

—Exame de Qualificação de Mestrado—

Problemas de Disposição de Propagandas

Aluno: Mauro Roberto Costa da Silva

Orientador: Rafael Crivellari Saliba Schouery

Instituto de Computação - UNICAMP

Resumo

O valor arrecadado em propagandas na internet cresceu muito no século 21. O crescimento na exibição de anúncios na internet criou uma indústria multibilionária, sendo que um terço de todas as propagandas na internet são exibidas em *banners*. Isso levanta a questão de como devem ser organizadas as propagandas dentro de *banners* de forma a maximizar o lucro. Nesse contexto, problemas de disposição de propagandas consistem em dispor um conjunto A de propagandas em um *banner* B , onde temos N unidades de tempo, chamados *slots* e, em cada unidade de tempo, as propagandas do *slot* correspondente são exibidas em B . Neste projeto, apresentamos os problemas MAXSPACE e MINSPACE e fazemos uma revisão bibliográfica de ambos, levando em consideração algoritmos exatos, heurísticas e algoritmos de aproximação encontrados na literatura. Em seguida, apresentamos quais variantes podem ser consideradas para o projeto de novos algoritmos para os problemas abordados.

1 Introdução

Muitos sites (como Google, Yahoo! e Facebook) disponibilizam serviços de forma gratuita, enquanto apresentam propagandas aos usuários. Para esses sites, as propagandas são a principal fonte de receita. Assim é importante imaginar a melhor maneira de dispor as propagandas no espaço disponível, com o objetivo de maximizar o valor arrecadado [15].

Em *sites* como o Google, os anúncios possuem um **Custo Por Clique** (CPC), que é o valor cobrado do anunciante quando seu anúncio é clicado por um usuário. Outra forma de receita, usada em *sites* como o Facebook, é o **Custo Por Milhar** (CPM), que consiste do valor cobrado para cada mil exibições do anúncios. Também podemos citar o **Custo Por Ação** (CPA), que é o valor cobrado quando uma determinada ação é realizada sobre um anúncio. Por exemplo, quando uma loja anuncia um produto em um *site* como o Google e um usuário clica nesse anúncio e realiza uma compra no *site* do anunciante, o Google recebe um valor referente à ação de compra.

Consideramos problemas de disposição de propagandas onde desejamos dispor um conjunto A de propagandas em um retângulo B , onde temos N unidades de tempo, chamados *slots*, sendo que, em cada unidade de tempo, as propagandas do *slot* correspondente são exibidas em B . Todas as propagandas possuem a mesma largura e cada propaganda A_i possui uma altura e uma frequência, respectivamente, s_i e w_i , em que $w_i \leq N$. A altura da propaganda A_i representa o espaço que A_i ocupa em um *slot* e a frequência define o número de *slots* em que A_i deve aparecer. Como não é muito interessante para os anunciantes que a sua propaganda apareça duas vezes no mesmo *slot*, uma propaganda pode ser adicionada no máximo uma vez por *slot* [7].

Os principais problemas dessa classe são o MINS-SPACE e o MAXSPACE. No MINS-SPACE, a altura do retângulo B não é especificada e o objetivo é dispor todas as propagandas minimizando a altura de B , ou seja, desejamos minimizar $\max\{S_i : \forall i \in B\}$, em que S_i é a altura do *slot* i . Já no MAXSPACE, um limite superior S é especificado para a altura de B , dessa forma, nenhum *slot* pode exceder a altura máxima S . Consideramos nesse caso que $s_i \leq S$, para toda propaganda A_i . Uma solução viável para o MAXSPACE consiste em um subconjunto $A' \subseteq A$ tal que, para todo $A_i \in A'$, exatamente w_i cópias

de A_i foram adicionadas aos *slots*, de forma que nenhum *slot* recebeu mais de uma cópia de A_i e nenhum *slot* excedeu a altura máxima S , isto é, para todo *slot* B_j , $\sum_{A_i \in B_j} s_i \leq S$. O objetivo do MAXSPACE é maximizar o lucro, dado por $\sum_{A_i \in A'} s_i \cdot w_i$, isso é, o lucro associado à propaganda é a sua altura multiplicada pela sua frequência [7]. Outras variações para esse problema são apresentadas na Seção 4.

Considere as propagandas descritas na Tabela 1.

Tabela 1: Propagandas

| A_i | A_1 | A_2 | A_3 | A_4 | A_5 | A_6 | A_7 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| s_i | 6 | 4 | 2 | 5 | 1 | 1 | 5 |
| w_i | 3 | 2 | 1 | 2 | 1 | 1 | 1 |

Na Figura 1 são apresentados exemplos de soluções para o MINSPACE com as propagandas da Tabela 1.

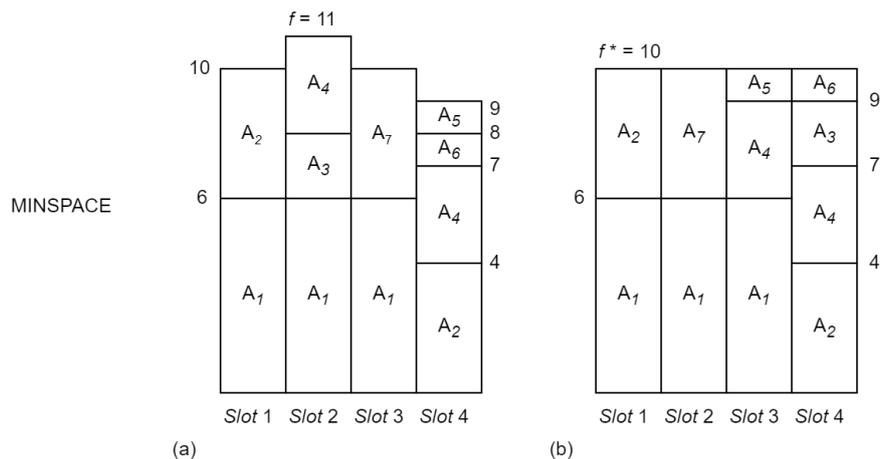


Figura 1: Exemplo de soluções para o MINSPACE usando as propagandas da Tabela 1, e considerando $N = 4$. Em (a) temos uma solução viável, em que $f = 11$. Em (b) temos uma solução ótima, em que $f = 10$.

