

5 Avaliação dos Algoritmos

Como discutimos na Seção 2.4, apesar de podermos muitas vezes determinar a competitividade de um algoritmo online, é muito importante sua avaliação experimental, pois diversas situações limite, que são levadas em consideração na análise de pior caso, dificilmente ocorrem na prática. Isto é, podem nunca ocorrer ou ocorrer tão poucas vezes que podemos decidir desconsiderá-las no momento de escolhermos o algoritmo que vamos utilizar para resolver um problema real. Em nosso problema torna-se ainda mais importante esta avaliação, já que não temos como determinar teoricamente a competitividade dos algoritmos.

Como não obtivemos dados reais a avaliação dos algoritmos foi feita apenas com dados sintéticos, gerados de modo a simular cenários diversos, onde procuramos reproduzir o que ocorre em casos reais.

Este capítulo contém a avaliação dos algoritmos descritos no Capítulo 4. Primeiramente explicamos como simulamos o ambiente real, da chegada das consultas à exibição dos anunciantes. Depois, como geramos os dados de entrada e medimos o desempenho dos algoritmos. Descrevemos então dois conjuntos cenários com os experimentos e o desempenho dos algoritmos nestas simulações. No primeiro conjunto simulamos os casos onde os anunciantes têm perfis parecidos e no segundo um ambiente com grandes e pequenos anunciantes. Para o primeiro conjunto de cenários, também apresentamos como varia o desempenho dos algoritmos ao longo do dia.

5.1 Simulação do Ambiente Online

Como não temos o ambiente real da máquina de busca recebendo consultas que chegam a todo momento e dos leilões ocorrendo de forma online, implementamos o programa de avaliação de modo que estas informações fossem lidas de arquivos para simularmos esta situação. No caso da saída, como não temos a efetiva exibição do anunciante para cada consulta, definimos um arquivo que demonstra o anunciante atribuído à consulta.

Nesta seção, descrevemos os arquivos de entrada e saída, os recursos utilizados e o tempo de execução.

5.1.1

Arquivos de Entrada e Saída

Os arquivos de entrada e saída estão descritos na Tabela 5.1. Estes arquivos foram definidos para ter uma informação completa por linha com seus campos em colunas separadas por qualquer quantidade de brancos. O identificador de três letras que atribuímos aos arquivos é apenas para facilitar sua identificação durante o texto deste capítulo.

A geração dos dados de entrada é feita de forma aleatória e será explicada mais adiante neste capítulo.

Para alimentar os algoritmos que utilizam predição sobre a quantidade esperada de consultas foi utilizado um arquivo idêntico ao arquivo de saída ATR como se este representasse a expectativa média de consultas por tipo e o resultado ótimo para esta distribuição prevista. Este arquivo, que denominamos OTM, é um arquivo de saída no cálculo da solução ótima e funciona como entrada para os algoritmos preditivos.

Arquivo	Tipo	Conteúdo	Campos
LIM	Entrada	Anunciantes e limites de exibição	anunciante limite
OFT	Entrada	Ofertas por tipo de consulta	anunciante tipo de consulta oferta
CON	Entrada	Consultas diárias por tipo	consulta tipo de consulta
ATR	Saída	Atribuições das consultas aos anunciantes	consulta tipo de consulta anunciante receita
OTM	Entrada Saída	Atribuição ótima das consultas	consulta tipo de consulta anunciante receita

Tabela 5.1: Arquivos de entrada e saída

5.1.2

Recursos Utilizados

Para implementar os algoritmos escolhemos a linguagem C++, por possuir recursos adequados para trabalhar este tipo de algoritmo e dados

envolvidos, além da facilidade para montar a interface com as bibliotecas de programação linear e fluxo em redes.

Os programas foram desenvolvidos no modo console utilizando o editor integrado *Code::Blocks 8.02* e o compilador *GNU G++*. Este ambiente nos permitiu desenvolver em ambiente *Windows*, mantendo um código portátil para outros ambientes.

Para resolver a programação linear utilizamos a biblioteca de programação linear UFFLP.DLL (UFFLP), que tem como base a biblioteca CLP da organização COIN-OR (COIN).

No cálculo do fluxo em redes utilizamos o algoritmo de *Cost Scaling*, desenvolvido por alunos da PUC-Rio para as maratonas de programação.

As simulações foram realizadas em um computador HP com processador Intel Core2Duo de 2,4GHz, 2GB de RAM e sistema operacional Microsoft Windows XP Professional com SP3.

5.1.3

Tempo de Execução

Para um dia de simulação, o tempo de execução dos algoritmo online é $O(mn)$, sendo m o número de consultas e n o de anunciantes. Para cada consulta precisamos percorrer as ofertas feitas pelos anunciantes para aquela consulta, buscando a oferta que maximiza alguma função. No caso do algoritmo Guloso, por exemplo, a maior oferta; nos algoritmos AdWords e Primal-Dual, uma combinação entre oferta e limite de exibição dos anunciantes. Assim, a execução é sempre muito rápida, quase instantânea.

O único tempo significativo em nossa simulação é o cálculo da solução ótima. Em nosso modelo de programação linear temos:

- função objetivo com mn variáveis;
- m restrições com n variáveis;
- n restrições com m variáveis.

No maior exemplo que executamos para geração dos cenários apresentados nesta dissertação usamos $m = 8000$ e $n = 50$, logo: função objetivo com 400.000 variáveis, 50 restrições com 8.000 variáveis cada e 8.000 restrições com 50 variáveis cada.

Na Tabela 5.2 apresentamos o tempo aproximado de execução de um dia de processamento para cada um dos cenários apresentados. Este tempo é referente à simulação completa de um dia e inclui: cálculo da solução ótima e execução dos cinco algoritmos online.

Cenário	n	m	Tempo em minutos (t)
1	50	500	$t \leq 1$
2	50	1000	$t \leq 1$
3	50	2000	$t \leq 1$
4	50	4000	$1 \leq t \leq 4$
5	50	8000	$1 \leq t \leq 16$

Tabela 5.2: Tempo de execução de um dia de processamento

A grande variação nos tempos, quando temos um número alto de consultas, está ligada a menor ou maior competição entre os anunciantes. Quando mais anunciantes estão interessados em uma mesma consulta, há um aumento dos coeficientes diferentes de zero na matriz de restrições e por conseguinte do tempo de processamento.

Como o objetivo do trabalho era avaliar os algoritmos online e não otimizar o tempo de solução, não investimos em buscar uma biblioteca mais rápida para resolver a programação linear ou o fluxo em redes. Isto seria fundamental para simulações com um número maior de consultas e anunciantes.

5.2

Geração dos Dados de Entrada

Para gerar os arquivos de entrada, montamos um programa gerador parametrizável que permite simularmos situações diversas.

Os parâmetros básicos para geração dos arquivos são:

- n : número de anunciantes
- m : número de consultas por dia
- Q : número de tipos de consulta
- Q_{max} : número máximo de consultas que pode ser atendido por dia

O número máximo de consultas que pode ser atendido em um dia depende dos limites de exibição dos anunciantes, ou seja:

$$Q_{max} = \sum_{i=1}^n L_i$$

Dividimos os cenários usados nesta avaliação, baseado na razão m/Q_{max} , buscando identificar os algoritmos que funcionam melhor em cada situação. Esta divisão será explicada mais adiante.

A seguir detalhamos como é feita a geração de cada um dos arquivos de entrada e os parâmetros que permitem algumas variações nesta geração.

5.2.1

Geração das Consultas (CON)

As consultas recebidas ao longo de um dia têm a característica de concentrar a maior parte delas em poucos tipos de consulta e uma parcela menor é distribuída por um conjunto maior de tipos. Para simular este comportamento geramos os arquivos de consultas utilizando distribuições de Zipf.

A distribuição de Zipf define a quantidade de elementos de cada tipo em uma determinada população em áreas de estudos diversas como linguística, física e ciências sociais. Esta distribuição determina uma curva que decresce do primeiro para o último elemento. A função que define a curva pode variar de acordo com uma potência que determina se a curva decresce mais rapidamente ou mais devagar. Quando aumentamos a potência a curva decresce de forma mais acentuada.

Em nossos cenários, distribuimos m consultas por Q tipos de consulta numa distribuição de Zipf de potência p . A quantidade de consultas D_j , a ser atribuída para cada tipo de consulta j , é definida por:

$$D_j = \frac{m}{j^p \sum_{q=1}^Q \left(\frac{1}{q^p}\right)}$$

Na Tabela 5.3, apresentamos um exemplo de distribuição de 1000 consultas por 10 tipos diferentes utilizando potência 1.

Para que a quantidade de consultas de cada tipo não seja sempre a mesma em todos os dias na simulação de cada cenário, primeiramente calculamos a quantidade D_j para cada tipo de consulta j seguindo a distribuição de Zipf com a potência escolhida. Depois sorteamos a quantidade Q_j de consultas a ser efetivamente gerada num intervalo um pouco acima e abaixo da quantidade D_j calculada. Na geração dos cenários mantivemos o sorteio no seguinte intervalo: $0,8D_j \leq Q_j \leq 1,2D_j$.

Exemplificamos, ainda na Tabela 5.3, nas linhas abaixo e acima da distribuição, as quantidades mínima e máxima de consultas de cada tipo que limitam o intervalo do sorteio.

Na última linha, apresentamos a distribuição percentual das consultas por tipo.

5.2.2

Geração de Anunciantes e Limites de Exibição (LIM)

Os anunciantes são sempre gerados de 1 a n , de acordo com o parâmetro n definido para cada cenário. Os limites de exibição dependem do tipo de

Distribuição	Tipos de Consulta										Total
	1	2	3	4	5	6	7	8	9	10	
Zipf $\times 0,8$	273	136	91	68	54	45	38	33	30	27	795
Zipf	343	171	114	85	68	57	48	42	38	34	1000
Zipf $\times 1,2$	410	205	136	102	81	68	57	50	45	40	1194
Percentual	31%	16%	10%	8%	6%	5%	4%	4%	3%	3%	

Tabela 5.3: Consultas por distribuição de Zipf

cenário, isto é, se queremos anunciantes similares ou diferenciados.

5.2.2.1

Geração de Anunciantes Similares

O limite de exibições de cada anunciante é escolhido por sorteio num intervalo de limites mínimo L_{min} e máximo L_{max} . Utilizando limites L_{min} e L_{max} razoavelmente próximos, conseguimos anunciantes com limites parecidos, mas não necessariamente iguais.

Neste tipo de cenário encontramos o valor de Q_{max} de forma aproximada:

$$Q_{max} = n \times \frac{(L_{min} + L_{max})}{2}$$

Considere, por exemplo, um cenário de 50 anunciantes com limites: $L_{min} = 30$ e $L_{max} = 50$. Substituindo os valores na fórmula anterior temos: $Q_{max} = 50 \times (30 + 50)/2 = 2000$. Neste cenário, poderiam ser atendidas 2000 consultas no dia, aproximadamente.

5.2.2.2

Geração de Anunciantes Diferenciados

A idéia neste tipo de cenário é representarmos grandes anunciantes, com limites de exibição maiores, e médios e pequenos anunciantes, com limites de exibição menores. Para atribuir limites aos anunciantes escolhemos a distribuição de Zipf.

Para definir os limites neste tipo de cenário, definimos primeiramente o número total de exibições Q_{max} . Em seguida, distribuimos este total pelos anunciantes obedecendo uma distribuição de Zipf com uma potência que também escolhemos.

Se para um conjunto de 10 anunciantes definimos, $Q_{max} = 1000$, teremos os limites de exibição de cada um dos 10 anunciantes exatamente como a linha central da Tabela 5.3, ou seja, limites de $L_1 = 343$ até $L_{10} = 34$.

5.2.3

Geração das Ofertas (OFT)

A geração das ofertas obedece a três parâmetros:

- b_{min} : valor mínimo das ofertas
- b_{max} : valor máximo das ofertas
- Q_a : por quantos tipos de consulta um anunciante faz ofertas

O parâmetro Q_a permite analisar o comportamento dos algoritmos quando temos um mercado com maior ou menor concorrência entre os anunciantes. Quando usamos um Q_a pequeno, significa que os anunciantes se interessam por poucas consultas e não temos muita competição entre os anunciantes; se aproximamos Q_a do número de tipos de consulta Q , aumentamos esta competição.

Com um Q_a definido, sorteamos até Q_a tipos de consulta para cada anunciante. Permitimos repetição no sorteio e se ocorrer este anunciante terá menos ofertas. Se escolhermos, por exemplo, $Q_a = 5$ e temos $Q = 20$, os anunciantes farão em sua maioria 5 ofertas, mas podem ocorrer casos de anunciantes com 4 ou até menos ofertas. Demos esta liberdade por entender que assim aproximamos melhor as incertezas de um cenário real.

A distribuição dos valores das ofertas depende dos cenários serem de anunciantes similares ou diferenciados.

5.2.3.1

Geração de Ofertas para Anunciantes Similares

Para definir o valor de cada oferta b_{ij} de um anunciante i por uma consulta j sorteamos um valor inteiro aleatório no intervalo $[b_{min}, b_{max}]$. Assim, aproximando os valores máximo e mínimo deixamos os anunciantes com um perfil de oferta mais parecido e a escolha de um ou outro para uma determinada consulta é menos impactante para a receita total no que diz respeito ao valor de cada exibição. Por outro lado, quando afastamos estes limites, tornamos as diferenças de escolha mais significativas.

5.2.3.2

Geração de Ofertas para Anunciantes Diferenciados

Neste cenário, consideramos que os grandes anunciantes pagam menos por cada consulta e os pequenos anunciantes pagam mais.

Para fazermos esta distribuição, consideramos que o menor anunciante pagará aproximadamente b_{max} por cada consulta e o maior anunciante pagará aproximadamente b_{min} . Entre estes dois limites decrescemos o valor da oferta

do menor para o maior anunciante através de uma distribuição de Zipf de potência 0,5.

Para que cada anunciante não faça a mesma oferta para todas as consultas, provocamos variações para cima de até 10% da diferença $b_{max} - b_{min}$ sobre o valor da cada oferta.

Por exemplo, se definimos $b_{min} = 10$ e $b_{max} = 30$, as ofertas do maior anunciante serão sorteadas entre 10 e 12, aleatoriamente, para cada consulta.

