustração abaixo mostra essa situação com maior clareza:

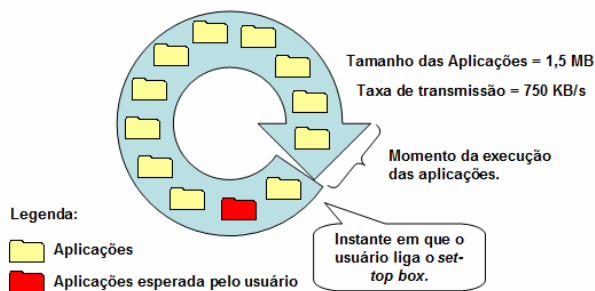


Figura 1. Representação da forma de transmissão em carrossel.

A figura anterior exibe uma “foto” de uma transmissão hipotética em carrossel no instante em que o usuário liga o *set-top box*. No cenário hipotético ilustrado, a taxa reservada para a transmissão de dados é de 750 KB/s, e o carrossel possui onze aplicações com 1,5 MB de tamanho cada uma. Repare que no instante em que o usuário liga o terminal de acesso, a aplicação na cor vermelha, pela qual o telespectador espera, já passou pelo momento da recepção e execução das aplicações. Agora, o usuário terá que esperar a passagem de todas as aplicações, até que chegue a vez aplicação esperada. O tempo decorrido para tal seria o de 18 segundos, já que existem nove aplicações a sua frente, cada uma com tempo de transmissão de 2 segundos (tamanho / taxa).

Atrasos como esses não são tolerados em hipótese alguma no contexto televisivo, independente do modelo de negócio adotado. Um segundo de publicidade nesse veículo de comunicação tem um custo elevadíssimo, sendo assim, atrasos e quebras de contratos significam multas exorbitantes e perdas financeiras insuportáveis às emissoras. Contudo, elas não são as únicas que têm o interesse em reduzir os atrasos na execução das aplicações.

É corriqueiro aos internautas esperarem vários minutos ou até mesmo horas para a conclusão de um *download*, pois eles convivem com esse tipo de situação desde o início do desenvolvimento da Internet. Todavia, esse não é um comportamento esperado dos telespectadores, que, historicamente, não costumam ter paciência e não suportam ficar de frente pra TV durante os intervalos comerciais. Semelhantes às propagandas de hoje, as aplicações de TV Interativa também tratarão de publicidade, anúncio, *marketing* e serão mais um “incômodo” para os telespectadores impacientes. Uma pequena espera para execução de uma determinada aplicação poderá significar uma mudança de canal (situação que aborta sua execução), o desligamento do terminal de acesso, ou até mesmo a inutilização de algumas aplicações num caso mais extremo.

Em vista disso e diante da impossibilidade de eliminação dos atrasos, uma vez que é comum o tamanho de um carrossel ser muitas vezes maior que a taxa reservada para a transmissão de dados, surge a necessidade de minimizá-los em um tempo computacional razoável. Para tal, as seguintes macro propriedades devem ser compreendidas:

- i. todas as aplicações serão inicialmente “candidatas” e poderão estar presentes ou não no carrossel;
- ii. elas terão frequências diferentes de acordo com seu benefício;
- iii. a distância entre aplicações idênticas dentro do carrossel será minimizada de forma a reduzir seu atraso máximo;
- iv. a receita obtida pela emissora será resultado da subtração do valor recebido com a veiculação das aplicações e a multa paga por atraso de cada uma delas;

Percebe-se das propriedades (i) e (ii) que o problema abordado é um tipo especial do problema clássico da Mochila 0-1, adicionada a possibilidade de inserção de um mesmo objeto mais de uma vez. Tal problema clássico é considerado pela literatura como um problema NP-HARD [7]. Uma vez adicionadas, as aplicações precisam ser organizadas dentro do carrossel de forma a minimizar a distância entre as aplicações replicadas e, consequentemente, minimizar o atraso e aumentar a receita da emissora (propriedades (iii) e (iv)). Para isso, deve-se alterar a posição das aplicações até que uma configuração de carrossel com o mínimo de atraso seja encontrada.

A Figura 2 mostra como um “layout” de carrossel organizado pode reduzir o atraso. A primeira representação de carrossel da esquerda para a direita exibe a replicação das aplicações de acordo com seu benefício (problema clássico da mochila), sem a preocupação de organizar sua estrutura interna. Note que a aplicação na cor vermelha — a de maior benefício — terá que esperar a transmissão das oito aplicações no pior caso. Minimizando a distância máxima entre as aplicações idênticas, essa mesma aplicação terá que esperar apenas por outras duas, reduzindo bastante o tempo necessário para sua execução.