Na Figura 2 são apresentados exemplos de soluções para o MAXSPACE com as propagandas da Tabela 1.

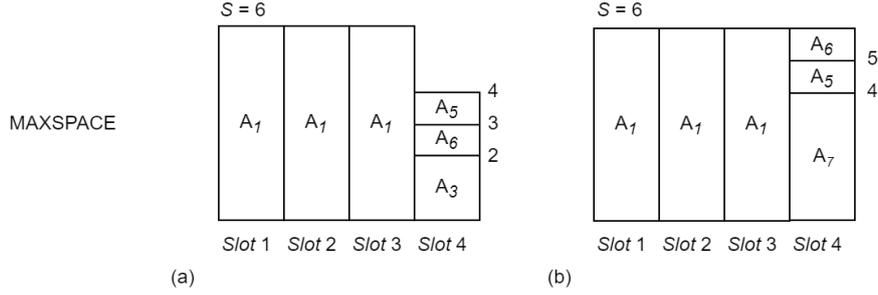


Figura 2: Exemplo de soluções para o MAXSPACE usando as propagandas da Tabela 1, $S = 6$ e $N = 4$. Em (a) temos uma solução viável. Em (b) temos uma solução ótima.

Neste projeto, abordaremos problemas de disposição de propagandas do ponto de vista prático, com o projeto de algoritmos exatos e heurísticas, e teórico, considerando algoritmos de aproximação. O foco do trabalho será em algoritmos exatos, pois são pouco explorados na literatura. Algoritmos de aproximação e heurísticas serão abordados de acordo com a disponibilidade de tempo do projeto. O objetivo é abordar variantes realistas para esses problemas, ou seja, que consideram restrições que ocorrem na prática na venda de anúncios. Na Seção 4 apresentamos os objetivos deste trabalho com mais detalhes, além das variantes que foram pouco exploradas na literatura e que podem ser consideradas.

1.1 Justificativa

Com o crescimento das mídias sociais como Facebook, Twitter e LinkedIn, ocorreu o grande impulso para a propaganda na internet. Por exemplo, no ano de 2013 foram gastos cerca de US\$5,1 bilhões em propaganda em redes sociais e o valor estimado para o ano de 2018 é de aproximadamente US\$15 bilhões de dólares. O Facebook arrecadou, sozinho, US\$2,96 bilhões em propagandas no terceiro trimestre de 2014 [15].

De forma geral, o valor arrecadado em propagandas na internet cresceu muito no século 21. Em 2013 o valor total arrecadado foi de US\$42,78 bilhões, um crescimento de 17% em relação ao ano anterior. Estima-se que o gasto em propagandas na internet no ano de 2016 tenha alcançado 35% de todo o gasto em propaganda, incluindo propagandas para a televisão. A expectativa é de que o setor movimente cerca de US\$143 bilhões em 2017 [15]. Em 2016, anúncios em *banners* representaram 31,4% dos anúncios de

internet (considerando *banners* para plataformas móveis), o que representa um valor de US\$22,7 bilhões de dólares [22].

O crescimento na exibição de anúncios na internet criou uma indústria multibilionária. Com isso, algoritmos para problemas de disposição de propagandas têm uma grande importância prática, pois é necessário achar boas soluções de forma rápida. Além disso, os problemas abordados representam um grande desafio teórico, pois são NP-difíceis.

Este pedido de bolsa de mestrado visa trabalhar em atividades integrantes do Projeto Regular "Problemas de Corte e Empacotamento: Abordagens Práticas e Teóricas" (FAPESP número 2016/23552-7) e do Projeto Temático "Investigação de Problemas Difíceis do Ponto de Vista Algorítmico e Estrutural" (FAPESP número 2015/11937-9).

O candidato já realizou trabalhos na área de Otimização Combinatória, sendo o último o seu Trabalho de Conclusão de Curso de sua graduação pela Universidade Federal do Ceará (UFC) [20], que lhe rendeu uma publicação no Simpósio Brasileiro de Pesquisa Operacional (SBPO) [21]. Este projeto de pesquisa é importante para dar continuidade a formação do candidato e pretende contribuir com o avanço científico da área de Otimização Combinatória.

1.2 Trabalhos Relacionados

A seguir apresentaremos trabalhos encontrados na literatura que abordam os problemas MAXSPACE e MINSPACE.

Adler et al. [1] apresentaram uma 2-aproximação chamada *Largest-Size Least-Full* (LSLF) para o problema MINSPACE. A complexidade de LSLF é da ordem de $O(n N \log N)$, em que N é o número de *slots*.

Dawande et al. [7] apresentaram uma 2-aproximação chamada *LP Rounding* (LPR) para o problema MINSPACE usando programação linear. Dawande et al. [7] também apresentaram uma r -aproximação chamada *Largest-Frequency Least-Full* (LFLF), descrito na Seção 3.1.2, para o problema MINSPACE. A complexidade de LFLF é a mesma de LSLF, $O(n N \log N)$.