5.3 Medição do Desempenho

Os algoritmos para seleção de links foram desenvolvidos de modo que, ao serem executados, retornem o número de atribuições e a receita total obtida, além do arquivo de atribuições que determina qual consulta foi atribuída a cada anunciante.

Para avaliarmos o desempenho de um algoritmo em determinado cenário geramos primeiramente um arquivo de anunciantes e limites (LIM) e um arquivo de ofertas (OFT) que servirão para todos os testes deste cenário. Depois geramos um arquivo de consultas em determinada distribuição para servir como padrão histórico (CON).

Sobre estes arquivos executamos um algoritmo para calcular a solução ótima e gerar um arquivo de atribuições (OTM). Este arquivo será usado para montagem de uma tabela similar a Tabela 4.7, que é a base para execução dos algoritmos preditivos determinístico e aleatório.

Após esta fase são gerados dez arquivos de consultas (CON). Para cada um destes arquivos calculamos a solução ótima e executamos os algoritmos online de seleção de links. Como nosso objetivo é maximizar a receita, medimos o desempenho dos algoritmos em cada experimento comparando a receita obtida por suas atribuições com a receita da solução ótima. Este percentual representa o desempenho do algoritmo em um dia de avaliação. Na Tabela 5.4 apresentamos um exemplo de medição com os algoritmos Guloso e AdWords para quatro dias de processamento de consultas. Para cada algoritmo são apresentadas, além da coluna de desempenho (Desemp), uma coluna com a quantidade de anunciantes exibidos (Exibe) e outra com a receita de cada algoritmo.

Após a simulação de diversos dias de processamento, calculamos a média aritmética (\bar{x}) e o desvio padrão (σ) do desempenho, como vemos na Tabela 5.4. Estes são os valores que usamos como medida de desempenho nas tabelas de avaliação que apresentamos para cada cenário e suas variações.

	Solução Ótima		Algoritmo Guloso			Algoritmo AdWords		
	Exibe	Receita	Desemp	Exibe	Receita	Desemp	Exibe	Receita
1	451	16664	87,8%	451	14632	88,2%	458	14694
2	458	17019	89,0%	464	15154	88,9%	471	15132
3	467	17119	87,4%	448	14962	88,2%	461	15096
4	456	16756	87,2%	445	14612	88,7%	459	14869
\bar{x}	458	16890	87,9%	452	14840	88,5%	462	14948
σ	7	215	0,8%	8	264	0,4%	6	205

Tabela 5.4: Exemplo de experimentos e medição de um cenário

Quando mudamos a potência da distribuição de consultas para um mesmo cenário, repetimos o processo: geramos um novo arquivo de histórico de consultas (CON), encontramos um novo arquivo de atribuições da solução ótima (OTM), geramos dez arquivos de consultas diárias e executamos as simulações.

5.4

Avaliação de Cenários com Anunciantes Similares

Analisamos nesta seção, cenários com anunciantes similares, isto é, anunciantes que possuem limites de exibição entre L_{min} e L_{max} , sorteados aleatoriamente com a mesma probabilidade, e ofertas entre b_{min} e b_{max} , com o mesmo tipo de sorteio.

Dividimos nossa análise nos cinco cenários definidos na Tabela 5.5, utilizando a razão m/Q_{max} , onde m é o número de consultas no dia e Q_{max} é o número máximo de consultas que pode ser atendido em um dia.

Cenário	m	Q_{max}	m/Q_{max}
1	500	2000	1/4
2	1000	2000	1/2
3	2000	2000	1
4	4000	2000	2
5	8000	2000	4

Tabela 5.5: Cenários

Em todos os cenários avaliados, mantivemos alguns parâmetros fixos e variamos outros conforme descrito na Tabela 5.6.

Parâmetros Fixos		
n	Anunciantes	50
Q	Tipos de consulta	20
T	Anunciantes exibidos por consulta	1
L_{min}	Limite mínimo de exibições por anunciante	30
L_{max}	Limite máximo de exibições por anunciante	50
b_{min}	Oferta mínima	10
Parâmetros Variáveis		
b_{max}	Oferta máxima	
	- pouca variação de ofertas	20
	- muita variação de ofertas	70
Q_a	Tipos de consulta que um anunciante faz oferta	
	- pouca competição entre os anunciantes	≤ 5
	- muita competição entre os anunciantes	≥ 15

Tabela 5.6: Parâmetros fixos e variáveis para anunciantes similares

Em todos os cenários discutidos nesta seção os anunciantes têm perfis semelhantes em relação ao limite de exibições, sempre sorteado entre 30 e 50.

As m consultas são geradas por distribuições de Zipf, com cinco potências distintas, conforme explicado na seção 5.2.1. O objetivo é avaliarmos desde situações onde todas as consultas têm alguma representatividade (potência 1) até aquelas onde temos uma grande concentração nas consultas que aparecem mais (potência 2).

Os valores de Q_{max} apresentados para definir os cenários são aproximados, pois dependem de sorteio no momento da geração dos dados, como explicado na Seção 5.2.2.

Não variamos T , pois isto seria similar a aumentar o número de consultas recebidos, o que descaracterizaria o cenário.

Quando utilizamos $b_{max} = 20$, buscamos simular cenários onde os anunciantes têm perfis de ofertas mais parecidos e a seleção de um ou outro anunciante para determinada consulta não representa um ganho diferenciado tão grande para o algoritmo. Quando aumentamos para $b_{max} = 70$ tornamos as ofertas mais heterogêneas e a seleção de um ou outro anunciante tem maior impacto na receita final.

Com a variação do número de consultas em que cada anunciante está interessado (Q_a), simulamos situações onde temos pequena competitividade entre os anunciantes ($Q_a \leq 5$) e ambientes altamente competitivos ($Q_a \geq 15$), onde os anunciantes estão interessados em quase todos os tipos de consulta.

5.4.1 Cenário 1

O primeiro cenário que analisamos contempla os casos onde a quantidade diária de consultas ($m = 500$) é muito menor que o limite total de exposições dos anunciantes ($Q_{max} = 2000$).

Uma característica observada em todas as simulações deste cenário é que, na solução ótima, todas as consultas são atribuídas a algum anunciante.

A Tabela 5.7 apresenta os resultados obtidos pelos algoritmos quando as ofertas variam em uma faixa pequena, entre 10 e 20, e cada anunciante faz ofertas por no máximo 5 tipos de consultas diferentes. O excelente desempenho do algoritmo Guloso, como podemos observar na tabela, deve-se ao fato de que, quando m é bem menor que Q_{max} , existe a tendência de que poucos anunciantes atinjam seu limite de exposições e assim escolher a melhor oferta é quase sempre a melhor opção. Podemos observar que o algoritmo Aleatório acompanha o resultado do algoritmo Guloso, pois a solução ótima tende a concentrar cada tipo de consulta em um ou poucos anunciantes e o sorteio por conseguinte será mais restrito.

Os algoritmos, AdWords e Primal-Dual, procuram distribuir as consultas pelos anunciantes para não perder oportunidades quando estes tiverem seus limites totalmente utilizados. Entretanto, quando m é bem menor que Q_{max} , esta economia não é necessária e acabam com um desempenho mais baixo devido a escolha de ofertas menores no início do dia, .

O desempenho do algoritmo Preditivo neste cenário fica abaixo dos outros algoritmos, porque este não atende as consultas que não estejam previstas na solução ótima. Assim, qualquer diferença na predição, não é recuperada mais à frente.

$b_{min} = 10 ; b_{max} = 20; Q_a \leq 5;$								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
1.00	\bar{x}	500	9607	99.8%	94.4%	94.1%	92.2%	99.4%
	σ	0	37	0.0%	0.2%	0.1%	2.1%	0.2%
1.25	\bar{x}	500	9725	99.9%	95.5%	95.1%	94.4%	99.9%
	σ	0	18	0.1%	0.4%	0.3%	2.2%	0.1%
1.50	\bar{x}	500	9275	99.8%	94.5%	94.4%	95.3%	100.0%
	σ	0	22	0.1%	0.3%	0.2%	0.9%	0.0%
1.75	\bar{x}	500	9040	99.8%	93.6%	93.4%	94.9%	99.4%
	σ	0	72	0.1%	0.1%	0.2%	1.9%	0.4%
2.00	\bar{x}	500	7809	99.9%	96.2%	96.1%	93.8%	99.8%
	σ	0	51	0.1%	0.4%	0.2%	1.7%	0.1%

Tabela 5.7: Cenário 1: $b_{max} = 20; Q_a \leq 5$

Quando aumentamos o intervalo das ofertas ($b_{max} = 70$), não percebemos quase nenhuma mudança no desempenho relativo dos algoritmos, como vemos na Tabela 5.8, e os comentários feitos para a tabela anterior cabem perfeitamente para esta tabela.

$b_{min} = 10 ; b_{max} = 70; Q_a \leq 5;$								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
1.00	\bar{x}	500	31126	99.9%	95.1%	94.8%	93.8%	99.6%
	σ	0	63	0.0%	0.1%	0.1%	1.3%	0.3%
1.25	\bar{x}	500	30999	99.4%	94.3%	94.1%	94.7%	99.7%
	σ	0	301	0.3%	0.2%	0.3%	1.8%	0.2%
1.50	\bar{x}	500	27093	100.0%	94.7%	94.4%	95.3%	99.8%
	σ	0	588	0.0%	0.2%	0.1%	1.2%	0.1%
1.75	\bar{x}	500	28547	99.1%	93.6%	93.2%	95.2%	99.0%
	σ	0	166	0.3%	0.3%	0.4%	2.0%	0.8%
2.00	\bar{x}	500	29739	100.0%	92.8%	92.6%	97.1%	100.0%
	σ	0	181	0.0%	0.2%	0.2%	1.1%	0.0%

Tabela 5.8: Cenário 1: $b_{max} = 70; Q_a \leq 5$

Quando ampliamos o número de consultas em que cada anunciante está interessado ($Q_a \geq 15$), implica que aumentamos a competição entre os anunciantes pela mesma consulta e, por conseguinte, a probabilidade de termos

anunciantes com oferta máxima igual por cada consulta, já que as ofertas variam numa faixa pequena. Este cenário favorece ainda mais o algoritmo Guloso de encontrar para cada consulta um anunciante com oferta máxima sem o limite esgotado.

Podemos observar na Tabela 5.9 que os algoritmos Guloso e Aleatório continuam com o melhor desempenho entre todos.

A grande diferença para a situação descrita na Tabela 5.7 é a melhora de desempenho dos algoritmos AdWords e Primal-Dual. Como há uma maior possibilidade de distribuir as ofertas por vários anunciantes e estes têm valores de ofertas bem próximos, entre 10 e 20, a estratégia destes algoritmos de variar os anunciantes para economizar limites funciona melhor nesta situação.

Podemos observar ainda que o desempenho destes algoritmos decresce conforme a distribuição de Zipf concentra mais consultas em um único tipo, justamente porque distribuir muito as consultas pelos anunciantes não é mais tão interessante para atingir a solução ótima.

$b_{min} = 10 ; b_{max} = 20; Q_a \geq 15$								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
1.00	\bar{x}	500	9916	100.0%	97.2%	97.2%	93.4%	100.0%
	σ	0	17	0.0%	0.1%	0.1%	1.5%	0.0%
1.25	\bar{x}	500	9953	100.0%	97.5%	97.3%	92.7%	100.0%
	σ	0	11	0.0%	0.2%	0.2%	2.6%	0.0%
1.50	\bar{x}	500	9938	100.0%	96.8%	96.9%	95.8%	100.0%
	σ	0	9	0.0%	0.1%	0.1%	1.6%	0.0%
1.75	\bar{x}	500	9902	100.0%	95.1%	95.1%	96.4%	100.0%
	σ	0	5	0.0%	0.2%	0.2%	1.1%	0.0%
2.00	\bar{x}	500	9464	100.0%	95.5%	94.5%	95.4%	99.9%
	σ	0	32	0.0%	0.1%	0.1%	1.7%	0.1%

Tabela 5.9: Cenário 1: $b_{max} = 20; Q_a \geq 15$

Quando entretanto, mantendo $Q_a \geq 15$, aumentamos a variação nos valores das ofertas fazendo $b_{max} = 70$, cai um pouco o desempenho dos algoritmos AdWords e Primal-Dual, pois nem sempre é possível economizar os limites de exibições dos anunciantes sem perder oportunidades de melhorar a receita. Este resultado é apresentado na Tabela 5.10

$b_{min} = 10 ; b_{max} = 70; Q_a \geq 15$								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
1.00	\bar{x}	500	34568	99.9%	96.7%	96.6%	94.7%	100.0%
	σ	0	30	0.0%	0.1%	0.1%	0.8%	0.0%
1.25	\bar{x}	500	33103	99.9%	95.5%	95.3%	93.0%	100.0%
	σ	0	162	0.0%	0.1%	0.1%	2.0%	0.0%
1.50	\bar{x}	500	33968	99.9%	93.9%	93.9%	95.9%	99.9%
	σ	0	80	0.1%	0.2%	0.2%	1.4%	0.1%
1.75	\bar{x}	500	32887	99.9%	94.1%	94.0%	94.9%	99.9%
	σ	0	127	0.0%	0.3%	0.3%	2.2%	0.1%
2.00	\bar{x}	500	31966	100.0%	93.4%	93.1%	97.4%	100.0%
	σ	0	76	0.0%	0.2%	0.2%	1.0%	0.0%

Tabela 5.10: Cenário 1: $b_{max} = 70; Q_a \geq 15$

Da análise dos experimentos deste Cenário 1, onde o número de consultas é bem menor que o limite total de exibições dos anunciantes, concluímos que os algoritmos Guloso e Aleatório mantêm o melhor desempenho independente da variação dos outros parâmetros. Desta forma, por ser de simples implementação, o algoritmo Guloso se apresenta como o mais indicado se sabemos previamente que raramente os limites de exibição dos anunciantes serão atingidos.

5.4.2 Cenário 2

Neste segundo cenário analisamos os casos onde a quantidade diária de consultas ($m = 1000$) é menor que o limite total de exibições dos anunciantes ($Q_{max} = 2000$). Por não ser muito menor, já implica no esgotamento de alguns limites de exibição quando temos uma maior concentração de consultas de determinados tipos.