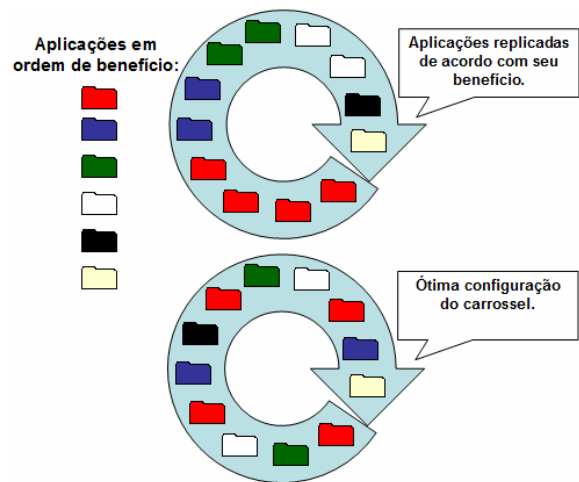


Figura 2. Otimização do atraso no carrossel.

O número de operações necessárias para encontrar a solução ótima é dado pela fórmula da permutação com repetição:

$$p_n = \frac{n!}{q_1!q_2!\dots q_n!}, \quad \text{onde } q_1, q_2 \text{ e } q_n \text{ representam a quantidade de aplicações de índice 1, 2 e n inseridas no carrossel, e n o número total de aplicações inseridas.}$$

Um algoritmo que teste todas as configurações possíveis na procura da solução ótima teria uma complexidade $\theta(n!)$, e um número de operações próximo a $n!$, visto que no problema específico em questão o numerador tende a ser um valor absurdamente maior que denominador.

Vale salientar que n não corresponde ao número de aplicações candidatas a serem inseridas no carrossel, e sim ao somatório das frequências das aplicações selecionadas.

O tamanho final do carrossel é variável e será obtido a partir do somatório citado anteriormente. Entretanto, ele terá um limite máximo que representa um valor que gera um atraso obrigatoriamente menor que o tempo de veiculação contratado pelos anunciantes.

Um problema semelhante na literatura, também considerado NP-HARD, é o denominado de “Pinwheel”. Ele é um caso especial do problema de programação de tarefas periódicas, versão discreta do problema de programação de tarefas de distâncias restritas (*Distance Constrained Tasks Scheduling*) [8], e consiste na formalização de um problema de comunicação via satélite. Imagine vários satélites podendo transmitir dados a qualquer momento a uma estação central, com a restrição de que esta apenas pode processar uma informação de cada vez. Visando evitar perda de dados, cada satélite pode enviar uma informação a qualquer instante, porém deve reenviá-la periodicamente em um intervalo de tempo conhecido. Uma vez que cada satélite tem a liberdade de definir a frequência de transmissão de seus dados, há a necessidade de programar suas transmissões, a fim de evitar colisões entre dados de satélites diferentes, conservando-se a frequência de envio de dados pré-definida de cada satélite. A busca por tal programação não pode ser solucionada em tempo polinomial, ocorrendo, inclusive, a possibilidade de não existir

solução. O “*Pinwheel*” é a formalização desse problema. Mais detalhes podem ser encontrados em [9].

A complexidade do problema abordado por este trabalho inviabiliza a busca pela solução ótima e torna necessário o uso de metaheurísticas que auxiliem na busca de um boa solução em um tempo computacional razoável. A partir delas que este trabalho se desenvolve e apresenta soluções para os problemas definidos.

4. MODELO DE NEGÓCIO PROPOSTO

Espera-se muito da interatividade na TV Digital aberta brasileira, porém há muito a ser discutido e definido. Sabe-se que revolucionará a maneira da assistir televisão e que surgirá um nova forma de publicidade e *marketing* capaz de gerar receitas no grande negócio que é a televisão. Entretanto, não há nada definido no tocante às regras que serão impostas para transformar a interatividade em negócio.

Um dos poucos consensos é que as aplicações serão veiculadas às programações das emissoras, assim como acontece hoje com as propagandas.

O modelo de negócios proposto não tem a pretensão de solucionar todos os entraves envolvidos na negociação da interatividade, tendo seu escopo reduzido ao desenvolvimento de uma solução que minimize o atraso na execução das aplicações de TV Digital transmitidas em carrossel e, conseqüentemente, aumente a receita das emissoras. Logo, ele se destina a propor regras de negócio com o intuito de especificar valores e parâmetros para a veiculação de aplicações interativas, e principalmente definir uma forma de calcular quanto se deve pagar por seus atrasos. Para isso foram criados dois cenários, A e B, que tratam de forma diferentes o valor a ser pago para a transmissão de tais aplicações.

Além dos parâmetros tempo e horário de exibição, que atualmente dominam a negociação da veiculação de propagandas, há uma inclinação, por parte das emissoras, em também negociar banda de transmissão. Dessa forma, os anunciantes pagarão valores diferenciados de acordo com os formatos de áudio e vídeo escolhidos. Por exemplo, um vídeo de alta definição, que necessita de uma maior largura de banda para ser transmitido, custará bem mais aos anunciantes se comparado com os vídeos em definição padrão.