Alguns casos particulares para o problema MINSPACE podem ser considerados. O primeiro ocorre quando todas as frequências são iguais, ou seja, $w_1 = w_2 = \dots = w_n$. Nesse caso, chamamos o problema de MIN_w . O segundo caso ocorre quando todas as frequências são iguais, e o número de *slots* N é um múltiplo da frequência w , que representa o valor da frequência de todas as propagandas. Assim, N/w é um inteiro e chamamos essa variação de $\text{MIN}_{N/w}$. O terceiro caso ocorre quando todas as propagandas possuem a mesma altura, ou seja, $s = s_1 = s_2 = \dots = s_n$. Chamamos esse caso de MIN_s [7]. De forma análoga, definimos MAX_w , $\text{MAX}_{N/w}$ e MAX_s como casos particulares de MAXSPACE [7].

O algoritmo LSLF é uma $\left(\frac{4}{3} - \frac{1}{3N/w}\right)$ -aproximação para $\text{MIN}_{N/w}$ [7]. Os algoritmos LSLF e LFLF são ótimos para o problema MIN_s [7].

Uma $\frac{1}{2}$ -aproximação para o MAXSPACE, chamada SUBSET-LSLF, foi apresentada por Adler et al. [1]. Dawande et al. [7] apresentam uma $\left(\frac{1}{4} + \frac{1}{4N}\right)$ -aproximação para o MAXSPACE, chamada MAX1. Esse algoritmo divide as propagandas em duas partes, baseando-se na frequência. Dawande et al. [7] também apresentam outras duas aproximações para esse problema, MAX2 e MAX3. MAX3 é uma $\frac{1}{3}$ -aproximação para MAX_w e uma $\frac{1}{2}$ -aproximação para $\text{MAX}_{N/w}$.

Dean e Goemans [8] propõem uma $\frac{3}{4}$ -aproximação para o problema MINSPACE utilizando o algoritmo de Graham, originalmente desenvolvido para o problema de alocação de tarefas para um número específico de máquinas em paralelo.

Wendan e Dingwei [25] apresentam três heurísticas para os problemas MINSPACE e MAXSPACE: *Smallest-Size Least-Full* (SSLF) para o problema MINSPACE, *Largest-Size Most-Full* (LSMF) para o MAXSPACE e um algoritmo híbrido entre SSLF e LSMF chamado MIN, para o problema MINSPACE. O algoritmo SSLF ordena as propagandas em ordem não decrescente de tamanho e as adiciona, seguindo a ordem, aos *slots* menos cheios, descartando as propagandas que excederem o tamanho máximo do *slot*. Já o algoritmo LSMF ordena as propagandas de forma não crescente de tamanho e as adiciona, nessa ordem, aos *slots* mais cheios, descartando as propagandas que excederem o limite de tamanho dos *slots*.

Kumar et al. [16] utiliza uma heurística combinada a um algoritmo genético para desenvolver um algoritmo genético híbrido para o problema MAXSPACE e compara os resultados com os da formulação para programação linear inteira.

Freund e Naor [9] consideram o problema MAXSPACE e apresentam uma $(3 + \varepsilon)$ -aproximação para o caso geral e uma $(2 + \varepsilon)$ -aproximação para os casos em que a altura de todas as propagandas não ultrapassa a metade do tamanho do *banner* ou quando a altura das propagandas está no intervalo $(S/2, S)$.

O Problema do Empacotamento Bidimensional e o Problema do Escalonamento, descritos a seguir, são semelhantes a problemas de disposição de propagandas, assim o estudo de técnicas usadas em tais problemas pode ser interessante para este trabalho.

Em algumas aplicações industriais, é necessário organizar um conjunto de itens retangulares em estoques de forma a reduzir o desperdício de espaço. Nessas aplicações, em geral, as unidades de armazenamento são retangulares e é necessário adicionar todos os itens, sem sobrepor, com o objetivo de minimizar a quantidade de armazenamento utilizado. Esse problema é conhecido na literatura como Problema do Empacotamento Bidimensional [17].

O **Problema do Empacotamento Bidimensional** (2BP) é definido da seguinte maneira: dado um conjunto de itens $I = \{i_1, i_2 \dots i_k\}$, em que cada item i possui altura h_i e largura w_i , e um conjunto ilimitado de caixas idênticas de altura H e largura W ; aloque todos os itens em caixas, de forma a reduzir o número de caixas utilizadas, sem que haja interseção entre itens e entre itens e a borda das caixas [17].

Chung et al. [6] apresentaram uma 2,125-aproximação para o 2BP. Outros trabalhos fornecem aproximações melhores: como Kenyon e Rémila [13] que apresentam uma $(2+\varepsilon)$ -aproximação (para qualquer $\varepsilon > 0$), Caprara [5] propõe uma $(1.691+\varepsilon)$ -aproximação (para qualquer $\varepsilon > 0$) e Bansal et al. [3] propõem uma 1,525-aproximação.

Lodi et al. [18] propuseram uma heurística para o 2BP e uma busca tabu unificada que se adapta ao problema apenas modificando a heurística usada para explorar a vizinhança. Em Lodi et al. [17], são discutidos modelos matemáticos, algoritmos de aproximação, heurísticas e meta-heurísticas e abordagens exatas que usam enumeração para o 2BP.

O **Problema do Escalonamento** consiste em, dado um conjunto $S = \{s_1, s_2 \dots s_k\}$ de tarefas, um valor w_i indicando o tempo necessário para que a tarefa i seja completamente executada e um número k de processadores, atribuir tarefas aos processadores de forma a reduzir o tempo total de execução [23].

Hochbaum e Shmoys [12] apresentam uma $(\frac{1}{5} + 2^{-k})$ -aproximação e uma $(\frac{1}{6} + 2^{-k})$ -aproximação que executam, respectivamente, em $O(n(k + \log n))$ e $O(n(km^4 + \log n))$, para o Problema do Escalonamento, em que k é o número de processadores.