Os resultados apresentados na Tabela 5.11, quando há pouca variação nas ofertas e pouca concorrência entre os anunciantes, são muito parecidos com os resultados exibidos na Tabela 5.7, do Cenário 1. A grande diferença é que o desempenho do algoritmo Guloso não é tão constante em todas as situações. Nas distribuições com potências de 1.50 e 1.75, onde existe uma maior concentração de consultas e a solução ótima atende todas ou quase todas as consultas, o algoritmo Guloso apresenta um desempenho mais baixo, pois esgota o limite de alguns anunciantes. Estas distribuições também apresentam

um desvio padrão maior, que mostra como o algoritmo variou mais devido a peculiaridades de uma ou outra sequência. Na distribuição de potência 2, embora exista uma concentração ainda maior de consultas, a solução ótima atende muito menos consultas e logo o algoritmo Guloso volta a ter um bom desempenho.

O algoritmo Aleatório, diferentemente do Guloso, mantém um bom rendimento em todas as potências, pois sua característica aleatória elimina as particularidades de uma dada sequência ou distribuição.

Os algoritmos, Primal-Dual e AdWords, têm um desempenho um pouco melhor que na mesma configuração do Cenário 1, pois com mais consultas, a perda de receita no início para preservar saldo nos limites dos anunciantes, tem maior possibilidade de ser recuperada mais à frente.

$b_{min} = 10 ; b_{max} = 20; Q_a \leq 5$								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
1.00	\bar{x}	1000	17937	99.6%	95.2%	95.1%	95.4%	99.6%
	σ	0	51	0.1%	0.1%	0.1%	1.0%	0.2%
1.25	\bar{x}	1000	18774	99.6%	95.8%	95.6%	94.5%	99.9%
	σ	0	91	0.1%	0.1%	0.2%	0.5%	0.1%
1.50	\bar{x}	1000	16657	97.8%	95.7%	95.3%	95.1%	99.7%
	σ	0	272	1.5%	0.6%	0.7%	1.6%	0.2%
1.75	\bar{x}	975	15963	95.2%	96.0%	96.3%	96.2%	99.3%
	σ	19	329	0.7%	0.3%	0.3%	1.9%	0.2%
2.00	\bar{x}	684	12258	99.1%	96.7%	96.5%	95.4%	99.7%
	σ	25	427	0.2%	0.4%	0.6%	2.0%	0.2%

Tabela 5.11: Cenário 2: $b_{max} = 20; Q_a \leq 5$

Quando aumentamos a variação das ofertas, não temos nenhum grande efeito no desempenho dos algoritmos, como podemos ver na Tabela 5.12. A diminuição do desvio padrão no desempenho do algoritmo Guloso mostra que este foi menos afetado por particularidades de alguma sequência específica. A variação de desempenho dos algoritmos, Primal-Dual e AdWords, em relação às distribuições, deve-se a maior variação entre os valores das ofertas. O desempenho destes é melhor onde as consultas ficam mais concentradas, pois têm menos opções de anunciantes para distribuir suas escolhas.

$b_{min} = 10 ; b_{max} = 70; Q_a \leq 5$								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
1.00	\bar{x}	1000	60505	99.0%	94.1%	93.9%	94.8%	99.7%
	σ	0	301	0.1%	0.2%	0.2%	1.4%	0.1%
1.25	\bar{x}	1000	56491	98.7%	94.6%	94.4%	91.8%	98.0%
	σ	0	1182	0.2%	1.1%	1.1%	2.7%	1.1%
1.50	\bar{x}	967	51367	98.2%	95.9%	95.9%	97.3%	99.6%
	σ	26	1571	0.2%	0.5%	0.5%	1.7%	0.2%
1.75	\bar{x}	928	47600	97.7%	95.4%	95.3%	97.2%	99.6%
	σ	21	1098	0.3%	0.1%	0.1%	1.2%	0.4%
2.00	\bar{x}	962	45791	99.5%	97.4%	97.3%	96.8%	99.9%
	σ	15	845	0.1%	0.1%	0.1%	1.5%	0.1%

Tabela 5.12: Cenário 2: $b_{max} = 70; Q_a \leq 5$

Quando aumentamos o número de ofertas por anunciante, o algoritmo Guloso volta a ter um desempenho muito bom, similar ao obtido no Cenário 1, como comprovam os resultados da Tabela 5.13, pois todas as consultas voltam a ser atendidas na solução ótima. Este resultado se repete mesmo quando ampliamos o intervalo de ofertas como na Tabela 5.14.

Os algoritmos, AdWords e Primal-Dual, melhoram seu desempenho quando as consultas são mais distribuídas (potência menor), pois têm melhores opções de distribuição pelos anunciantes. Esta vantagem é ainda maior quando temos um intervalo menor de ofertas e a possibilidade de termos vários anunciantes com a mesma oferta é maior. Este comportamento é similar ao observado no Cenário 1.

$b_{min} = 10 ; b_{max} = 20; Q_a \geq 15$								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
1.00	\bar{x}	1000	19963	100.0%	97.9%	97.9%	95.2%	100.0%
	σ	0	7	0.0%	0.1%	0.1%	1.5%	0.0%
1.25	\bar{x}	1000	19761	99.7%	96.4%	96.3%	94.5%	99.9%
	σ	0	46	0.1%	0.2%	0.2%	1.2%	0.1%
1.50	\bar{x}	1000	19376	99.9%	95.5%	95.8%	95.4%	99.9%
	σ	0	30	0.0%	0.1%	0.1%	0.3%	0.0%
1.75	\bar{x}	1000	19385	99.9%	94.6%	95.0%	95.5%	100.0%
	σ	0	80	0.0%	0.1%	0.1%	1.0%	0.1%
2.00	\bar{x}	1000	18706	99.7%	94.5%	94.2%	95.2%	99.9%
	σ	0	58	0.0%	0.1%	0.1%	1.3%	0.0%

Tabela 5.13: Cenário 2: $b_{max} = 20; Q_a \geq 15$

$b_{min} = 10 ; b_{max} = 70; Q_a \geq 15$								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
1.00	\bar{x}	1000	67540	99.4%	96.1%	96.0%	94.7%	99.9%
	σ	0	266	0.1%	0.1%	0.1%	0.9%	0.0%
1.25	\bar{x}	1000	65306	99.3%	95.0%	94.9%	95.3%	99.9%
	σ	0	258	0.1%	0.1%	0.2%	1.6%	0.1%
1.50	\bar{x}	1000	65246	99.1%	95.5%	95.4%	95.2%	99.9%
	σ	0	322	0.1%	0.3%	0.2%	1.5%	0.1%
1.75	\bar{x}	1000	59594	99.4%	92.7%	92.7%	96.5%	99.9%
	σ	0	387	0.1%	0.1%	0.2%	0.5%	0.1%
2.00	\bar{x}	1000	57270	99.7%	94.2%	93.9%	95.8%	99.7%
	σ	0	404	0.0%	0.1%	0.1%	2.0%	0.2%

Tabela 5.14: Cenário 2: $b_{max} = 70; Q_a \geq 15$

Neste segundo cenário, o melhor e mais constante desempenho é o do algoritmo Aleatório que atinge quase sempre acima de 98% da solução ótima, independente das variações nas ofertas, da competição entre os anunciantes e das peculiaridades de uma distribuição ou de uma sequência. O algoritmo Guloso também tem desempenho muito bom, mas é afetado por particularidades de algumas distribuições. Os outros algoritmos, apesar de não alcançarem o mesmo nível de desempenho destes dois, também têm um bom desempenho.

5.4.3 Cenário 3

No Cenário 3 simulamos os casos onde a quantidade diária de consultas é igual ao limite total de exibições dos anunciantes ($m = Q_{max}$). Neste cenário observamos um desempenho dos algoritmos um pouco diferente dos Cenários 1 e 2.

Como nos cenários anteriores, iniciamos nossa análise com experimentos com pouca variação nas ofertas e pouca competição entre os anunciantes. Na Tabela 5.15, podemos observar uma considerável queda de desempenho do algoritmo Guloso quando temos uma distribuição de Zipf de potência mais baixa. Na primeira distribuição, por exemplo, para se atingir o resultado da solução ótima, observamos que quase todas as consultas (1931) devem ser atendidas. Como $m = Q_{max}$, significa que quase todos os anunciantes terão seus limites esgotados. Pela natureza de sua estratégia, o algoritmo Guloso gera um rápido esgotamento dos limites de alguns anunciantes e a queda da receita está diretamente ligada a atender menos consultas. Não apresentamos na Tabela 5.15, mas a média de atendimento do algoritmo Guloso é de 1757 consultas, aproximadamente 9% abaixo do ideal, o que implica numa diminuição de 6% na receita.

Os algoritmos, AdWords e Primal-Dual, têm um desempenho melhor nesta situação, pois economizam os limites dos anunciantes e praticamente não são afetados pelas diferentes distribuições.

Quando as potências aumentam e com elas a concentração das consultas em determinados tipos, o esgotamento ocorre praticamente da mesma forma para todos os algoritmos, inclusive para a solução ótima, pois há menos opções de escolha. Neste caso o algoritmo Guloso volta a ter um bom desempenho.

O algoritmo Aleatório continua com o melhor desempenho, pois o ideal é atendermos a todas as consultas, já que o número total de consultas não ultrapassa o limite máximo de exibições.

Pela primeira vez em nossos experimentos o algoritmo Preditivo torna-se competitivo em relação aos outros algoritmos. No Cenário 4 discutiremos melhor as características de seu desempenho.

$b_{min} = 10 ; b_{max} = 20; Q_a \leq 5$								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
1.00	\bar{x}	1931	32243	94.1%	95.8%	96.1%	96.6%	98.4%
	σ	33	491	0.4%	0.4%	0.4%	0.8%	0.4%
1.25	\bar{x}	1478	23909	97.1%	96.6%	96.5%	94.5%	98.8%
	σ	37	398	1.0%	0.3%	0.3%	1.2%	1.1%
1.50	\bar{x}	1529	24946	97.3%	97.3%	97.2%	95.0%	98.9%
	σ	45	547	1.0%	0.3%	0.3%	1.7%	0.6%
1.75	\bar{x}	1391	22516	98.3%	97.2%	97.2%	98.4%	99.5%
	σ	23	334	0.3%	0.2%	0.2%	0.8%	0.2%
2.00	\bar{x}	987	16334	99.1%	96.4%	96.0%	98.1%	99.6%
	σ	38	534	0.3%	0.4%	0.3%	1.5%	0.4%

Tabela 5.15: Cenário 3: $b_{max} = 20; Q_a \leq 5$

Ao aumentarmos a faixa das ofertas obtemos um resultado próximo ao cenário anterior, mas na maioria dos casos um pouco abaixo. Apenas o algoritmo Guloso melhora na potência mais baixa, pois menos consultas devem ser atendidas para se atingir a solução ótima.

$b_{min} = 10 ; b_{max} = 70; Q_a \leq 5$								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
1.00	\bar{x}	1785	97149	95.3%	95.2%	95.1%	94.8%	98.4%
	σ	21	1086	0.3%	0.2%	0.2%	1.1%	0.4%
1.25	\bar{x}	1588	80933	96.1%	95.8%	95.6%	97.4%	98.1%
	σ	35	1199	0.9%	0.3%	0.3%	0.5%	0.5%
1.50	\bar{x}	1412	72816	97.8%	96.3%	96.3%	96.3%	99.4%
	σ	18	855	0.2%	0.3%	0.3%	0.8%	0.3%
1.75	\bar{x}	1175	61081	99.0%	94.7%	94.6%	94.4%	97.8%
	σ	42	1667	0.7%	0.5%	0.5%	1.6%	0.8%
2.00	\bar{x}	966	47367	97.3%	95.5%	95.4%	96.3%	99.6%
	σ	30	1253	0.7%	0.6%	0.6%	1.3%	0.2%

Tabela 5.16: Cenário 3: $b_{max} = 70; Q_a \leq 5$

Ao gerarmos um cenário com maior competição entre os anunciantes, fazendo $Q_a \geq 15$, ocorre que todas ou quase todas as ofertas podem novamente ser atendidas como nos cenários 1 e 2. A diferença é que neste caso, como

o número de consultas ficará muito próximo ao limite total de exibições, é necessário que se faça uma distribuição adequada por todos os anunciantes para que tenhamos uma boa disponibilidade na maior parte do tempo.

Este é o cenário ideal para um bom desempenho dos algoritmos AdWords e Primal-Dual, como vemos na Tabela 5.17. O algoritmo Guloso também apresenta um bom desempenho devido a pouca variação de ofertas, mas abaixo dos outros dois, pois esgota mais rapidamente o limite dos anunciantes com maiores ofertas. Na Tabela 5.18, quando aumentamos a variação das ofertas, o algoritmo Guloso é mais prejudicado.

Em ambas as situações, o algoritmo Aleatório continua sendo o de melhor desempenho entre todos.

$b_{min} = 10 ; b_{max} = 20 ; Q_a \geq 15$								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
1.00	\bar{x}	1998	38367	98.2%	99.2%	99.3%	94.3%	99.6%
	σ	0	96	0.1%	0.1%	0.0%	1.4%	0.1%
1.25	\bar{x}	2000	38091	98.5%	99.2%	99.3%	94.9%	99.4%
	σ	0	142	0.1%	0.1%	0.1%	1.7%	0.3%
1.50	\bar{x}	2000	37064	96.9%	98.8%	98.9%	95.9%	99.6%
	σ	0	172	0.3%	0.1%	0.1%	0.7%	0.2%
1.75	\bar{x}	1980	35131	96.7%	98.6%	98.8%	96.4%	99.2%
	σ	0	181	0.2%	0.1%	0.1%	1.1%	0.4%
2.00	\bar{x}	2000	35468	97.4%	98.5%	98.5%	96.5%	99.5%
	σ	0	213	0.3%	0.1%	0.2%	1.1%	0.2%

Tabela 5.17: Cenário 3: $b_{max} = 20; Q_a \geq 15$

$b_{min} = 10 ; b_{max} = 70; Q_a \geq 15$								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
1.00	\bar{x}	2000	130233	96.9%	98.6%	98.6%	92.8%	99.1%
	σ	0	546	0.1%	0.2%	0.2%	2.4%	0.4%
1.25	\bar{x}	1970	118512	97.2%	98.7%	98.7%	94.9%	98.7%
	σ	0	1555	0.2%	0.2%	0.1%	1.7%	0.3%
1.50	\bar{x}	1980	118184	94.5%	98.0%	98.0%	95.8%	99.1%
	σ	0	1077	0.3%	0.2%	0.2%	0.9%	0.2%
1.75	\bar{x}	2000	111408	97.5%	98.8%	98.8%	94.6%	99.1%
	σ	0	933	0.2%	0.1%	0.1%	1.2%	0.2%
2.00	\bar{x}	1975	97495	96.2%	97.7%	97.7%	97.4%	99.2%
	σ	4	1146	0.4%	0.3%	0.3%	0.5%	0.2%

Tabela 5.18: Cenário 3: $b_{max} = 70; Q_a \geq 15$

5.4.4 Cenário 4

No Cenário 4, simulamos dias onde a quantidade diária de consultas ($m = 4000$) é superior ao limite total de exibições dos anunciantes ($Q_{max} = 2000$). Neste cenário, o desempenho dos algoritmos é bem diferente do encontrado nos cenários anteriores.