Apesar de interessante para fluxos de áudio e vídeo, essa estratégia prejudica a negociação das aplicações interativas, pois limitam o tamanho máximo do carrossel à taxa reservada para transmissão de dados, que de acordo com a Seção 3 estará por volta de 6Mbps ou 750KB/s (transformando a unidade) no Sistema Brasileiro de TV Digital. Isso implica na rejeição de contratos com anunciantes, pois dessa forma a soma das aplicações presentes em um carrossel nunca ultrapassariam 720 KB, ficando limitadas à exibição de imagens em definição padrão e sem a utilização de recursos interessantes, como áudio, vídeo e banco de dados.

Para não fugir totalmente da tendência atual de vender *Kbytes*, foi definido o Cenário A, que embora não apresente a limitação citada no parágrafo anterior, exige que aplicações maiores paguem mais por sua veiculação.

Além do tamanho, faixas de valores foram criadas como forma de replicar e minimizar o atraso das aplicações que pagam mais por sua veiculação. Exemplificando, nos testes realizados foram utilizadas faixas de valores que variaram de R\$ 0,10 a R\$ 1,00 por

KByte contratado. Sendo assim, uma aplicação com 1 MB de tamanho e que escolheu a faixa 5 para veiculação (faixa 1 = R\$ 0,10/KB, faixa 2 = R\$ 0,20/KB, etc), terá que pagar R\$ 500,00 / s durante o período em que participará da transmissão em carrossel. Vale salientar que o período de manipulação da aplicação é diferente do período de participação no carrossel. Contratar um período de 60 segundos ($500 * 60 = R\$ 30.000$ a ser pago no exemplo anterior) não quer dizer que o usuário terá todo esse tempo para manipulá-la. Significa apenas que uma determinada aplicação será transmitida para os telespectadores, junto com outras aplicações inclusive, por esse período de tempo. O tempo de manipulação (t_m), nesse caso, seria a soma do instante (t_0) da execução da aplicação com seu atraso (at), subtraído do período contratado. Isto é:

$$t_m = 60 - (t_0 + at), \text{ onde } t_0 \text{ estaria no intervalo } [0, 60].$$

O valor da multa a ser paga pelo atraso seria calculado a partir do valor pago pela veiculação, levando em consideração um atraso máximo tolerado pelos anunciantes devido à natureza da transmissão em carrossel em ambientes de TV Digital, a qual difere do modelo de negócios atual como foi explicado anteriormente. Após tal atraso, visando evitar longos períodos de espera para a execução de aplicações interativas, a multa por segundo de atraso seria aumentada gradativamente de acordo com cláusulas contratuais acordadas entre as emissoras e os anunciantes.

O cenário B compartilha todas as características do Cenário A, com exceção do valor a ser pago para veiculação de uma aplicação. No último há uma vinculação entre o tamanho da aplicação e o valor total pago, cujo cálculo se dá em função do preço do *Kbyte*. Em conseqüência disso, não existe a possibilidade de aplicações grandes pagarem pequenas quantias, mesmo que seus anunciantes queiram arcar com longos períodos de atraso. No cenário B também são definidas faixas de valores, porém estas não tem vinculação nenhuma com o tamanho das aplicações, permitindo uma maior flexibilidade na negociação entre emissoras e anunciantes.

5. METODOLOGIA

Com o objetivo de implementar algoritmos capazes de otimizar a geração de carrossel no Sistema Brasileiro de TV Digital, fez-se necessário a adoção de um procedimento metodológico composto pela elaboração das etapas descritas nos tópicos posteriores.

5.1 Modelagem Matemática

O modelo matemático proposto baseia-se na idéia de reduzir o tempo máximo entre dois downloads subseqüentes de um mesmo *Xlet* (com finalidade didática, as aplicações interativas, procedurais ou declarativas, são todas denominadas de *Xlets* nesta seção). Em geral, a solução produzida tende a distribuir as possíveis cópias de cada *Xlet* de maneira uniforme ao longo da seqüência de *Xlets* do carrossel.