Guan et al. [11] apresentam uma heurística para o Problema do Escalonamento, considerando que os processadores estão dispostos em uma linha reta, e que cada tarefa requer processamento simultâneo por vários processadores consecutivos.

2 Abordagens de Pesquisa

Na Teoria da Computação um algoritmo é chamado de eficiente se resolve um determinado problema em tempo polinomial no tamanho da representação da entrada. Um problema é chamado de *Problema de Decisão* quando a solução consiste em responder “sim” ou “não”. A classe de todos os problemas de decisão que podem ser resolvidos em tempo polinomial é chamada de P e chamamos de NP a classe de problemas de decisão que podemos verificar se a resposta é “sim” em tempo polinomial. A grande questão em aberto da Teoria da Computação é determinar se $P = NP$.

Chamamos de NP-difíceis os problemas que são tão difíceis quanto qualquer problema em NP. Como MAXSPACE e MINSPACE são NP-difíceis [16], não existe algoritmo para resolvê-los de forma eficiente, a menos que $P = NP$ [19].

Neste projeto, desejamos criar novos algoritmos exatos, heurísticas e algoritmos de aproximação para problemas de disposição de propagandas. A seguir apresentamos as abordagens de pesquisa consideradas.

2.1 Algoritmos Exatos

No contexto de algoritmos exatos, procuramos desenvolver algoritmos para encontrar uma solução ótima. Em problemas de otimização combinatória existe, em geral, um número muito grande de soluções viáveis e enumerá-las pode tomar muito tempo. Por isso, é importante projetar um algoritmo que encontre soluções ótimas mais rapidamente.

Uma das principais abordagens para projetar algoritmos exatos é a Programação Linear Inteira (PLI), em que o problema primeiramente é formulado usando Programação Linear com a restrição que todas ou algumas variáveis assumam valores inteiros [2]. Em PLI podemos usar métodos exatos como o *Branch and Bound*, o *Branch and Price* e o *Branch and Cut*.

Algoritmos exatos também podem ser baseados em programação dinâmica e enumeração (usando o método *Branch and Bound*, por exemplo), entre outros métodos. O método *Branch and Bound* consiste na enumeração de todas as soluções candidatas, através da qual é possível eliminar subconjuntos de soluções infrutíferas, usando cálculos de limites superiores e inferiores. Já a programação dinâmica pode ser aplicada quando a solução ótima de um problema pode ser calculada a partir de soluções para problemas menores, que sobrepostos compõem o problema original.

2.2 Heurísticas

Com *heurísticas* estamos interessados em encontrar soluções boas o suficiente, mas não necessariamente ótimas. Um algoritmo heurístico pode encontrar uma solução cujo valor é distante do valor da ótima, porém, para alguns problemas práticos uma heurística pode encontrar soluções boas o suficiente e com a vantagem de calculá-las rapidamente. Em geral heurísticas são interessantes pelas limitações encontradas para resolver o problema de forma exata, tais como limitação de processamento, memória e tempo para encontrar uma solução ótima.

Já as *meta-heurísticas* são heurísticas de alto nível, que não foram desenvolvidas para um problema específico, mas que podem ser aplicadas a vários problemas [10]. Algumas das principais meta-heurísticas são as de busca tabu, GRASP, BRKGA e algoritmos

genéticos.

2.3 Algoritmos de Aproximação

Muitas vezes, na prática, é suficiente encontrar soluções com valor próximo ao valor da solução ótima e que sejam calculadas de forma eficiente. Um *Algoritmo de Aproximação* encontra, em tempo polinomial, uma solução viável que possui valor garantidamente próximo daquele da solução ótima [24].

Considere um problema de maximização Π , tal que $\text{OPT}(E)$ é o valor de uma solução ótima para a instância E . Dizemos que um algoritmo A é uma α -aproximação para o problema Π se ele consome tempo polinomial no tamanho da representação binária de E e produz uma solução com valor $A(E)$, tal que $A(E) \geq \alpha \cdot \text{OPT}(E)$, para $\alpha \leq 1$. Quando temos um problema Π de minimização, uma α -aproximação A é tal que A é um algoritmo polinomial no tamanho da representação binária da entrada E e $A(E) \leq \alpha \cdot \text{OPT}(E)$, para $\alpha \geq 1$.

3 Algoritmos da Literatura

A seguir descrevemos alguns dos algoritmos existentes para o MAXSPACE e o MINS-SPACE.

3.1 MINS-SPACE

Nesta seção, são apresentados exemplos de algoritmos encontrados na literatura para o MINS-SPACE.

3.1.1 Algoritmo Exato

Dawande et al. [7] definiram a seguinte formulação em Programação Linear Inteira para o problema MINS-SPACE.

Seja x_{ij} uma variável inteira que recebe o valor 1 se a propaganda A_i foi adicionada ao *slot* j e recebe o valor 0 caso contrário; e sejam w_i e s_i , respectivamente, a frequência

e o tamanho da propaganda A_i , $i = 1, 2, \dots, n$.

$$\begin{aligned}
& \min && B \\
& \text{sujeito a} && \sum_{i=1}^n s_i x_{ij} \leq B, && j = 1, 2, \dots, N \\
& && \sum_{j=1}^N x_{ij} = w_i, && i = 1, 2, \dots, n \\
& && B \in \mathbb{R}, x_{ij} \in \{0, 1\}
\end{aligned}$$

O primeiro conjunto de restrições, limita a altura do *banner* por B . O segundo conjunto de restrições garante que uma propaganda A_i será adicionada exatamente w_i vezes aos *slots*.