A principal característica deste cenário é que a solução ótima atinge no máximo Q_{max} exibições e logo $m - Q_{max}$ consultas devem ser deixadas de lado. Em nosso ambiente, excetuando o algoritmo Preditivo, todos os outros sempre tentam atender a uma nova consulta atribuindo a esta algum anunciante. Esta consulta só não é atendida se não existir nenhuma oferta para ela ou se todos os anunciantes com ofertas para esta consulta não tiverem mais disponibilidade de exibição. O algoritmo Preditivo, por outro lado, deixa de atender consultas que não estão previstas na solução ótima, na expectativa de atender outras consultas de maior receita mais à frente. Esta capacidade de abdicar de atender consultas é que o torna mais competitivo quando a quantidade diária de consultas ultrapassa o limite total de exibições.

Quando analisamos a Tabela 5.19, com pouca variação nas ofertas e pouca competição entre os anunciantes, observamos a superioridade do algoritmo Preditivo sobre os outros algoritmos nas distribuições com menor concentração de consultas. Conforme aumentamos a concentração, os desempenhos ficam mais próximos, pois não existem muitas opções diferentes de anunciantes para

atender às consultas e mesmo a solução ótima atende bem menos consultas que Q_{max} . Para uma potência de 2, o número de consultas atendidas fica 20% abaixo da capacidade total.

$b_{min} = 10 ; b_{max} = 20; Q_a \leq 5$								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
1.00	\bar{x}	1956	35621	92.8%	91.8%	91.8%	97.1%	92.5%
	σ	0	65	0.2%	0.3%	0.3%	1.1%	0.3%
1.25	\bar{x}	1967	34445	94.4%	94.3%	94.2%	97.2%	95.5%
	σ	0	80	0.4%	0.4%	0.4%	0.4%	0.5%
1.50	\bar{x}	1958	31776	93.9%	95.2%	94.9%	97.5%	96.5%
	σ	0	234	0.5%	0.3%	0.3%	0.7%	0.4%
1.75	\bar{x}	1837	28928	95.7%	97.1%	97.2%	97.9%	99.2%
	σ	30	356	0.4%	0.3%	0.3%	0.8%	0.2%
2.00	\bar{x}	1594	24644	97.0%	97.2%	97.1%	96.3%	98.4%
	σ	13	175	0.6%	0.2%	0.3%	0.8%	0.5%

Tabela 5.19: Cenário 4: $b_{max} = 20; Q_a \leq 5$

Esta superioridade do algoritmo Preditivo fica mais acentuada quando ampliamos o intervalo das ofertas como vemos nos resultados apresentados na Tabela 5.20. Enquanto este algoritmo mantém o mesmo desempenho da situação anterior, os outros caem de rendimento, pois passam a atender consultas de receita mais baixa, visto que não deixam de atender se houver disponibilidade de anunciante.

$b_{min} = 10 ; b_{max} = 70; Q_a \leq 5$								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
1.00	\bar{x}	2045	113804	87.9%	87.8%	87.7%	97.2%	89.5%
	σ	0	367	0.7%	0.9%	0.9%	0.8%	0.7%
1.25	\bar{x}	1981	110488	90.4%	90.5%	90.5%	97.5%	91.9%
	σ	0	563	0.9%	0.4%	0.5%	0.6%	0.4%
1.50	\bar{x}	1905	91969	92.9%	93.1%	93.1%	97.7%	93.8%
	σ	10	828	0.7%	0.7%	0.7%	0.5%	0.4%
1.75	\bar{x}	1620	81189	96.4%	94.3%	94.2%	96.0%	95.8%
	σ	26	949	0.4%	0.2%	0.2%	1.1%	0.7%
2.00	\bar{x}	1611	75167	98.6%	96.9%	96.8%	95.3%	98.7%
	σ	9	850	0.2%	0.2%	0.2%	0.8%	0.3%

Tabela 5.20: Cenário 4: $b_{max} = 70; Q_a \leq 5$

Na Tabela 5.21, ao tornarmos o cenário mais competitivo ($Q_a \geq 15$), os algoritmos passam a ter um rendimento melhor para as distribuições de potência mais baixa. Isto ocorre pois é possível atender a todas as consultas independente da concentração. Assim, como nas distribuições de potência mais alta, chegam mais consultas repetidas, os algoritmos acabam atribuindo mesmo quando os anunciantes não têm uma boa oferta para aquele tipo de consulta, em lugar de aguardar por uma melhor oportunidade.

$b_{min} = 10 ; b_{max} = 20; Q_a \geq 15$								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
1.00	\bar{x}	1984	38938	95.5%	96.1%	96.3%	99.1%	95.3%
	σ	0	12	0.2%	0.2%	0.2%	0.6%	0.4%
1.25	\bar{x}	2019	39294	93.1%	93.5%	93.7%	96.6%	92.7%
	σ	0	53	0.6%	0.6%	0.6%	0.5%	0.6%
1.50	\bar{x}	1996	38673	92.0%	92.5%	92.7%	96.8%	91.9%
	σ	0	39	0.2%	0.2%	0.2%	0.7%	0.2%
1.75	\bar{x}	1993	38032	91.7%	92.0%	92.1%	95.6%	91.6%
	σ	0	104	0.3%	0.3%	0.4%	1.3%	0.2%
2.00	\bar{x}	2028	37332	91.6%	91.8%	92.0%	96.5%	92.1%
	σ	0	62	0.3%	0.4%	0.4%	0.6%	0.3%

Tabela 5.21: Cenário 4: $b_{max} = 20; Q_a \geq 15$

Na Tabela 5.22, com a ampliação do intervalo de ofertas, a queda de rendimento dos algoritmos nas distribuições de potências mais altas fica ainda mais acentuada. Novamente o algoritmo Preditivo não sofre esta variação devido a sua estratégia de abandonar consultas não previstas na solução ótima.

$b_{min} = 10 ; b_{max} = 70; Q_a \geq 15$								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
1.00	\bar{x}	2041	134253	93.2%	95.2%	95.2%	98.5%	92.5%
	σ	0	51	0.4%	0.3%	0.3%	0.1%	0.2%
1.25	\bar{x}	2055	136504	89.0%	92.1%	92.2%	94.7%	90.4%
	σ	0	177	0.4%	0.4%	0.4%	1.3%	0.5%
1.50	\bar{x}	1993	128974	88.0%	90.1%	90.1%	96.0%	87.7%
	σ	0	390	0.5%	0.3%	0.3%	0.9%	0.4%
1.75	\bar{x}	1977	126173	89.1%	89.7%	89.7%	97.1%	88.1%
	σ	0	341	0.6%	0.8%	0.8%	1.1%	0.8%
2.00	\bar{x}	1989	117654	83.3%	83.2%	83.2%	96.5%	83.0%
	σ	0	594	0.4%	0.8%	0.9%	2.0%	0.5%

Tabela 5.22: Cenário 4: $b_{max} = 70; Q_a \geq 15$

Em resumo, o Cenário 4 aponta para uma tendência de que quando o número de consultas do dia (m) passa a ser razoavelmente maior que o limite total de exibições (Q_{max}), o algoritmo Preditivo torna-se a melhor escolha, pelo bom desempenho obtido em todas as distribuições verificadas. Em boa parte das simulações, este algoritmo foi bem superior aos outros e quando não foi superior, a diferença para o de melhor desempenho foi pequena.

5.4.5 Cenário 5

No Cenário 5 analisamos experimentos com uma quantidade diária de consultas ($m = 8000$) bem maior que o limite total de exibições dos anunciantes ($Q_{max} = 2000$), aumentando a razão m/Q_{max} em relação ao Cenário 4.

Este aumento melhorou o desempenho do algoritmo Preditivo em todas as distribuições estudadas. Como a quantidade de consultas recebidas neste cenário é bem maior que a quantidade que será atribuída a algum anunciante, a probabilidade de encontrarmos consultas que constam na previsão da solução ótima é ainda maior que no cenário anterior. Por isso, nos casos com pouca concentração de consultas, este algoritmo aproxima-se de 100% de desempenho,

como vemos nas distribuições de baixa concentração de consultas (potência 1) das Tabelas 5.23 e 5.24.

$b_{min} = 10 ; b_{max} = 20; Q_a \leq 5$								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
1.00	\bar{x}	1956	36342	89.5%	88.6%	88.4%	99.2%	89.3%
	σ	0	19	0.5%	0.6%	0.5%	0.6%	0.6%
1.25	\bar{x}	1968	35556	91.3%	89.8%	89.7%	98.7%	90.6%
	σ	0	60	0.4%	0.5%	0.6%	0.6%	0.7%
1.50	\bar{x}	1964	35715	89.4%	88.8%	88.8%	97.5%	89.8%
	σ	0	76	0.1%	0.3%	0.3%	1.0%	0.3%
1.75	\bar{x}	2006	35015	91.8%	90.1%	90.0%	96.1%	91.3%
	σ	0	91	0.2%	0.1%	0.2%	1.6%	0.2%
2.00	\bar{x}	1914	31863	96.4%	96.5%	96.5%	96.5%	97.9%
	σ	25	289	0.5%	0.3%	0.2%	1.0%	0.5%

Tabela 5.23: Cenário 5: $b_{max} = 20; Q_a \leq 5$

$b_{min} = 10 ; b_{max} = 70; Q_a \leq 5$								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
1.00	\bar{x}	2028	120881	89.8%	89.1%	88.9%	99.5%	89.8%
	σ	0	212	0.7%	0.4%	0.4%	0.4%	0.5%
1.25	\bar{x}	1935	114613	85.7%	83.7%	83.5%	99.1%	83.8%
	σ	0	155	0.5%	0.5%	0.4%	0.5%	0.6%
1.50	\bar{x}	2019	108897	82.3%	83.2%	83.1%	97.4%	83.9%
	σ	0	219	0.6%	0.6%	0.6%	0.5%	0.6%
1.75	\bar{x}	1976	104939	89.1%	88.7%	88.7%	97.3%	89.6%
	σ	2	535	0.4%	0.3%	0.2%	0.8%	0.2%
2.00	\bar{x}	1922	97694	95.1%	94.7%	94.7%	97.2%	96.1%
	σ	30	1593	0.7%	0.6%	0.7%	0.5%	0.5%

Tabela 5.24: Cenário 5: $b_{max} = 70; Q_a \leq 5$

Por outro lado, os outros algoritmos tiveram um desempenho pior, pois a probabilidade de perder boas oportunidades é muito maior, visto que chegam ainda muitas consultas após os anunciantes terem o limite de exibições esgotado.

Os comentários feitos para o Cenário 4 continuam válidos para as tabelas dos experimentos deste cenário.

$b_{min} = 10 ; b_{max} = 20; Q_a \geq 15$								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
1.00	\bar{x}	2001	39521	94.9%	95.7%	95.9%	98.7%	94.8%
	σ	0	7	0.3%	0.2%	0.1%	1.0%	0.3%
1.25	\bar{x}	1992	39078	92.0%	92.2%	92.4%	99.0%	91.0%
	σ	0	23	0.2%	0.3%	0.3%	0.3%	0.3%
1.50	\bar{x}	1985	38800	90.4%	91.2%	91.3%	98.3%	90.1%
	σ	0	29	0.5%	0.7%	0.6%	0.8%	0.4%
1.75	\bar{x}	2091	40689	93.4%	93.9%	94.1%	98.2%	92.8%
	σ	0	24	0.5%	0.5%	0.5%	0.7%	0.6%
2.00	\bar{x}	1974	38128	88.3%	87.4%	87.5%	97.3%	87.9%
	σ	0	35	0.3%	0.5%	0.6%	0.5%	0.4%

Tabela 5.25: Cenário 5: $b_{max} = 20; Q_a \geq 15$

$b_{min} = 10 ; b_{max} = 70; Q_a \geq 15$								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
1.00	\bar{x}	2002	133439	92.5%	93.7%	93.7%	98.6%	92.5%
	σ	0	37	0.4%	0.4%	0.4%	0.6%	0.5%
1.25	\bar{x}	2101	140583	87.4%	89.3%	89.3%	96.7%	86.6%
	σ	0	48	0.5%	0.5%	0.4%	1.1%	0.5%
1.50	\bar{x}	1945	129854	84.4%	87.7%	87.7%	98.9%	86.0%
	σ	0	112	0.7%	0.8%	0.8%	0.2%	0.8%
1.75	\bar{x}	2006	130640	81.1%	79.9%	79.9%	97.3%	80.0%
	σ	0	226	0.3%	0.5%	0.6%	0.8%	0.5%
2.00	\bar{x}	2081	132333	83.1%	82.0%	81.9%	97.0%	82.2%
	σ	0	224	1.1%	1.4%	1.4%	0.9%	0.8%

Tabela 5.26: Cenário 5: $b_{max} = 70; Q_a \geq 15$

Efetamos ainda testes com um sexto cenário, com 16.000 consultas, e o desempenho dos algoritmos confirmou a superioridade do algoritmo Preditivo sobre os outros algoritmos, já vista nos Cenários 4 e 5, quando temos uma quantidade de consultas superior a que pode ser atendida ($m > Q_{max}$). Os

outros algoritmos diminuíram seu desempenho em alguns casos e aumentaram em outros, mas de uma forma geral ficaram entre 80% e 90% da solução ótima.