$$\text{Max} \sum_{i \in X} p_i * z_i * TV - \sum_{i \in X} w_i * (p_i * 2) + y_i * p_i$$

s.t :

$$\sum_{j \in T} \sum_{i \in X} t_i * x_{ij} \leq TMax, \quad (1)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad \forall j \in [1...q] \quad (2)$$

$$x_{ij} = x_{(i+n)j}, \quad \forall i \in X, \forall j \in [1...q] \quad (3)$$

$$z_i \leq \sum_{j \in T} x_{ij}, \quad \forall i \in X \quad (4)$$

$$y_i \geq (D_i / R) - [AT + (t_i / R)], \quad \forall i \in X \quad (5)$$

$$w_i \geq (D_i / R) - y_i, \quad \forall i \in X \quad (6)$$

$$d_{kj}^i \geq \sum_{s=1}^n \sum_{l=k}^{j-1} t_s * x_{sl} - M(1 - x_{ik}) - M(1 - x_{ij}) - \sum_{l=k+1}^{j-1} M * x_{il}, \quad \forall i \in X, \forall k, j \in [1, \dots, q], k < j \quad (7)$$

$$D_i \geq d_{kj}^i, \quad \forall k, j \in [1, \dots, q], k < j \quad (8)$$

$$y_i, w_i \geq 0, \quad \forall i \in X \quad (9)$$

$$z_i, x_{ij} \in \{0,1\} \quad \forall i \in X, \forall j \in [1...q] \quad (10)$$

$$D_i, d_{kj}^i \geq 0, \quad \forall i \in X, \forall k, j \in [1...q] \quad (11)$$

O modelo apresentado é formulado com base na abstração de uma matriz de q colunas (posição de cada *Xlet* no carrossel) e $2n$ linhas (cada uma representa um *Xlet*), em que o conjunto de *Xlets* é denotado por X e $n = |X|$. O carrossel é representado duas vezes na matriz ($2n$ linhas) a fim de possibilitar o cálculo das distâncias entre aplicações idênticas e subsequentes, levando em conta o giro do carrossel. O número máximo de colunas q é calculado a partir da divisão de $TMax$ pela menor aplicação pertencente a X , onde $TMax = TV * R$. O tamanho máximo do carrossel é dado por $TMax$, o qual é obtido ao multiplicar o tempo de exibição das aplicações pela taxa reservada para a transmissão de dados.

O primeiro termo da função objetivo representa o lucro obtido com a adição de *Xlets* no carrossel (denota o lucro logrado com a contratação das aplicações) e o segundo as multas aplicadas devido aos atrasos em sua execução.

A restrição (1) impõe um limite máximo ao tamanho do carrossel, enquanto que a (2) garante que apenas um único *Xlet* é alocado a cada posição do carrossel. A restrição (3) força a representação dupla do carrossel na matriz, e a (4) sinaliza quais *Xlets* serão selecionados a compor o carrossel. As restrições (5) e (6) calculam as parcelas do atraso das aplicações a serem multadas com as diferentes penalidades na função objetivo. Já as restrições (7) e (8) capturam o atraso máximo (em *KBytes*) D_i para cada

Xlet. Finalmente, as restrições de integralidade e não-negatividade são apresentadas em (9), (10) e (11).

5.2 GVND

O algoritmo em questão é composto das metaheurísticas GRASP e VND, cuja integração é detalhada nas seções seguintes.

5.2.1 GRASP

Nesta seção é descrita a metaheurística GRASP – *Greedy Randomized Adaptive Search Procedure* – proposta para a otimização do Gerador de Carrossel de acordo com o modelo de negócios apresentado anteriormente.

Um GRASP é um procedimento iterativo que combina várias propriedades favoráveis de outras heurísticas [15]. Mais especificamente, cada iteração do GRASP consiste de dois estágios: uma fase de construção da solução e uma fase de busca local. Em cada iteração, uma solução é encontrada. A melhor solução obtida dentre todas as iterações é considerada a solução final. O pseudocódigo abaixo descreve a metaheurística GRASP.

Algoritmo GRASP ($f(\cdot)$, GRASPmax, s)

1. $f^* = \infty$;
2. **para** $Iter = 1$ **até** GRASPmax **faça**
3. {
4. *Fase-Construcao*(s);
5. *Fase-BuscaLocal*($f(\cdot)$, s);
6. **se** ($f(s) > f^*$) **então** {
7. $s^* = s$;
8. $f^* = f(s)$;
9. }
10. }
11. Retorne s^* ;

Fim GRASP

Na fase de construção de uma solução, inicia-se um conjunto vazio, que iterativamente recebe um elemento, até formar uma solução viável. Em cada iteração da fase de construção, dois aspectos são analisados: a aleatoriedade e a adaptação. A aleatoriedade se deve ao fato de que o próximo elemento a ser escolhido para compor a solução não será necessariamente o melhor, segundo o critério guloso utilizado. Na verdade, a escolha é feita de forma aleatória, a partir de uma lista, denominada Lista Restrita de Candidatos (LRC), que contém os β melhores elementos candidatos segundo o critério guloso especificado. Um algoritmo de busca local é utilizado nesta segunda fase para, sucessivamente, substituir a solução atual por uma solução melhor, encontrada na vizinhança da solução atual. O algoritmo termina quando nenhuma solução melhor é encontrada na vizinhança da solução atual ou após algum outro critério de parada ter sido atingido.