3.1.2 Algoritmo de Aproximação

O Algoritmo LFLF (*Largest-Frequency Least-Full*) [7] é um algoritmo de aproximação para o problema MINSPACE e consiste em ordenar de forma não decrescente as propagandas pela frequência, assim $w_1 \geq w_2 \geq \dots \geq w_n$. Depois de ordenadas, o algoritmo adiciona, seguindo a ordem, a propaganda A_i aos w_i *slots* menos cheios.

Teorema 1. (Dawande et al. [7]) *O algoritmo LFLF é uma r -aproximação para o problema MINSPACE, em que:*

$$r = \begin{cases} 2 - 1/(N - w_{\min} + 1), & \text{se } w_{\min} \leq (N + 1)/2 \\ \max\{1, (2N/(N + 2))((S^* - s_{\min})/S^*)\}, & \text{caso contrário,} \end{cases}$$

com $w_{\min} = \min\{w_1, \dots, w_n\}$, $s_{\min} = \min\{s_1, \dots, s_n\}$, $S^* = \sum_{i=1}^N s_i$.

Demonstração. (Dawande et al. [7]) Vamos reindexar as propagandas $A = \{A_1, A_2, \dots, A_n\}$ de forma que $w_1 \geq w_2 \geq \dots \geq w_n$. Assim, $w_{\min} = w_n$. Note que a última propaganda adicionada por LFLF é A_n . Seja f^* o valor da solução ótima para uma dada instância A e seja f o valor da solução produzida pelo algoritmo LFLF para a instância A . Sem perda de generalidade podemos supor que o maior preenchimento $f(A)$ é dado pelo *slot* em que uma cópia da propaganda A_n é adicionada,

caso contrário podemos restringir nossa análise ao subconjunto de propagandas até a primeira propaganda que atinge $f(A)$.

Começamos com o caso em que $w_n \leq (N + 1)/2$, considerando que a última propaganda, A_n , é adicionada aos w_n slots menos cheios. Antes de adicionar A_n , o preenchimento de pelo menos $N - w_n + 1$ slots deve ser maior ou igual a $f - s_n$. Isso implica que $\sum_{i=1}^n s_i w_i \geq (f - s_n)(N - w_n + 1) + s_n w_n$. Após rearranjar, temos que

$$f \leq s_n \left(1 - \frac{w_n}{N - w_n + 1} \right) + \frac{\sum_{i=1}^n s_i w_i}{N - w_n + 1} \quad (1)$$

Notamos dois limites inferiores para f^* :

$$f^* \geq s_n \quad (2)$$

e

$$f^* \geq \frac{\sum_{i=1}^n s_i w_i}{N}. \quad (3)$$

Como $w_n < (N + 1)/2$, temos que

$$1 - \frac{w_n}{N - w_n + 1} > 0. \quad (4)$$

Usando (1)–(4), encontramos

$$f \leq f^* \left(1 - \frac{w_n}{N - w_n + 1} \right) + \frac{\sum_{i=1}^n s_i w_i}{N - w_n + 1} \leq f^* \left(1 - \frac{w_n}{N - w_n + 1} \right) + \frac{N f^*}{N - w_n + 1}$$

e concluímos que $f/f^* \leq 2 - (1/(N - w_n + 1)) = 2 - (1/(N - w_{\min} + 1))$.

Agora considere o caso em que $w_n > (N + 1)/2$.

Primeiro mostraremos que se $f = S^*$, então $f^* = S^*$. Suponha que $f = S^*$. Após adicionar a j -ésima propaganda, seja v_j o número de slots com o preenchimento de pelo menos $\sum_{i=1}^j s_i$ e note que $v_j \geq 1$, para todo $1 \leq j \leq n$. Note também

que $v_1 = w_1$ e $v_i = w_i - (N - v_{i-1})$, para todo $2 \leq i \leq n$. Somando essas equações, obtemos $\sum_{i=1}^n w_i = (n - 1)N + v_n \geq (n - 1)N + 1$. Conseqüentemente, ao menos um *slot* terá uma cópia de cada propaganda na solução ótima, e assim $f^* = f = S^*$ e a solução encontrada por LFLF é ótima.

Agora, suponha que $f < S^*$. Então, $f \leq S^* - s_{min}$. Como a última propaganda adicionada, A_n , tem a menor frequência, w_n , e $w_n > (N + 1)/2$, ou seja, $w_n \geq (N + 2)/2$, temos

$$\sum_{i=1}^n s_i w_i \geq \left(\frac{N + 2}{2} \right) S^* \geq \left(\frac{N + 2}{2} \right) \left(\frac{S^*}{S^* - s_{min}} \right) f. \quad (5)$$

A partir de (3) e (5), obtemos

$$Nf^* \geq \sum_{i=1}^n s_i w_i \geq \left(\frac{N + 2}{2} \right) \left(\frac{S^*}{S^* - s_{min}} \right) f, \quad (6)$$

e a partir de (6), temos que

$$\frac{f}{f^*} \leq \left(\frac{2N}{N + 2} \right) \left(\frac{S^* - s_{min}}{S^*} \right).$$

Como $f \geq f^*$, concluímos que

$$\frac{f}{f^*} \leq \max \left\{ 1, \left(\frac{2N}{N + 2} \right) \left(\frac{S^* - s_{min}}{S^*} \right) \right\}. \quad \square$$

3.2 MAXSPACE

Nesta seção, são apresentados exemplos de algoritmos para o problema MAXSPACE.

3.2.1 Algoritmo Exato

Dawande et al. [7] definiram a seguinte formulação em Programação Linear Inteira para o problema MAXSPACE.