5.5

Análise Gráfica dos Algoritmos no Tempo

Nesta seção, comparamos graficamente o desempenho dos algoritmos ao longo do dia, representando como evolui a receita acumulada pelo algoritmo (eixo y) em função das consultas (eixo x). Nos gráficos, incluímos também uma solução ótima, que não é necessariamente única, mas que ilustra um caminho possível para a melhor solução.

Para a montagem dos gráficos não utilizamos nenhum tipo de média, mas valores absolutos de uma execução completa, isto é, para um determinado dia de processamento, coletamos os resultados de todos os algoritmos registrados nos arquivos de atribuição (ATR), planilhamos os resultados e destes resultados geramos os gráficos.

A Tabela 5.27 apresenta os parâmetros utilizados na geração dos gráficos, que foram extraídos dos cenários da Seção 5.4.

Cenário 1										
Gráfico	n	m	q	T	L_{min}	L_{max}	Q_a	b_{min}	b_{max}	Zipf
01	50	500	20	1	30	50	≤ 5	10	70	1,75
Cenário 2										
Gráfico	n	m	q	T	L_{min}	L_{max}	Q_a	b_{min}	b_{max}	Zipf
01	50	1000	20	1	30	50	≤ 5	10	70	1,50
02	50	1000	20	1	30	50	≥ 15	10	70	1,50
Cenário 3										
Gráfico	n	m	q	T	L_{min}	L_{max}	Q_a	b_{min}	b_{max}	Zipf
01	50	2000	20	1	30	50	≥ 15	10	70	1,50
Cenário 4										
Gráfico	n	m	q	T	L_{min}	L_{max}	Q_a	b_{min}	b_{max}	Zipf
01	50	4000	20	1	30	50	≤ 5	10	70	1,00
02	50	4000	20	1	30	50	≥ 15	10	70	1,00
03	50	4000	20	1	30	50	≥ 15	10	70	2,00
Cenário 5										
Gráfico	n	m	q	T	L_{min}	L_{max}	Q_a	b_{min}	b_{max}	Zipf
01	50	8000	20	1	30	50	≤ 5	10	70	1,50
02	50	8000	20	1	30	50	≥ 15	10	70	1,50

Tabela 5.27: Parâmetros dos gráficos

Não representamos o algoritmo Primal-Dual nos gráficos, por seu desempenho ser muito parecido com o do algoritmo AdWords. Mantivemos apenas este último como representante deste tipo de estratégia, que é resguardar os limites dos anunciantes para aproveitamento futuro. Esta eliminação visa exclusivamente melhorar a visualização dos gráficos deixando-os menos carregados.

5.5.1 Cenário 1

No primeiro cenário, onde m é bem menor que Q_{max} , os melhores desempenhos foram dos algoritmos Guloso e Aleatório, mas com estratégias bem diferentes como podemos observar na Figura 5.1. Enquanto o algoritmo Guloso passa a maior parte do dia com a maior receita acumulada, o algoritmo Aleatório acompanha bem de perto a solução ótima. No final do dia os três estão praticamente iguais.

O algoritmo Preditivo, que como o Aleatório utiliza informações estocásticas, acaba tendo um comportamento mais próximo do Guloso. É razoável pensar que se o algoritmo Guloso nos leva a uma solução quase ótima, nossa predição pode estar ligada a uma solução parecida. Este algoritmo, entretanto, esgota sua predição mais cedo, pois como m é pequeno, qualquer consulta perdida por estar fora da predição, não será compensada mais tarde.

Já o algoritmo AdWords começa parecido com o Guloso, mas chegando próximo de 100 consultas começa a se afastar. Este afastamento é causado pela opção por ofertas mais baixas, com o objetivo de preservar os limites de exibição para uso mais à frente. Este comportamento prejudica o desempenho do algoritmo AdWords. Como m é muito menor que Q_{max} , o algoritmo não recupera as oportunidades perdidas ao longo do dia. Enquanto o algoritmo Guloso é beneficiado, pois os limites não são atingidos.

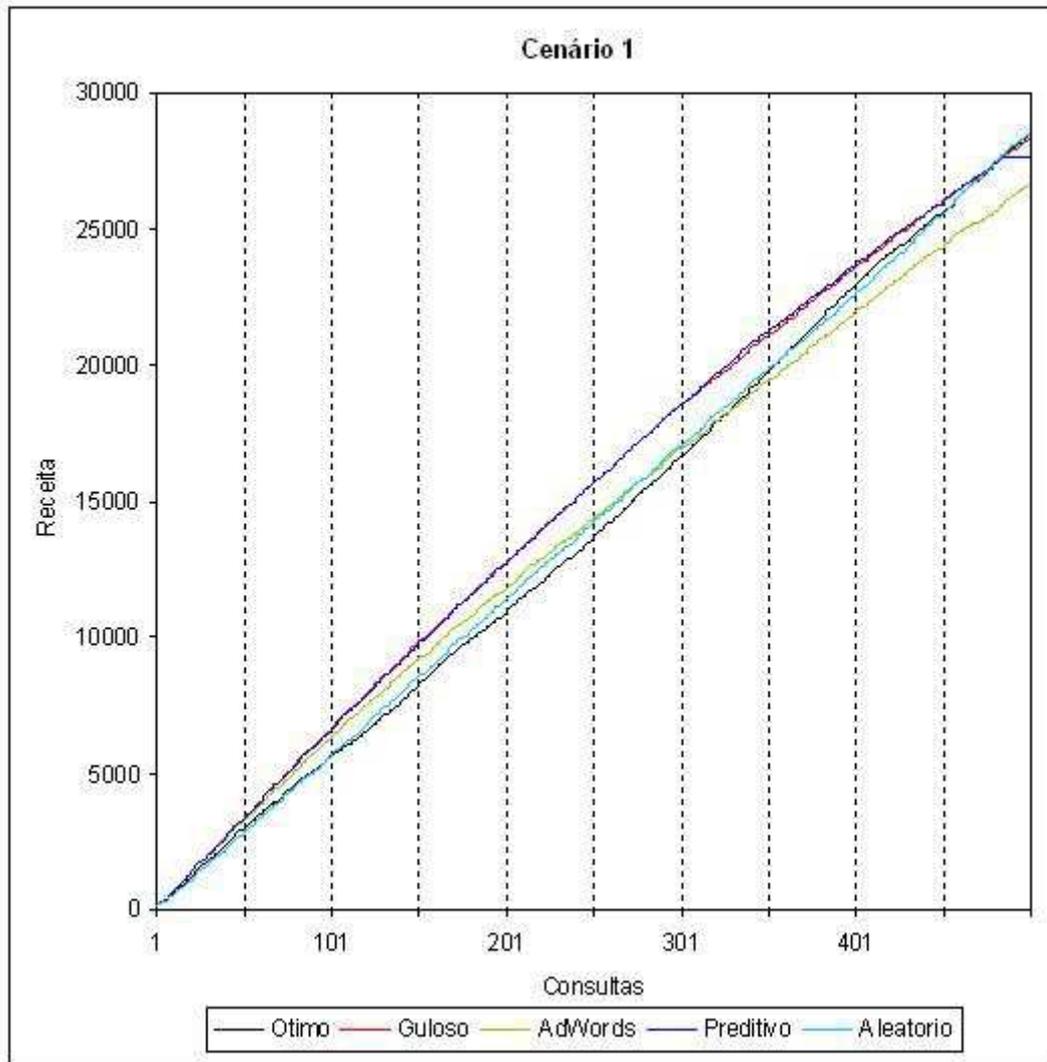


Figura 5.1: Cenário 1 - Gráfico 1

5.5.2 Cenário 2

No segundo cenário, onde m é menor que Q_{max} , os algoritmos têm um desempenho final parecido com o do primeiro cenário, sendo que o caminho apresenta algumas diferenças.

Na Figura 5.2, observamos que o algoritmo Aleatório continua acompanhando a tendência da solução ótima, mas os dois bem mais próximos do algoritmo Guloso que no cenário anterior. O motivo da proximidade não é o aumento de m , mas a utilização de um cenário menos competitivo entre os anunciantes. Neste cenário não existem tantas opções para a solução ótima que a diferenciem muito do algoritmo Guloso.

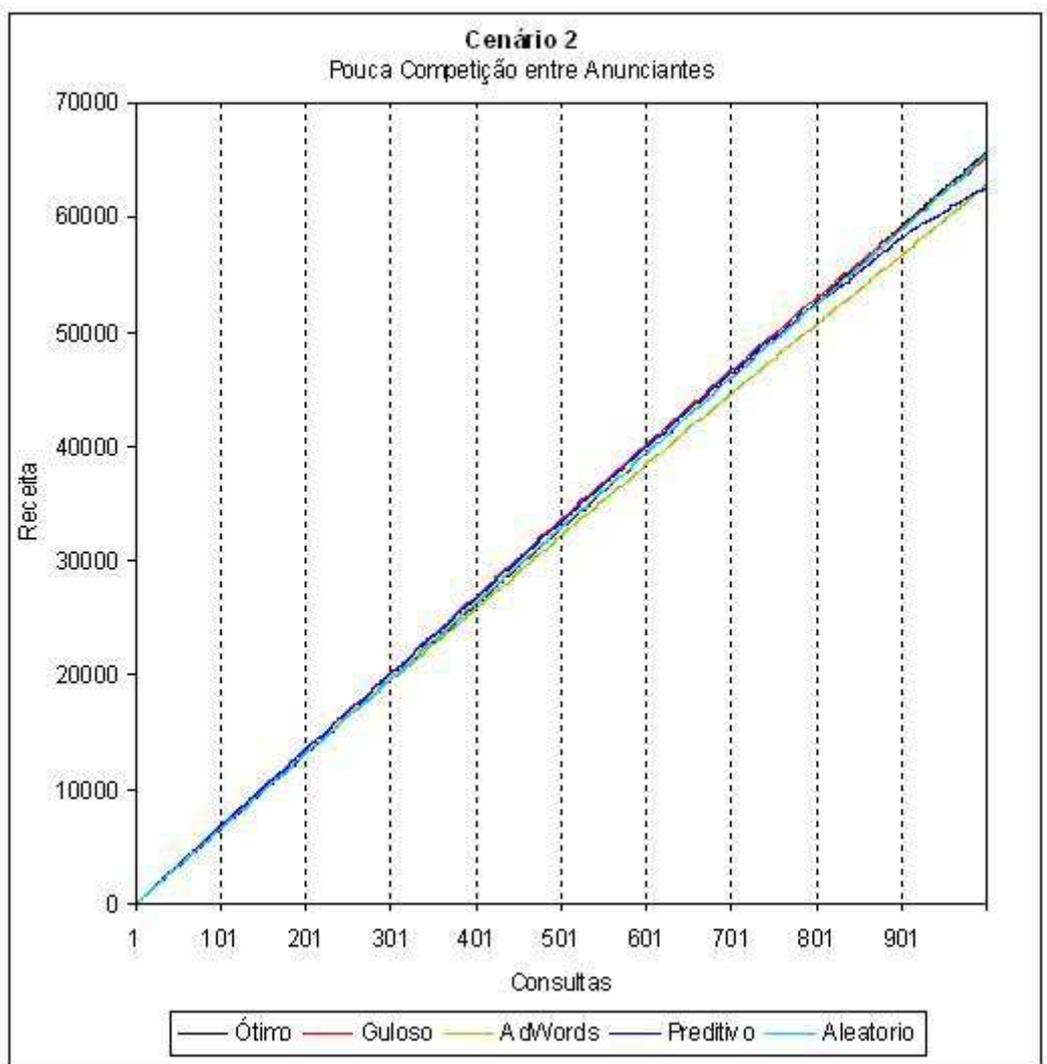


Figura 5.2: Cenário 2 - Gráfico 1

Na Figura 5.3 apresentamos um outro dia de processamento, num cenário com maior competição entre os anunciantes. Assim, por termos muito mais ofertas para uma mesma consulta, temos mais opções para a solução ótima. Podemos ver no gráfico que o algoritmo Aleatório e a solução ótima se afastam dos outros algoritmos até a metade das consultas e somente a partir de 800 consultas, as soluções voltam a ficar mais próximas.