5.2.1.1 Construção

A fase de construção tem seu início com uma solução S vazia, ou seja, nenhuma aplicação interativa foi selecionada para compor o carrossel. A cada iteração uma aplicação é escolhida, de forma aleatória, a partir da lista restrita de candidatos (LRC).

A lista LRC corresponde às aplicações que possuem um melhor “benefício” em analogia ao problema clássico da Mochila 0-1. Essa lista é obtida com o auxílio de uma função gulosa, denotada por $g(x)$, que estima o benefício de seleção de cada aplicação x candidata a compor o carrossel.

$$g(x) = vp(x) * tv(x) / tam(x) + esp(x), \quad (2)$$

onde:

- $vp(x)$: corresponde ao valor pago pela veiculação da aplicação x ;
- $tv(x)$: corresponde ao tempo de veiculação da aplicação x ;
- $tam(x)$: corresponde ao tamanho da aplicação x ;
- $esp(x)$: corresponde ao espaço ocupado no carrossel pela aplicação x em cada iteração;

As melhores aplicações candidatas a formar o carrossel serão aquelas que satisfizerem a condição abaixo:

$$g(x) \geq g_{max} - \alpha(g_{max} - g_{min}), \quad (3)$$

onde $g_{min} = \min\{g(x) \mid x \text{ é uma aplicação candidata}\}$, $g_{max} = \max\{g(x) \mid x \text{ é uma aplicação candidata}\}$ e $\alpha \in [0,1]$.

O parâmetro α , que determina o tamanho da lista de candidatos restrita, é basicamente o único parâmetro a ser ajustado na implementação de um procedimento GRASP. Valores de α que limitam a lista restrita de candidatos, isto é, valores que levam a uma escolha próxima da gulosa, implicam em soluções finais de qualidade muito próxima àquela obtida de forma puramente gulosa, com um baixo esforço computacional. Por outro lado, deixam a desejar na diversidade das soluções construídas. Já uma escolha de α próxima da seleção puramente aleatória leva a uma grande diversidade de soluções construídas mas, por outro lado, muitas das soluções construídas são de qualidade inferior, tornando mais lento o processo de busca local [16]. No problema em questão, testes demonstraram que os melhores resultados foram encontrados com α próximo a 0,5.

Após a obtenção de *LRC* e a escolha aleatória de uma aplicação $x \in LRC$, há a adição desta no carrossel, fazendo-se necessária a atualização das listas de aplicações selecionadas e não-selecionadas.

A fase de construção é encerrada quando o tamanho do carrossel atinge um valor limite, que corresponde a um múltiplo da soma das aplicações candidatas. Valores próximos a duas ou três vezes a soma das aplicações candidatas alcançaram os melhores resultados.

5.2.1.2 Busca Local

A fase de Busca Local utilizou o VND como estratégia de refinamento da solução, o que a transformou no ponto de hibridização do algoritmo GVND.

5.2.2 VND

No método VND (*Variable Neighborhood Descent*), proposto por Mladenovic e Hansen [13], explora-se o espaço de soluções através de trocas sistemáticas de estruturas de vizinhança. Uma nova solução é aceita somente quando for de melhora em relação à solução corrente e retorna-se à primeira estrutura quando isto acontece [4]. Quando não há mais melhora da solução, troca-se a estrutura de vizinhança e o algoritmo segue até que todas as vizinhanças sejam exploradas. Abaixo encontra-se seu pseudocódigo:

Algoritmo VND($f(\cdot)$, $N(\cdot)$, r , s)

```

1. Seja  $r$  o número de estruturas diferentes de vizinhança;
2.  $s \leftarrow s_0$ ; // Solução corrente
3.  $k \leftarrow 1$ ; // Tipo de estrutura de vizinhança corrente
4. enquanto ( $k \leq r$ ) faça {
5.     Encontre o melhor vizinho  $s \in N^{(k)}(s)$ ;
6.     se ( $f(s') > f(s)$ ) então {
7.          $s \leftarrow s'$ ;
8.          $k \leftarrow 1$ ;
9.     } senão {
10.         $k \leftarrow k + 1$ ;
11.    }
12. }
13. Retorne  $s$ ;

```

Fim VND;

Foram definidas três estruturas de vizinhança, N1, N2 e N3, definidas a partir dos seguintes movimentos, respectivamente:

- permutar a posição dos elementos selecionados no carrossel um a um (complexidade: $\theta(n^2)$);
- excluir uma aplicação em uma posição específica do carrossel (complexidade: $\theta(1)$);
- inserir cada aplicação, selecionada ou não, em todas as posições do carrossel (complexidade: $\theta(n^2)$);

A estrutura de vizinhança N1 foi escolhida como a primeira vizinhança a ser explorada por ser aquela que tem mais influência no melhoramento da solução corrente. A segunda estrutura de vizinhança tenta reduzir o carrossel ao seu tamanho mínimo como forma de encontrar uma boa solução. É nela que a geração de vizinhos se dá com uma menor complexidade, sendo, portanto, um fator importante na redução do custo computacional. Por último, temos N3, que além de aumentar o carrossel como meio de encontrar uma boa solução, tenta reduzir a distância entre as aplicações idênticas ao realizar inserções em todas as posições do carrossel.