Seja x_{ij} uma variável inteira que recebe o valor 1 se a propaganda A_i foi adicionada

ao *slot* j e recebe o valor 0 caso contrário; e seja y_i uma variável inteira que recebe o valor 1 quando a propaganda A_i é adicionada a algum *slot* e y_i recebe 0 caso contrário.

$$\begin{aligned} & \max \sum_{j=1}^N \sum_{i=1}^n s_i x_{ij} \\ & \text{sujeito a} \quad \sum_{i=1}^n s_i x_{ij} \leq S, \quad j = 1, 2, \dots, N \\ & \quad \quad \quad \sum_{j=1}^N x_{ij} = w_i y_i, \quad i = 1, 2, \dots, n \\ & \quad \quad \quad x_{ij} \in \{0, 1\}, y_i \in \{0, 1\} \end{aligned}$$

O primeiro conjunto de restrições garante que a soma dos tamanhos das propagandas adicionadas a um *slot* deve ser no máximo S . O segundo conjunto de restrições garante que uma propaganda A_i será adicionada exatamente w_i vezes nos *slots*.

3.2.2 Algoritmo de Aproximação

O algoritmo MAX1 apresentado por Dawande et al. [7] é uma $(1/4 + 1/4N)$ -aproximação para o problema MAXSPACE. Ele utiliza como subrotina o procedimento LSLF.

O procedimento LSLF (*Largest-Size Least-Full*) [1] consiste em ordenar de forma não decrescente as propagandas pela altura, assim $s_1 \geq s_2 \geq \dots \geq s_n$. Depois de ordenadas, LSLF adiciona, seguindo essa ordem, a propaganda A_i aos w_i *slots* menos cheios, caso isso seja possível, senão, o procedimento descarta A_i , seguindo para a próxima propaganda.

Algoritmo 1 Algoritmo MAX1

- 1: **procedimento** MAX1(A, S)
 - 2: $\underline{u} \leftarrow \{A_i | w_i < (N + 1)/2\}, B_{\underline{u}} \leftarrow \sum_{A_i \in \underline{u}} s_i w_i$
 - 3: $\bar{u} \leftarrow A \setminus \underline{u}, B_{\bar{u}} \leftarrow \sum_{A_i \in \bar{u}} s_i w_i$
 - 4: $P_0 \leftarrow \sum_{i=1}^N s_i w_i$
 - 5: Se $B_{\underline{u}} > P_0/2$, $u \leftarrow \underline{u}$
 - 6: Se $B_{\bar{u}} \leq P_0/2$, $u \leftarrow \bar{u}$
 - 7: $S_u \leftarrow \text{LSLF}(u, S)$
 - 8: comece com S_u e execute LSLF($A \setminus u, S$)
-

Seja $A' \subseteq A$ o conjunto de propagandas escolhido pelo algoritmo e seja $P = \sum_{A_i \in A'} s_i w_i$ o peso da solução encontrada pelo algoritmo. Seja P^* o peso da solução ótima, podemos destacar dois limites superiores para P^* :

$$P^* \leq \sum_{i=1}^N s_i w_i = P_0 \quad (7)$$

e

$$P^* \leq NS. \quad (8)$$

Teorema 2. (Dawande et al. [7]) *MAX1 é uma $(1/4 + 1/4N)$ -aproximação para o problema MAXSPACE.*

Demonstração. (Dawande et al. [7]) Note que $B_{\bar{u}} \geq P_0/2$ ou $B_u \geq P_0/2$.

Caso 1: $B_u \geq P_0/2$. Se todas as propagandas em u são escolhidos pelo algoritmo, então $P \geq P_0/2$ e conseqüentemente $P/P^* \geq 1/2$ a partir de (7). Suponha então que, ao menos, uma propaganda, quando adicionada, viola o tamanho de algum *slot* e é descartada. Seja A_f a primeira propaganda descartada. Note que A_f não pode ser a primeira propaganda da lista, pois $s_i \leq S$ e $w_i \leq N$ para cada $A_i \in A$. Dado que A_f não pode ser adicionada, o preenchimento de ao menos $N - w_f + 1$ *slots* é maior que $S - s_f$. Assim, ao menos $N - w_f + 1$ *slots* estão ao menos preenchidos até a metade. Essa observação combinada com o fato de que $w_f < (N + 1)/2$ implica

$$P \geq \left(N - \frac{N + 1}{2} + 1 \right) \left(\frac{S}{2} \right) \geq NS \left(\frac{1}{4} + \frac{1}{4N} \right)$$

o que resulta no limite, dado que $P^* \leq NS$, a partir de (8).

Caso 2: $B_{\bar{u}} \geq P_0/2$. Novamente, se todas as propagandas em \bar{u} são adicionadas sem nenhuma violação de limite em algum *slot*, como no Caso 1, temos $P/P^* \geq 1/2$.

Seja A_f a primeira propaganda descartada pelo algoritmo. Dado que $w_f \geq 1$, existe pelo menos um *slot* j com preenchimento maior que $S - s_f$. E como as propagandas são adicionadas em ordem decrescente de tamanho, o *slot* j está pelo menos cheio até a metade. Seja $J \subseteq A$ o conjunto de propagandas que possuem uma cópia no *slot* j .

Então $\sum_{A_k \in J} s_k \geq S/2$. Usando $w_k \geq (N+1)/2$, para todo $A_k \in J$, temos que

$$P \geq \sum_{A_k \in J} s_k w_k \geq \left(N - \frac{N+1}{2} + 1 \right) \sum_{A_k \in J} s_k \geq NS \left(\frac{1}{4} + \frac{1}{4N} \right)$$

e obtemos o limite requerido. □

4 Objetivos

O objetivo principal desse projeto é desenvolver novos algoritmos para problemas de disposição de propagandas. Mais especificamente, desejamos desenvolver algoritmos exatos, heurísticas ou algoritmos de aproximação para os problemas MAXSPACE e MINSPACE, levando em consideração variantes encontradas na prática para a classe de problemas de disposição de propagandas.