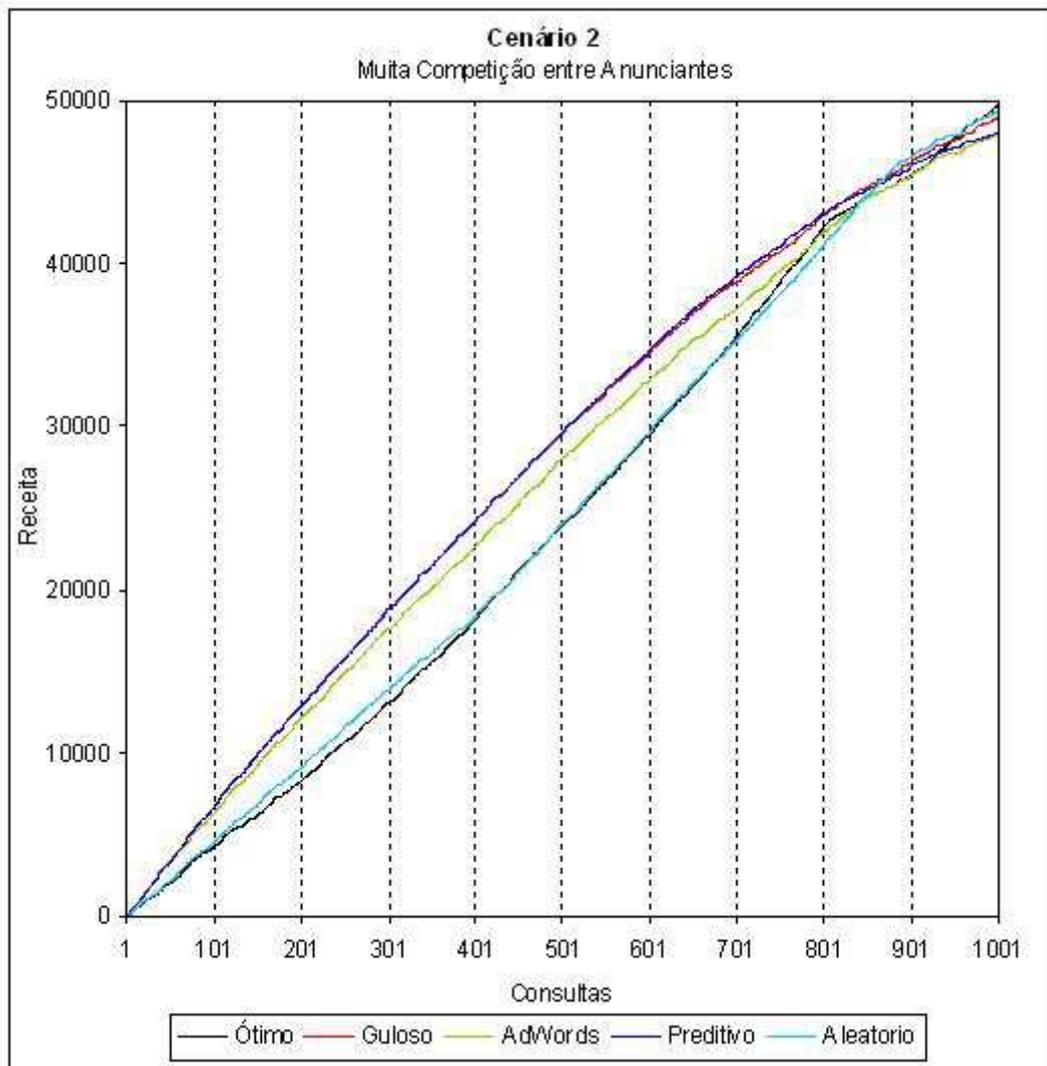


Figura 5.3: Cenário 2 - Gráfico 2

Com comportamento similar ao do Cenário 1, o algoritmo AdWords afasta-se gradativamente do Guloso e por conseguinte da melhor solução, enquanto o Preditivo que acompanha inicialmente este último, afasta-se de forma abrupta após 800 consultas, aproximadamente, e acaba com resultado próximo ao do algoritmo AdWords.

5.5.3 Cenário 3

No Cenário 3, onde m é igual a Q_{max} , é onde os algoritmos apresentam o desempenho mais próximo um do outro. Neste cenário a estratégia de reservar limites para uso futuro do AdWords funciona muito bem e a estratégia do algoritmo Guloso não funciona tão mal, pois temos um equilíbrio entre limites que se esgotam e os que se mantêm até o final.

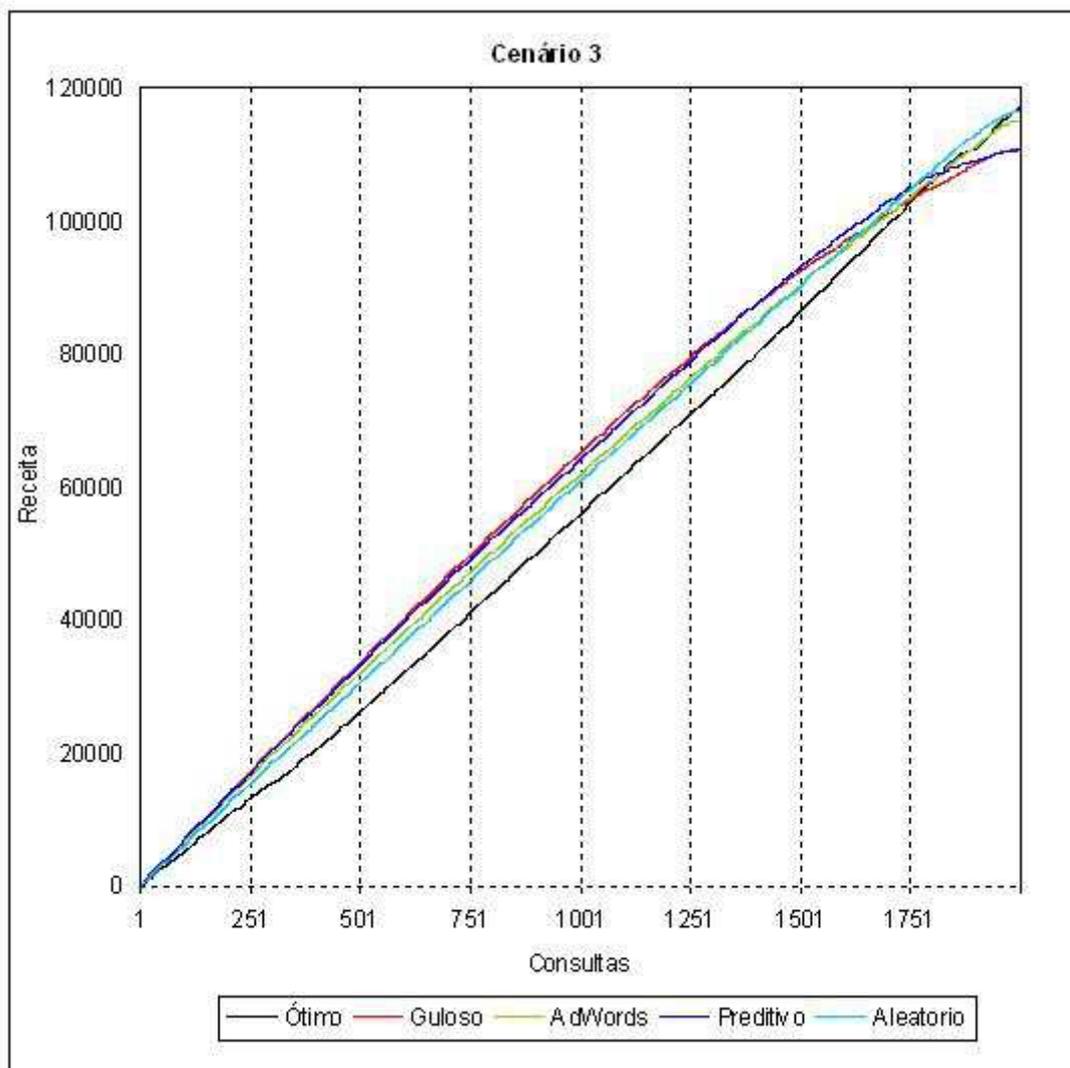


Figura 5.4: Cenário 3 - Gráfico 1

Dentro deste cenário, podemos observar na Figura 5.4 um desempenho bastante próximo dos algoritmos, mas o detalhe das primeiras 500 consultas, exibido na Figura 5.5 mostra que é mantida a ordem em que vimos os algoritmos nos cenários anteriores.

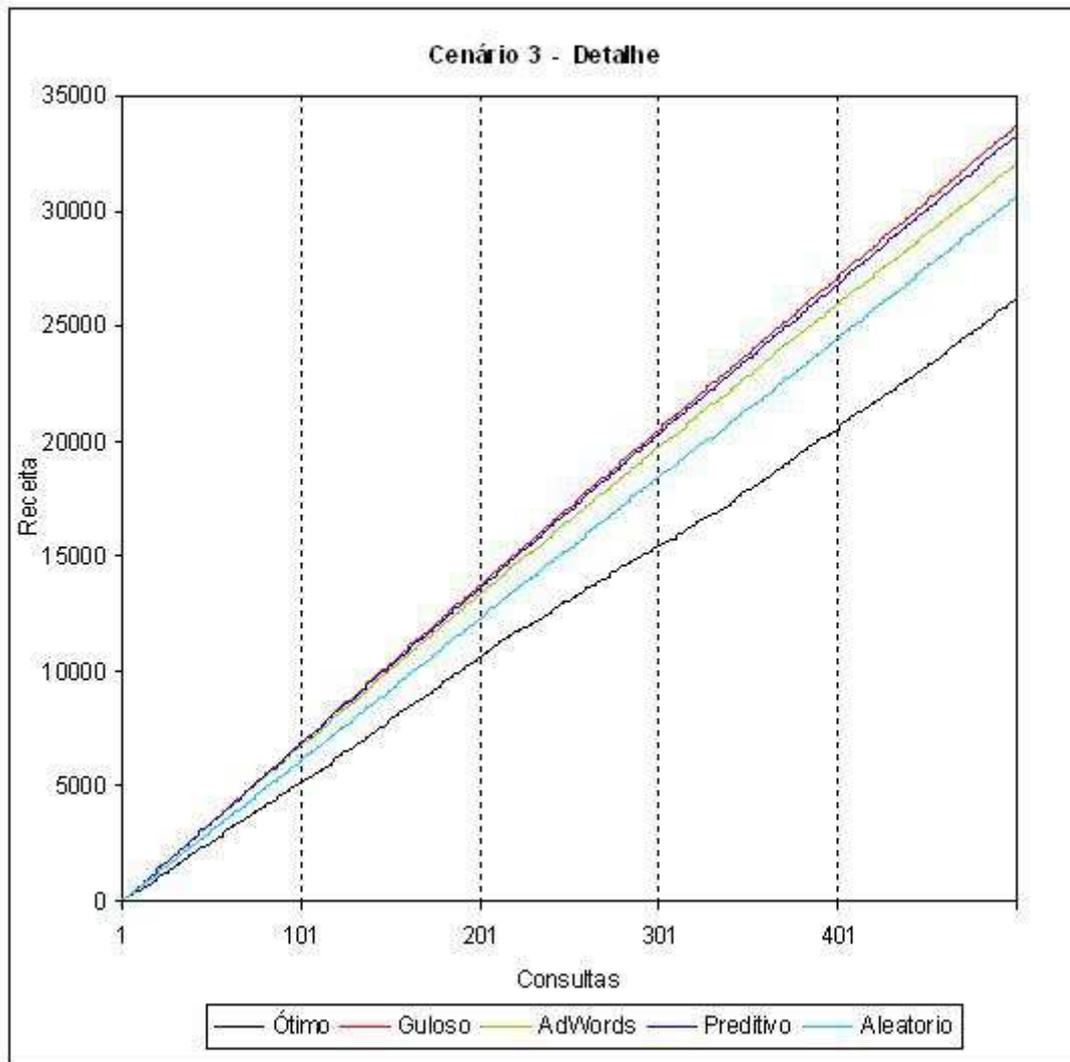


Figura 5.5: Cenário 3 - Gráfico 1

O algoritmo Guloso começa na frente, mas começa a cair em torno de 1500 consultas, quando m se aproxima de Q_{max} , pois este algoritmo esgota os limites de vários anunciantes mais rapidamente que os outros. É seguido de perto pelo Preditivo que, assim como o Guloso, para de crescer antes de $m = Q_{max}$, mas apenas em torno de 1750 consultas. Este algoritmo não esgota os limites, mas as previsões da solução ótima. Por isso, não tem um bom desempenho para $m \leq Q_{max}$.

Um pouco mais abaixo vem o algoritmo AdWords, que por manter um equilíbrio entre aproveitar as melhores ofertas e preveservar os limites, mantém um ganho contínuo até m atingir Q_{max} , ficando bastante próximo da solução ótima.

Neste cenário, pela primeira vez, o algoritmo Aleatório não acompanha o caminho da solução ótima, pois existem mais possibilidades para a solução

ótima que nos cenários anteriores. Esta distância é reforçada devido a termos uma grande competição entre os anunciante e com isso maior variação nas ofertas. Ainda assim, chegam bem próximos ao final.

5.5.4 Cenário 4

Para o Cenário 4, quando temos m é maior que Q_{max} , selecionamos 3 gráficos variando a competição entre os anunciante e a concentração das consultas.

Na Figura 5.6, a pouca competição entre os anunciantes ($Q_{max} \leq 5$) faz com que a curva de receita suba mais devagar que nos outros dois gráficos deste cenário.

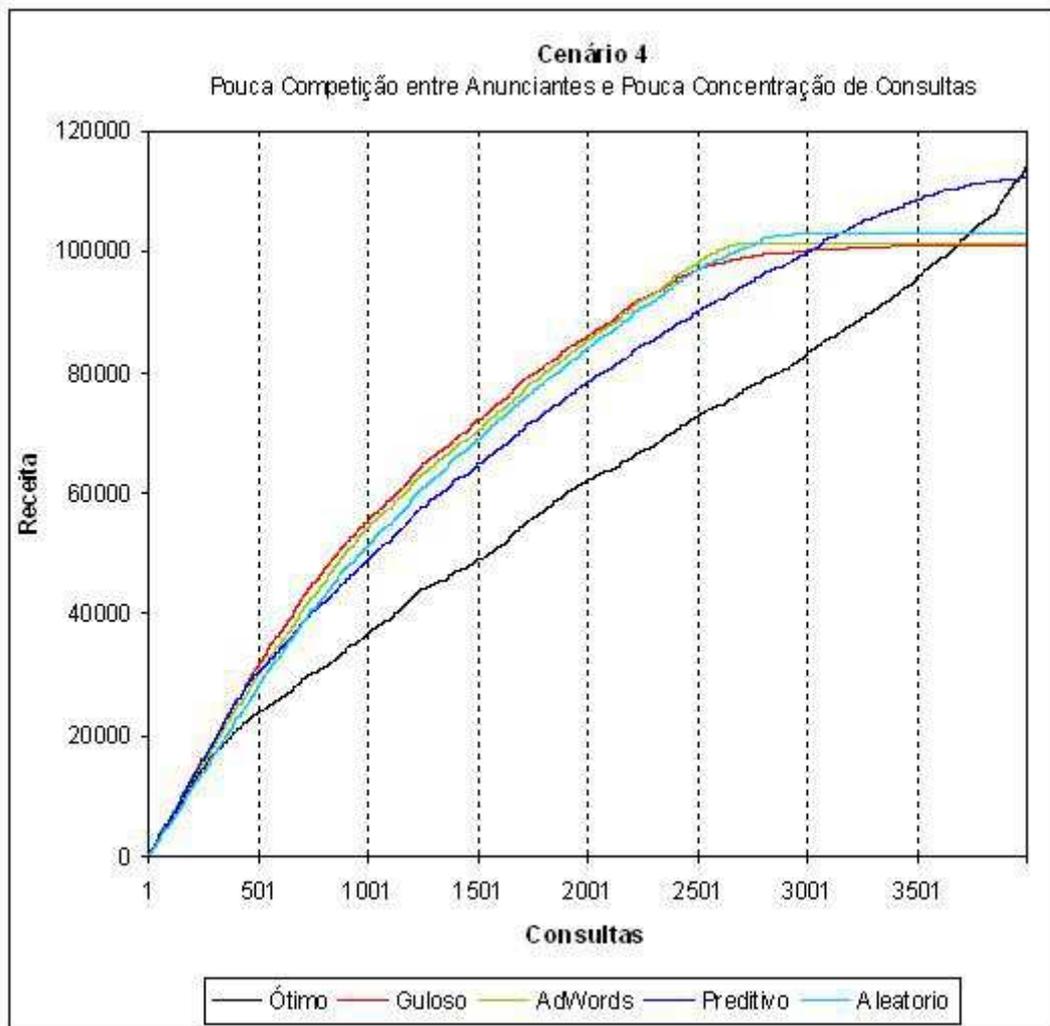


Figura 5.6: Cenário 4 - Gráfico 1

Na Figura 5.7, quando aumentamos a competição entre os anunciantes

e mantemos uma baixa concentração de consultas, a receita dos algoritmos Guloso, AdWords e Aleatório cresce mais rapidamente e chega ao patamar final quando m atinge Q_{max} , ou seja, com menos consultas que na Figura 5.6. Devido a grande competição, mesmo com alguns anunciantes com limites esgotados, existem outros interessados nas consultas. O aumento da competição, na Figura 5.7, favorece o algoritmo AdWords, que trabalha melhor a reserva dos diversos limites.