6. RESULTADOS

A base de comparação para os resultados obtidos foi definida de acordo com a solução adotada na implementação dos Geradores de Carrossel atuais, que apenas transmitem as aplicações uma após a outra sem a preocupação em reduzir atrasos. Essa solução foi avaliada de acordo com a modelagem matemática apresentada na "Seção 4" e confrontada com os resultados obtidos a partir do algoritmo GVND para os dois cenários do modelo de negócios proposto. A diferença entre parâmetros específicos da área de aplicação do problema em consideração, e os parâmetros geralmente envolvidos com o problema "Pinwheel" (vide Seção 3), aplicado principalmente a Sistemas de Tempo Real, dificultou uma possível comparação de resultados.

A implementação do algoritmo proposto utilizou a linguagem C++, e os experimentos foram todos realizados em um computador com sistema operacional Linux 32bits, kernel 2.6.24-16, distribuição Ubuntu 8.04, processador Core 2 Duo T5250 de 1.5 Ghz e memória RAM de 1 GB.

6.1 Descrição das Instâncias

Uma instância é formada por um conjunto de aplicações inicialmente candidatas a fazerem parte do grupo seletor de aplicações que comporão o carrossel. Cada aplicação possui os seguintes atributos: identificador (ID), tamanho, faixa, valor pago por segundo de veiculação, tempo total de veiculação, atraso máximo tolerado.

Foram gerados três grupos de instâncias, um com dez aplicações e os outros com 15 e 20, respectivamente, para cada cenário. A produção de instâncias se deu de forma aleatória, com os atributos das aplicações obedecendo a determinados limites.

O tamanho das aplicações foi gerado a partir de um levantamento prévio de aplicações reais desenvolvidas no LAVID (Laboratório de Aplicações de Vídeo Digital), laboratório especializado no desenvolvimento de soluções para TV Digital, onde o presente trabalho foi originado. Em tal levantamento foi observado que as aplicações interativas variavam de 250 KB a 2,5 MB, que serviram de limites para a geração aleatória das instâncias.

Para o Cenário A as faixas de veiculação variaram de 1 a 10, com a mesma semântica apresentada na “Seção 4. Modelo de Negócio Proposto” (Faixa 1 = R\$ 0.10, Faixa 2 = R\$ 0.20, e assim sucessivamente); o valor pago por segundo de veiculação foi deduzido multiplicando-se o tamanho da aplicação com o valor referente a sua faixa de veiculação; o tempo total de veiculação foi fixado em 120 segundos para todas aplicações; e pra finalizar, o atraso máximo tolerado foi definido como 10% do tempo total de veiculação, ou seja, 12 segundos. Após isso as multas por segundo de atraso são cobradas em dobro com relação ao valor pago por segundo de veiculação.

Já para o cenário B foram definidas dez faixas que representam as quantias a serem pagas por segundo para veiculação de aplicações independentemente de seus tamanhos. As quantias para cada faixa, do menor para o maior valor, são em reais (R\$): 25, 50, 100, 200, 400, 800, 1200, 1600, 2000, 2500; o tempo total de veiculação foi fixado em 180 segundos para todas aplicações, simulando o período de um intervalo comercial; o atraso máximo tolerado continuou sendo os mesmos 12 segundos do cenário A, assim como o cálculo das multas.

O único parâmetro relacionado ao modelo de negócios proposto que não faz parte do conjunto de propriedades de uma aplicação é a taxa reservada para transmissão de dados, a qual foi estabelecida como 720 KB/s — valor dentro do limite máximo de 6 Mbps ou 750 KB/s do Sistema Brasileiro de TV Digital, para ambos os cenários.

6.2 GVND versus Solução Atual

Por não levar em conta aspectos importantes como o tamanho, a prioridade (faixa) e o atraso máximo das aplicações, a solução atual implementada nos Geradores de Carrossel acaba sofrendo bastante com as multas, o que reduz consideravelmente a diferença receita-multa, representada como “lucro” nas ilustrações desta seção.

Para cada instância foram realizadas 10 execuções, o que resultou em um total de 300 execuções, 150 para cada cenário do modelo de negócios. Os valores utilizados nesta seção representam a média dos encontrados nas execuções mencionadas.