Objetiva-se também realizar testes computacionais com instâncias das variantes abordadas e que possuam semelhança com os dados encontrados em situações reais. Espera-se que os algoritmos desenvolvidos obtenham a solução ótima ou o mais próximo possível da ótima em tempo de execução viável. Além disso, deseja-se que os resultados obtidos possam ser publicados em conferências e periódicos importantes da área.

A seguir apresentamos as variantes para problemas de disposição de propagandas que consideramos mais interessantes e que podem ser abordadas durante o trabalho.

As abordagens citadas na Seção 1 consideram que o lucro de uma propaganda é dada pela altura multiplicada pelo número de *slots* (ou unidades de tempo) em que ela aparece. Na prática o lucro de uma propaganda pode ser influenciado por outros fatores, como o número de cliques (ou a esperança do número de cliques) que a propaganda gera para o anunciante [4].

A unidade de tempo relativa a cada *slot* pode representar dias, semanas ou períodos menores, como minutos e segundos. Nos casos em que cada *slot* é exibido por segundos ou minutos, o *banner* será dinâmico, ou seja, será atualizado enquanto o usuário está na página. Assim, em *banners* dinâmicos podemos levar em consideração a probabilidade de

cada *slot* ser exibido, pois, na prática, o tempo que um usuário passa em um *site* pode não ser suficiente para que todos os N *slots* sejam apresentados.

Outra variante que pode ser interessante na prática é considerar que cada propaganda possui um orçamento (ao invés de uma frequência) e que esse orçamento é reduzido quando a propaganda é exibida (considerando a probabilidade de clique, por exemplo). Também é importante observar que, na prática, uma propaganda pode possuir um *deadline*. Por exemplo, propagandas para o natal devem ser exibidas até dia 24 de dezembro. Dessa forma, propagandas com *deadlines* mais curtos devem ser priorizadas. Uma propaganda também pode possuir uma *release date*, que indica a data em que a propaganda pode começar a ser exibida. A probabilidade de clique de uma propaganda também pode variar com os dias. Por exemplo, propagandas de natal provavelmente receberão mais cliques à medida que o natal se aproxima.

Outra opção é considerar um modelo de cobranças baseado em leilão, como é feito pelo Google. Nesse modelo, cada anunciante indica quanto deseja pagar por propaganda. De acordo com os lances, as propagandas são escolhidas, de forma que o maior lance paga o valor do segundo maior, o segundo paga o valor do terceiro e assim por diante [14].

Começaremos abordando uma versão com *deadline* e *release date*, no caso particular do MAXSPACE em que o lucro de uma propaganda não é necessariamente $s_i w_i$. O foco principal do trabalho é em algoritmos exatos, pois são pouco explorados na literatura relacionada a esse problema. Algoritmos de aproximação, heurísticas, as demais variantes e o problema MINSPACE serão abordados de acordo com a disponibilidade de tempo e andamento da pesquisa.

5 Materiais e Métodos

Este trabalho será realizado no *Laboratório de Otimização Combinatória* (LOCo) do *Instituto de Computação*. Os novos algoritmos exatos e heurísticos desenvolvidos para o Problema de Disposição de Propagandas serão testados empiricamente. Dessa forma, serão necessários servidores para a execução dos testes, já que podem ser necessários

longos períodos de execução. O LOCo disponibiliza servidores para testes de algoritmos, além de um ambiente adequado para estudo e desenvolvimento dos mesmos.

O *Instituto de Computação* disponibiliza livros para os alunos. Dessa forma, é possível ter acesso aos principais livros da área. A Unicamp e o portal de periódicos da *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES) disponibilizam gratuitamente os artigos necessários para a pesquisa.

Durante o desenvolvimento deste projeto, o candidato cursará disciplinas do *Instituto de Computação*, além de estudar livros e artigos importantes da área, e assistirá aos seminários do LOCo e às defesas dos colegas, a fim de obter experiência para sua própria defesa. Serão realizadas reuniões semanais com o orientador para discutir o desenvolvimento do projeto e o desempenho do candidato nas disciplinas. O candidato já cursou as seguintes disciplinas do *Instituto de Computação*: *Programação Paralela*, *Complexidade de Algoritmos I*, *Aprendizado de Máquina e Reconhecimento de Padrões* e obteve conceito A em todas.

6 Cronograma

Durante o primeiro ano do projeto o candidato cursará as disciplinas do Instituto de Computação e, durante o terceiro semestre do curso, o candidato participará do Programa de Estágio a Docência (PED) da Unicamp. Após a defesa do Exame de Qualificação de Mestrado (EQM), espera-se iniciar o desenvolvimento de algoritmos para a abordagem escolhida, considerando algoritmos exatos, heurísticos e/ou algoritmos de aproximação. Paralelamente ao desenvolvimento dos algoritmos, será escrita a Dissertação de Mestrado do candidato. As soluções desenvolvidas serão submetidas a avaliações, como descrito na Seção 7. Ao final do segundo ano de projeto, o candidato defenderá a sua Dissertação de Mestrado.

As atividades que devem ser desenvolvidas e o cronograma são apresentados na Tabela 2.