Em todos os gráficos deste cenário, o algoritmo Preditivo segue a maior parte do dia bem abaixo dos outros, acompanhando um pouco distante a mesma tendência da solução ótima em uma curva bem mais suave que a dos outros três algoritmos. O motivo de subir mais devagar é que deixa de atender mais consultas pelo caminho que os outros. Como chegam mais consultas (m) que o limite total de exibições (Q_{max}), muitas consultas ainda são atendidas com boa receita pelo algoritmo Preditivo, quando os outros algoritmos já chegaram em seu patamar máximo de receita.

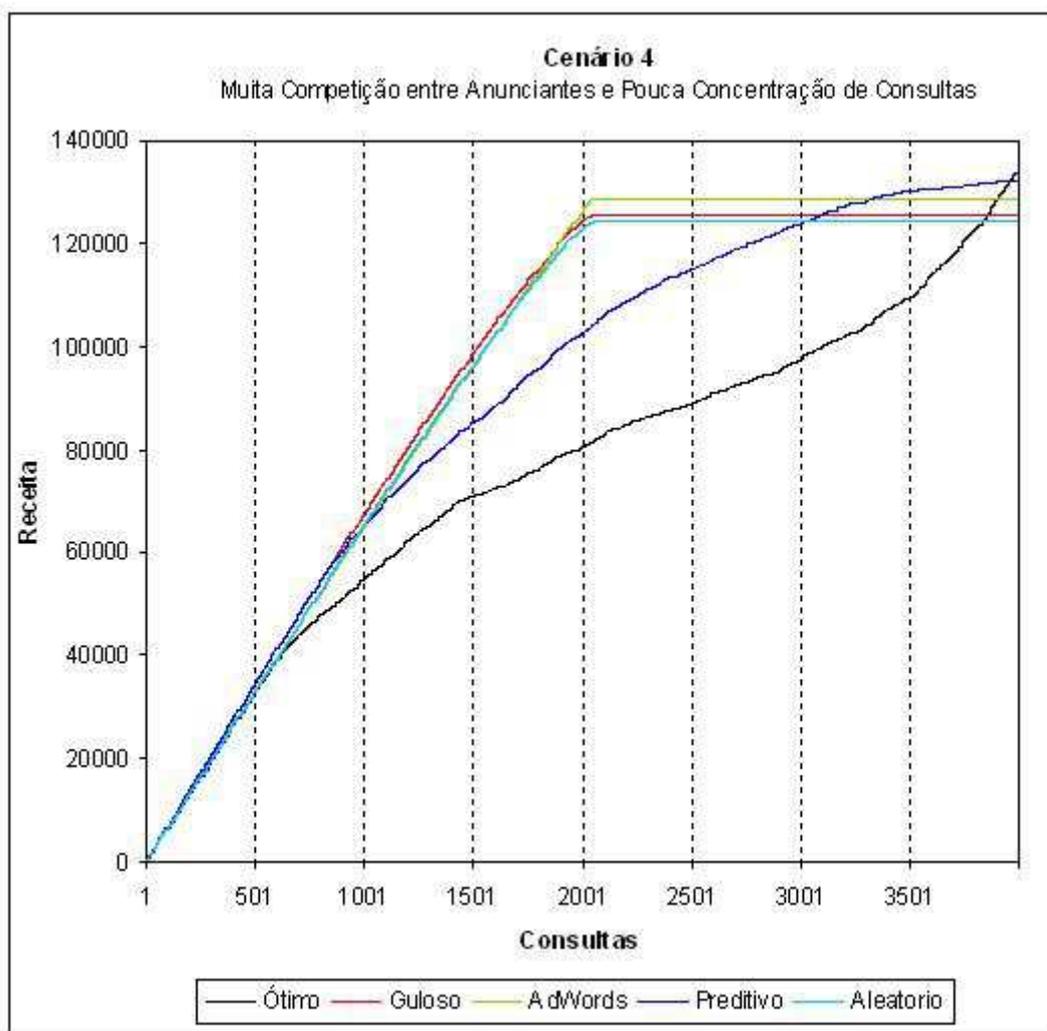


Figura 5.7: Cenário 4 - Gráfico 2

Na Figura 5.8, quando aumentamos a concentração de consultas, o algoritmo Preditivo se afasta dos outros mais rapidamente que na Figura 5.7, pois os três atendem diversas consultas com anunciantes de ofertas baixas, que não estão na solução prevista. Apesar deste afastamento e subir com uma inclinação menor que no gráfico anterior, o algoritmo Preditivo aumenta seu desempenho final em relação aos outros.

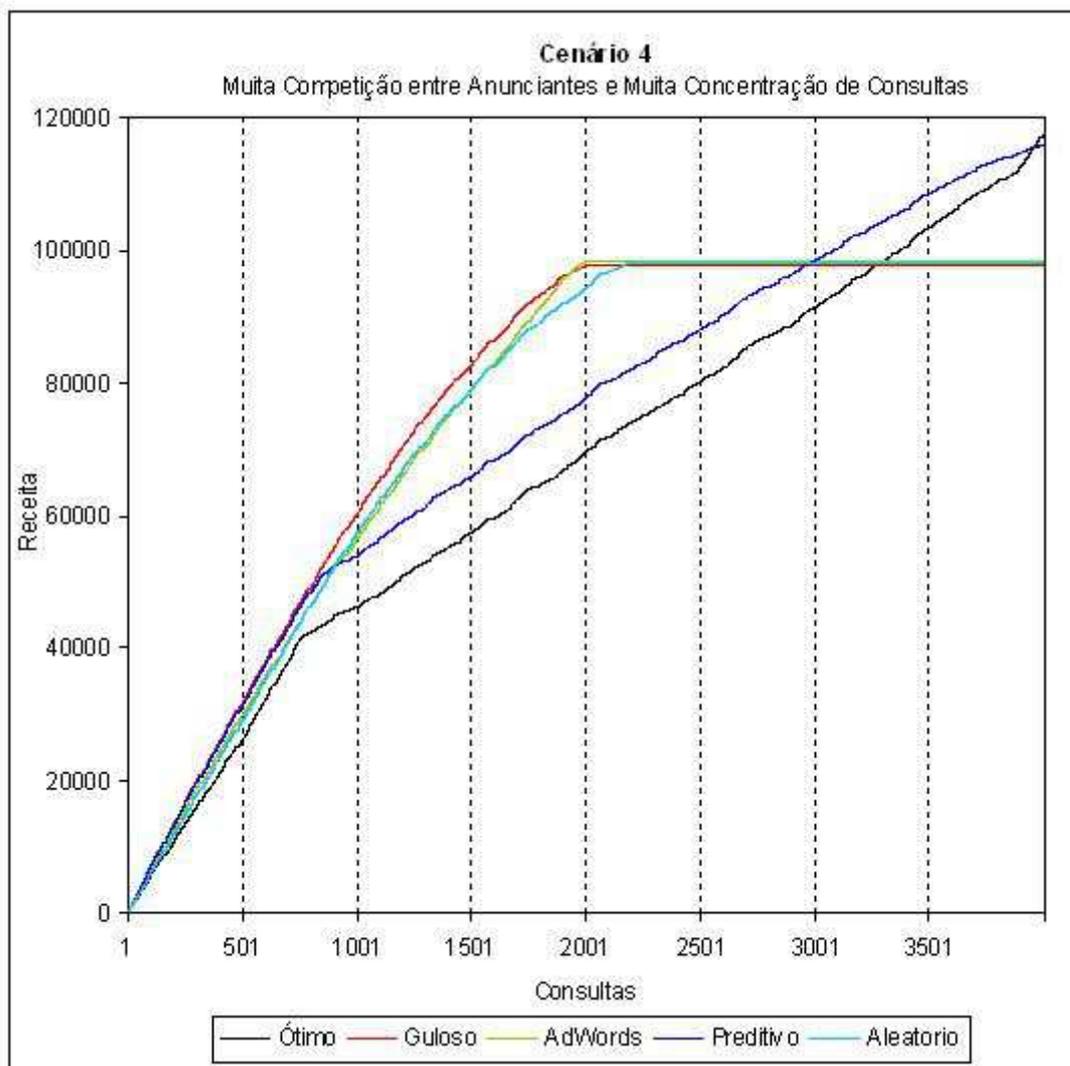


Figura 5.8: Cenário 4 - Gráfico 3

5.5.5 Cenário 5

O Cenário 5, quando temos m é bem maior que Q_{max} , quase não apresenta variações na evolução dos gráficos diários em relação ao Cenário 4, como podemos ver nas Figuras 5.9 e 5.10.

A diferença entre os dois cenários, que merece destaque, é que, ao final do dia, o algoritmo Preditivo tem uma receita final ainda maior que a dos outros algoritmos. Este crescimento é devido ao aumento de probabilidade de chegarem consultas que estão na solução ótima, em virtude do aumento total das consultas.