As tabelas abaixo apresentam, para cada cenário do modelo de negócios, os valores da função de avaliação para a Solução Atual,

as médias dos valores obtidos nas execuções do algoritmo GVND e o seu tempo de execução médio. Além disso, são apresentados os percentuais de ganho do GVND em relação à Solução Atual.

Tabela1. Comparação entre a Solução Atual e o GVND para o Cenário A.

Instância	Solução Atual (R\$)	GVND (R\$)	Ganho (%)	Tempo (s)
1	494.213,63	513.041,36	3,81	0,5
2	847.817,20	856.288,85	1,00	0,4
3	665.352,83	665.352,83	0,00	0,6
4	686.040,24	695.174,89	1,33	0,8
5	564.443,99	592.469,45	4,97	0,8
6	836.188,44	858.764,58	2,70	4,9
7	802.245,38	917.090,16	14,32	4,2
8	852.626,47	936.838,61	9,88	5,1
9	749.796,63	795.508,55	6,10	6,2
10	858.005,81	929.091,26	8,28	2,2
11	974.174,94	1.086.103,8	11,49	23,4
12	907.228,55	1.102.782,5	21,56	31,0
13	840.132,76	1.115.208,9	32,74	20,5
14	842.525,36	1.132.070,5	34,37	28,8
15	778.131,17	1.023.192,7	31,49	27,7

Tabela2. Comparação entre a Solução Atual e o GVND para o Cenário A.

Instância	Solução Atual (R\$)	GVND (R\$)	Ganho (%)	Tempo (s)
1	1.366.756,61	1.444.388,39	5,68	1,6
2	902.172,71	942.499,83	4,47	0,6
3	434.001,04	463.990,51	6,91	1,4
4	1.193.405,92	1.211.545,69	1,52	2,4
5	1.251.874,58	1.417.998,33	13,27	0,6
6	1.526.063,89	1.823.341,14	19,48	5,1
7	1.109.602,25	1.237.983,23	11,57	3,0
8	2.388.033,33	2.570.717,88	7,65	9,6
9	2.275.026,54	2.467.266,28	8,45	11,8
10	1.592.715,83	1.816.651,68	14,06	3,8
11	2.251.665,61	2.734.422,72	21,44	27,8
12	1.013.190,90	1.275.506,02	25,89	31,4
13	1.125.306,10	1.541.331,76	36,97	55,1
14	1.175.151,46	1.458.245,45	24,09	29,4
15	1.642.403,28	1.974.004,50	20,19	63,4

Como visto anteriormente, a principal diferença entre os Cenários A e B está na forma como os anunciantes pagarão pela veiculação de suas aplicações. No cenário A ocorre uma vinculação do valor a ser pago com o tamanho das aplicações, os quais são os principais parâmetros que definem a replicação e a prioridade das aplicações quanto à minimização de atrasos. Tal vinculação gera uma dependência entre eles de tal forma que o algoritmo GVND

encontra pouca diversidade entre as aplicações de uma instância, o que prejudica a busca por boas soluções. É como se um parâmetro anulasse o outro, fazendo com que as aplicações fossem relativamente iguais. Como é sabido, quanto mais as aplicações são semelhantes, menos necessidade há de replicá-las ou excluí-las, e assim menos trabalho é realizado pelos algoritmos.

Vale salientar, ainda, que o tempo de veiculação no Cenário B é de 180 segundos enquanto que o do Cenário A é de apenas 120. Isso quer dizer que reduzir um segundo de atraso em 180 segundos de veiculação gera um ganho percentual bem menor que a redução do mesmo segundo em 120 segundos de veiculação. Caso fosse reduzido tal tempo de veiculação, os valores alcançados no Cenário B seriam bem superiores àqueles do outro cenário.

7. CONSIDERAÇÕES FINAIS

Independentemente do modelo de negócios a ser adotado para veiculação das aplicações de TV Digital, sempre existirá a necessidade de reduzir-se ao máximo o tempo em que elas estarão disponíveis para o usuário. O tempo é um parâmetro muito valorizado no ambiente televisivo, e, cientes disso, os anunciantes serão totalmente resistentes a investir grandes cifras em aplicações que demoram a ser executadas.

Outros que serão afetados pelos atrasos referidos são os telespectadores. Historicamente impacientes, não suportarão esperar por aplicações em detrimento da programação das emissoras. A aversão criada aos intervalos comerciais dificilmente será revertida a curto prazo, assim, minimizar os atrasos se tornará uma preocupação essencial ao modelo publicitário adotado com a chegada da TV Digital.

Adiantando-se a todos esses problemas, o presente trabalho, além de sugerir um modelo de negócios para a veiculação das aplicações interativas, propõe um modelo matemático para um problema novo na literatura, e um algoritmo capaz de solucioná-lo em um tempo computacional razoável.