Tabela 2: Cronograma

| Atividades | 2017 | | | | 2018 | | | |
|--|------------|---|---|---|------------|---|---|---|
| | Trimestres | | | | Trimestres | | | |
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| Estudar as variantes do problema e definir quais abordar | ✓ | ✓ | | | | | | |
| Cursar as disciplinas do IC | ✓ | ✓ | • | • | | | | |
| Escrever o EQM e Defender EQM | | ✓ | • | | | | | |
| Participação do PED | | | | | • | • | | |
| Desenvolver soluções para a abordagem escolhida | | | | • | • | • | | |
| Testes das soluções desenvolvidas | | | | | | • | • | • |
| Escrita da Dissertação e Defesa da Dissertação | | | | | | • | • | • |

7 Forma de análise dos resultados

Os algoritmos propostos serão submetidos a avaliações teóricas (nos casos de algoritmos de aproximação) e empíricas (usando instâncias geradas aleatoriamente e instâncias da literatura). Os testes empíricos considerarão classes variadas de instâncias, com o objetivo de explorar a diferença entre os algoritmos e identificar qual é melhor para o problema em determinada classe de instância. Os testes empíricos permitirão analisar o consumo de tempo e a qualidade das soluções encontradas pelos algoritmos. Serão utilizadas instâncias geradas de forma aleatória e instâncias obtidas da literatura.

Outra forma de avaliar o trabalho desenvolvido é realizar publicações dos resultados obtidos em conferências e periódicos importantes da área.

Referências

- [1] Adler, M., Gibbons, P. B., and Matias, Y. (2002). Scheduling space-sharing for internet advertising. *Proceedings of Journal of Scheduling*, 5(2):103–119.
- [2] Alves, R. and Delgado, C. (1997). Programação linear inteira. *Proceedings of Porto, Faculdade de Economia–Universidade do Porto. Apostila*.
- [3] Bansal, N., Caprara, A., and Sviridenko, M. (2009). A new approximation method for set covering problems, with applications to multidimensional bin packing. *Proceedings of SIAM Journal on Computing*, 39(4):1256–1278.
- [4] Briggs, R. and Hollis, N. (1997). Advertising on the web: Is there response before click-through? *Proceedings of Journal of Advertising research*, 37(2):33–46.
- [5] Caprara, A. (2002). Packing 2-dimensional bins in harmony. In *Proceedings of Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, pages 490–499.
- [6] Chung, F. R., Garey, M. R., and Johnson, D. S. (1982). On packing two-dimensional bins. *Proceedings of SIAM Journal on Algebraic Discrete Methods*, 3(1):66–76.
- [7] Dawande, M., Kumar, S., and Sriskandarajah, C. (2003). Performance bounds of al-

- gorithms for scheduling advertisements on a web page. *Proceedings of Journal of Scheduling*, 6(4):373–394.
- [8] Dean, B. C. and Goemans, M. X. (2003). Improved approximation algorithms for minimum-space advertisement scheduling. In *Proceedings of International Colloquium on Automata, Languages, and Programming*, pages 1138–1152.
- [9] Freund, A. and Naor, J. S. (2002). Approximating the advertisement placement problem. In *Proceedings of International Conference on Integer Programming and Combinatorial Optimization*, pages 415–424.
- [10] Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Proceedings of Computers & operations research*, 13(5):533–549.
- [11] Guan, Y., Xiao, W.-Q., Cheung, R. K., and Li, C.-L. (2002). A multiprocessor task scheduling model for berth allocation: heuristic and worst-case analysis. *Proceedings of Operations Research Letters*, 30(5):343–350.
- [12] Hochbaum, D. S. and Shmoys, D. B. (1987). Using dual approximation algorithms for scheduling problems theoretical and practical results. *Proceedings of Journal of the ACM*, 34(1):144–162.
- [13] Kenyon, C. and Rémila, E. (2000). A near-optimal solution to a two-dimensional cutting stock problem. *Proceedings of Mathematics of Operations Research*, 25(4):645–656.
- [14] King, M., Atkins, J., and Schwarz, M. (2007). Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *The American economic review*, 97(1):242–259.
- [15] Kumar, S. (2015). *Optimization Issues in Web and Mobile Advertising: Past and Future Trends*. Springer.
- [16] Kumar, S., Jacob, V. S., and Sriskandarajah, C. (2006). Scheduling advertisements on a web page to maximize revenue. *Proceedings of European journal of operational research*, 173(3):1067–1089.
- [17] Lodi, A., Martello, S., and Monaci, M. (2002). Two-dimensional packing problems: A survey. *European journal of operational research*, 141(2):241–252.
- [18] Lodi, A., Martello, S., and Vigo, D. (1999). Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. *Proceedings of INFORMS Journal on Computing*, 11(4):345–357.
- [19] Meurant, G. (2014). *Algorithms and complexity*. Elsevier.
- [20] Silva, M. R. C. and Tavares, W. A. (2016). Algoritmos para o problema do kplex máximo utilizando paralelismo de bits e coloração. *Universidade Federal do Ceará - Campus Quixadá*.
- [21] Silva, M. R. C., Tavares, W. A., Dias, F., and Campêlo, M. (2017). Algoritmo branch-and-bound para o problema do kplex máximo. *XLIX Simpósio Brasileiro de Pesquisa Operacional*.
- [22] Silverman, D. (2010). Iab internet advertising revenue report. *Proceedings of Interactive Advertising Bureau. New York*, page 26.
- [23] Ullman, J. D. (1975). Np-complete scheduling problems. *Proceedings of Journal of Computer and System sciences*, 10(3):384–393.
- [24] Vazirani, V. V. (2013). *Approximation algorithms*. Springer Science & Business Media.
- [25] Wendan, Z. and Dingwei, W. (2010). Study on internet advertising placement problem. In *Proceedings of Logistics Systems and Intelligent Management, 2010 International Conference on*, volume 3, pages 1798–1801.