Por focar na redução de atrasos, preocupação presente em qualquer modelo publicitário que vise anunciar através de aplicações interativas, a solução proposta neste trabalho de certa forma se torna independente de um modelo de negócios em particular, o que a habilita para qualquer realidade futura.

8. REFERENCES

[1] Advanced Common Application Platform (ACAP). Disponível em: <http://www.atsc.org/standards/a_101.pdf>. Acesso em Novembro de 2007.

[2] AIEX, R. M., RESENDE, M. G. C. "Parallel strategies for GRASP with path-relinking", *Metaheuristics: Progress as Real Problem Solvers*, (T. Ibaraki, K. Nonobe and M. Yagiura, editores), pp. 301-331, Springer, 2005.

[3] BATISTA, C. E. C. F., ARAÚJO, T. M. U. de, OMAIA, D., ANJOS, T. C. dos, CASTRO, G. M. L. de, BRASILEIRO, F. V., SOUZA FILHO, G. L. de. "TVGrid: A Grid Architecture to use the idle resources on a Digital TV network". *Seventh IEEE International Symposium on Cluster Computing and the Grid* — CCGrid, 2007.

[4] CAVALCANTI, F. M. C., SOUZA, M. J. F., SOUZA, F. S. H., COELHO, V. S. "Uma Metodologia Heurística baseada em GRASP, VND E VNS para a resolução do problema de

dimensionamento de Redes IP". *XXXVI Simpósio Brasileiro de Pesquisa Operacional*, 2004.

[5] ETSI. "ETSI. TS 102 819: Globally Executable MHP (GEM)." ETSI Standard, maio, 2004. Disponível em: <http://webapp.etsi.org/workprogram/Report_WorkItem.asp?WKI_ID=19737>. Acesso em Novembro de 2007.

[6] FESTA, P., RESENDE, M. G. C. "GRASP: an annotated bibliography", *Essays and Surveys on Metaheuristics* (C.C. Ribeiro e P. Hansen, editores), pp. 325-367, Kluwer Academic Publishers, 2002.

[7] GAREY, M. R., JOHNSON, D. S. (1979). "Computers and Intractability: A Guide to the Theory of NP-Completeness". W.H. Freeman and Co., San Francisco.

[8] HAN, C.-C., LIN, K.J., HOU, C.-J., "Distance-constrained Scheduling and Its Application to Real-Time Systems", *IEEE Transactions on Computers* 45, 814-826, 1996.

[9] HOLTE, R., MOK, A., ROSIER, L., TULCHINSKY, I., VARVEL, D. "The Pinwheel: A real-time scheduling problem", *IEEE*, 1989.

[10] ISO/IEC TR 13818-6. Information technology — Generic coding of moving pictures and associated audio information — Part 6: Extensions for DSM-CC, 1998.

[11] LEITE, L. E. C., BATISTA, C. E. C. F., SOUZA FILHO, G.L. de, KULESZA, R., ALVES, L. G. P., BRESSAN, G., RODRIGUES, R. F., SOARES, L. F. G. "FlexTV – Uma Proposta de Arquitetura de Middleware para o Sistema Brasileiro de TV Digital". *Revista de Engenharia de Computação e Sistemas Digitais*, v. 2, p. 29-50, 2005.

[12] LEITE, L. E. C., SOUZA FILHO, G. L. de. "Recomendações para o Modelo de Referência: MIDDLEWARE (RFP04/2004) FlexTV". Disponível em: <<http://www.tvdigital.inf.br/pdfs/relatorios/Middleware-UFPB.pdf>>. Acesso em Novembro de 2007.

[13] MLADENOVIC, N., HANSEN, P. "Variable Neighborhood Search", *Computers and Operations Research*, Vol. 24, No. 1 I. pp. 1097-1100, 1997.

[14] PESSOA, B. J., SALVADOR, E. M., BEZERRA, E. P. "Implementação do componente para controle de decodificação e reprodução de áudio e vídeo em ambientes de TV Digital". *Anais do IV Fórum de Oportunidades sobre Televisão Digital Interativa*, 2006, Poços de Caldas.

[15] RESENDE, M. G. C. "Greedy randomized adaptive search procedures (GRASP)", *Encyclopedia of Optimization*, (C. Floudas e P.M. Pardalos, editores), vol. 2, pp. 373-382, Kluwer Academic Press, 2001.

[16] SOUZA, M. J. F. "Inteligência Computacional para Otimização". Disponível em: <<http://www.decom.ufop.br/prof/marcone/Disciplinas/InteligenciaComputacional/InteligenciaComputacional.pdf>>. Acesso em Outubro de 2007.

[17] YAMADA, F. et al, "Sistema de TV Digital". In *Revista Mackenzie de Engenharia e Computação – Ano 5, Número 5*. Editora Mackenzie, São Paulo, 2004.