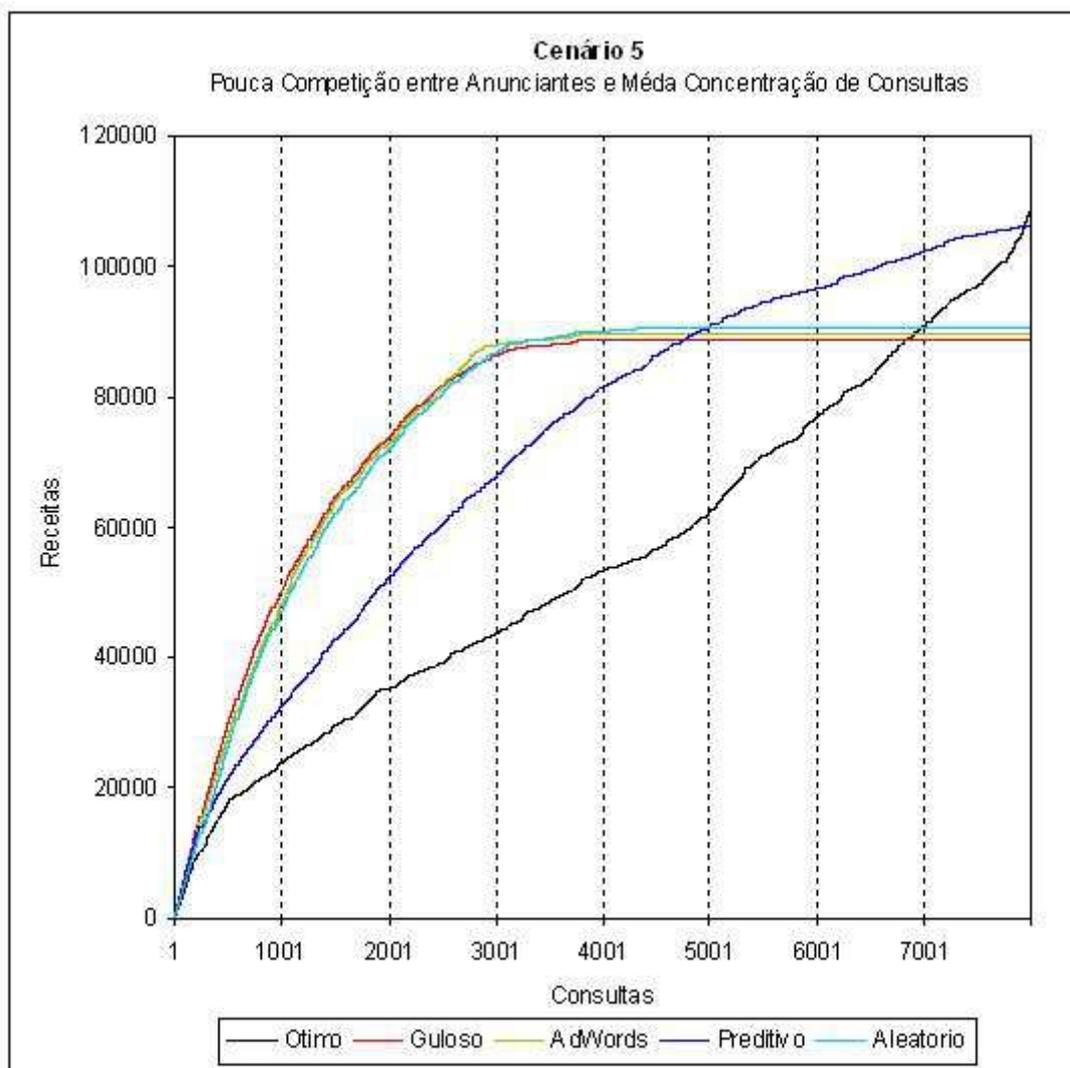


Figura 5.9: Cenário 5 - Gráfico 1

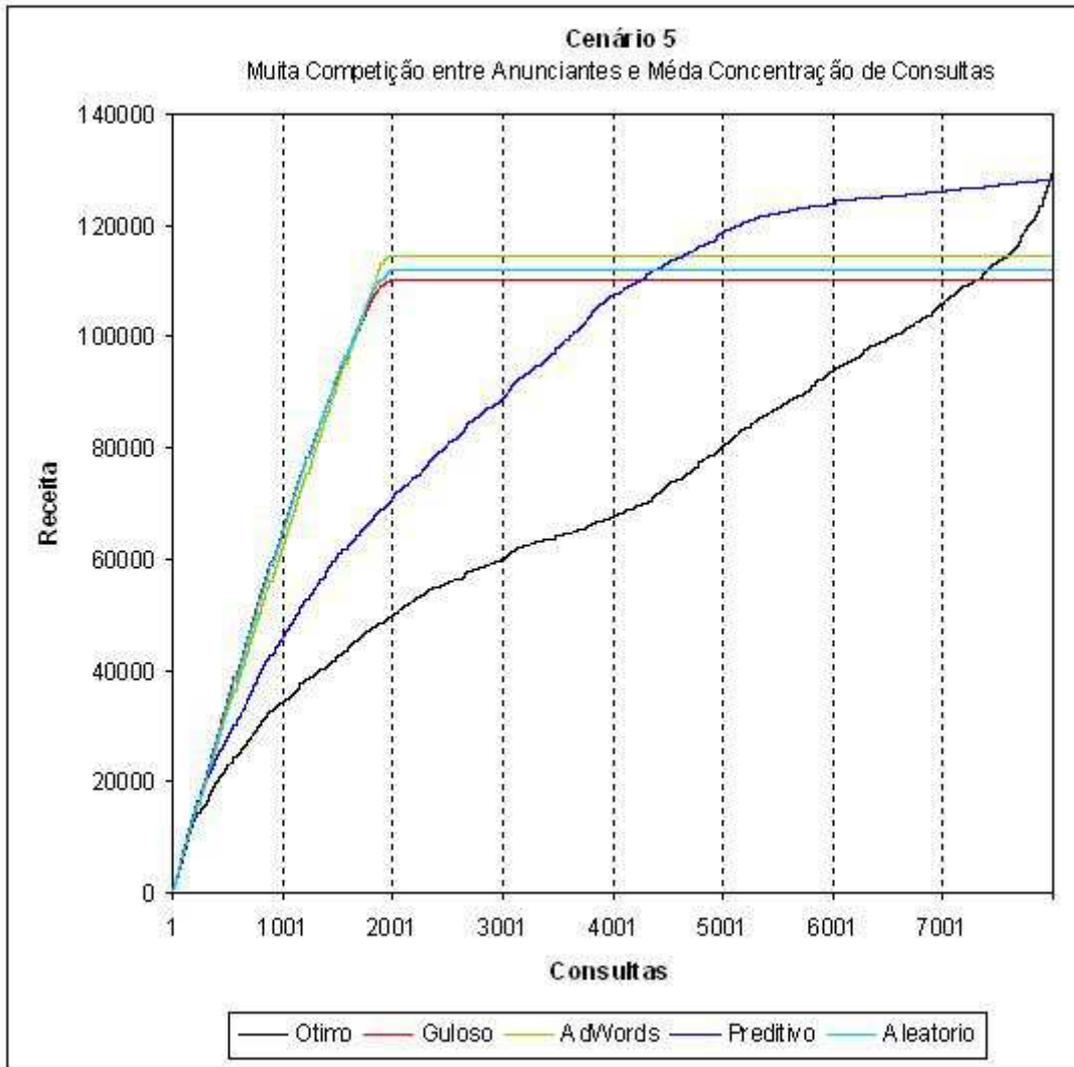


Figura 5.10: Cenário 5 - Gráfico 2

5.6 Avaliação de Cenários para Anunciantes Diferenciados

Nesta seção analisamos cenários em que consideramos a existência de alguns grandes anunciantes e muitos anunciantes de menor porte. A divisão em cinco cenários, considerando a razão m/Q_{max} , é a mesma utilizada na análise de anunciantes similares e apresentada na Tabela 5.5. Os outros parâmetros estão descritos na Tabela 5.28.

Parâmetros		
n	Anunciantes	50
Q	Tipos de consulta	20
T	Anunciantes exibidos por consulta	1
b_{min}	Oferta mínima	10
b_{max}	Oferta máxima	30
Q_a	Tipos de consulta que um anunciante faz oferta	≥ 15

Tabela 5.28: Parâmetros para anunciantes diferenciados

A geração dos limites dos anunciantes e das ofertas obedecem distribuições de Zipf, conforme descrevemos nas Seções 5.2.2 e 5.2.3, respectivamente.

Em cada um dos cenários apresentados, utilizamos duas distribuições diferentes para os limites de exibição. A primeira com potência 0.5, gera uma diferença menor entre o maior e o menor anunciante, e a segunda com potência 1, amplia significativamente esta diferença.

Em ambos os casos, utilizamos a potência 0.5 para gerar as ofertas entre 10 e 30.

As m consultas também são geradas por 4 distribuições de Zipf de potências distintas conforme explicado na seção 5.2.1.

Em todos os cenários utilizamos $Q_a \geq 15$ para termos bastante competição entre os anunciantes.

5.6.1

Cenários com $m < Q_{max}$

Nos Cenário 1 e 2, onde a quantidade diária de consultas é menor que o limite total de exibições dos anunciantes, os algoritmos tiveram um desempenho bastante próximo dos resultados obtidos com anunciantes similares.

Nas tabelas que apresentam os resultados destes cenários, podemos ver como se destacam os algoritmos Guloso e Aleatório. Como falamos anteriormente, como as consultas recebidas não são suficientes para esgotar a maior parte dos limites dos anunciantes, os algoritmos AdWords e Primal-Dual não têm boa recuperação pelos limites preservados no decorrer do tempo.

Uma característica importante da distribuição usada nestes cenários, é que os anunciantes com limites menores são os que têm ofertas maiores por uma consulta, independente do tipo. Isto significa que, apesar de terem seus limites esgotados pelo algoritmo Guloso, isto não é um problema, pois provavelmente também participam da solução ótima. A escolha destes anunciantes, para

uma ou outra consulta, não muda muito o resultado final, pois os anunciantes têm ofertas parecidas, independente da consulta.

Distribuição de Anunciantes por Zipf - Potência = 0.5								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
0.50	\bar{x}	500	14180	100.0%	96.1%	95.6%	93.0%	100.0%
	σ	0	0	0.0%	0.0%	0.1%	1.0%	0.0%
1.00	\bar{x}	500	14180	100.0%	95.8%	95.7%	94.5%	100.0%
	σ	0	0	0.0%	0.0%	0.1%	0.8%	0.0%
1.50	\bar{x}	500	14162	99.7%	95.2%	95.2%	94.8%	99.9%
	σ	0	10	0.1%	0.1%	0.1%	0.8%	0.0%
2.00	\bar{x}	500	13980	100.0%	95.8%	95.3%	94.3%	100.0%
	σ	0	12	0.0%	0.1%	0.1%	1.6%	0.0%

Tabela 5.29: Cenário 1 - Anunciantes por Zipf de potência 0.5

Distribuição de Anunciantes por Zipf - Potência = 1								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
0.50	\bar{x}	500	13874	99.9%	94.2%	93.8%	93.1%	99.9%
	σ	0	0	0.0%	0.1%	0.0%	1.2%	0.0%
1.00	\bar{x}	500	13882	99.8%	94.0%	93.3%	93.1%	99.9%
	σ	0	0	0.0%	0.0%	0.1%	2.1%	0.0%
1.50	\bar{x}	500	13726	99.9%	93.8%	93.5%	96.8%	99.8%
	σ	0	6	0.1%	0.2%	0.2%	1.4%	0.0%
2.00	\bar{x}	500	13457	99.7%	93.8%	93.5%	94.8%	99.7%
	σ	0	3	0.0%	0.1%	0.1%	0.4%	0.0%

Tabela 5.30: Cenário 1 - Anunciantes por Zipf de potência 1

Distribuição de Anunciantes por Zipf - Potência = 0.5								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
0.50	\bar{x}	1000	27822	99.9%	94.4%	94.4%	92.6%	100.0%
	σ	0	0	0.1%	0.1%	0.0%	0.5%	0.0%
1.00	\bar{x}	1000	27782	99.7%	94.4%	94.3%	94.6%	99.9%
	σ	0	21	0.0%	0.0%	0.0%	2.0%	0.1%
1.50	\bar{x}	1000	27655	99.8%	93.9%	93.9%	95.9%	99.7%
	σ	0	9	0.0%	0.1%	0.1%	1.1%	0.0%
2.00	\bar{x}	1000	27380	99.8%	94.7%	94.7%	96.7%	99.9%
	σ	0	10	0.0%	0.1%	0.0%	0.7%	0.0%

Tabela 5.31: Cenário 2 - Anunciantes por Zipf de potência 0.5

Distribuição de Anunciantes por Zipf - Potência = 1								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
0.50	\bar{x}	1000	26290	99.4%	93.0%	92.6%	92.5%	99.3%
	σ	0	0	0.0%	0.1%	0.1%	1.0%	0.2%
1.00	\bar{x}	1000	26245	99.4%	93.1%	92.8%	94.3%	99.3%
	σ	0	30	0.1%	0.1%	0.1%	1.5%	0.2%
1.50	\bar{x}	1000	26104	99.3%	93.0%	92.8%	94.3%	99.3%
	σ	0	37	0.1%	0.1%	0.1%	3.1%	0.1%
2.00	\bar{x}	1000	25858	99.6%	92.2%	91.2%	96.4%	99.7%
	σ	0	21	0.0%	0.5%	0.3%	1.9%	0.2%

Tabela 5.32: Cenário 2 - Anunciantes por Zipf de potência 1

5.6.2

Cenário com $m = Q_{max}$

No Cenário 3, quando a quantidade diária de consultas é igual ao limite total de exibições dos anunciantes, os algoritmos apresentaram um desempenho próximo ao obtido quando tínhamos anunciantes similares.

Neste cenário a estratégia de reservar limites dos anunciantes mostra-se mais eficiente e os algoritmos AdWords e Primal-Dual passam a ter um ótimo desempenho, mas o Guloso continua com um resultado muito bom. O algoritmo Aleatório continua com o melhor resultado entre todos os algoritmos.

Distribuição de Anunciantes por Zipf - Potência = 0.5								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
0.50	\bar{x}	1977	48632	98.3%	99.2%	99.2%	95.0%	100.0%
	σ	0	0	0.1%	0.0%	0.0%	0.5%	0.0%
1.00	\bar{x}	1977	48550	98.2%	99.1%	99.1%	95.4%	99.9%
	σ	0	36	0.2%	0.1%	0.1%	1.2%	0.0%
1.50	\bar{x}	1977	47882	97.1%	99.2%	99.2%	95.7%	99.8%
	σ	0	80	0.1%	0.1%	0.0%	2.2%	0.1%
2.00	\bar{x}	1977	47742	98.5%	99.6%	99.5%	97.4%	99.6%
	σ	0	41	0.3%	0.0%	0.0%	1.2%	0.4%

Tabela 5.33: Cenário 3 - Anunciantes por Zipf de potência 0.5

Distribuição de Anunciantes por Zipf - Potência = 1								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
0.50	\bar{x}	1978	42235	95.0%	98.9%	96.0%	96.2%	100.0%
	σ	0	0	0.6%	0.1%	0.0%	0.6%	0.0%
1.00	\bar{x}	1978	41995	96.3%	98.9%	95.9%	95.0%	99.9%
	σ	0	47	0.3%	0.0%	0.0%	2.1%	0.0%
1.50	\bar{x}	1978	41372	93.7%	97.4%	95.8%	96.7%	99.8%
	σ	0	58	0.2%	0.7%	0.5%	0.7%	0.1%
2.00	\bar{x}	1978	40794	99.0%	99.5%	96.5%	97.8%	99.5%
	σ	0	73	0.1%	0.1%	0.0%	0.9%	0.1%

Tabela 5.34: Cenário 3 - Anunciantes por Zipf de potência 1

5.6.3

Cenários com $m > Q_{max}$

Nos dois últimos cenários, quando a quantidade diária de consultas ultrapassa a capacidade total dos anunciantes, o algoritmo Preditivo foi o que obteve o melhor desempenho, como no conjunto anterior de cenários.

A grande diferença é que os outros algoritmos continuaram com um desempenho muito bom, apenas um pouco abaixo do Preditivo. O motivo de não haver uma grande mudança é que as ofertas de um mesmo anunciante são parecidas e a atribuição do anunciante para uma ou outra oferta, pouco altera o resultado final.

Distribuição de Anunciantes por Zipf - Potência = 0.5								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
0.50	\bar{x}	1977	48632	98.5%	99.1%	99.1%	100.0%	99.9%
	σ	0	0	0.1%	0.0%	0.0%	0.0%	0.0%
1.00	\bar{x}	1977	48632	98.7%	99.2%	99.1%	100.0%	99.8%
	σ	0	0	0.1%	0.1%	0.0%	0.0%	0.1%
1.50	\bar{x}	1977	48610	97.9%	98.2%	98.1%	97.5%	98.2%
	σ	0	3	0.1%	0.2%	0.1%	0.7%	0.2%
2.00	\bar{x}	1977	48468	97.7%	97.8%	97.7%	98.4%	98.6%
	σ	0	30	0.1%	0.1%	0.2%	0.6%	0.1%

Tabela 5.35: Cenário 4 - Anunciantes por Zipf de potência 0.5

Distribuição de Anunciantes por Zipf - Potência = 1								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
0.50	\bar{x}	1978	42234	97.5%	98.3%	95.3%	99.6%	99.2%
	σ	0	2	0.0%	0.0%	0.0%	0.5%	0.0%
1.00	\bar{x}	1978	42235	98.0%	98.3%	95.4%	99.9%	99.4%
	σ	0	0	0.1%	0.1%	0.1%	0.1%	0.1%
1.50	\bar{x}	1978	42191	96.5%	97.3%	94.3%	99.7%	97.9%
	σ	0	3	0.1%	0.2%	0.1%	0.1%	0.2%
2.00	\bar{x}	1978	41843	97.6%	98.1%	95.1%	98.9%	98.7%
	σ	0	22	0.0%	0.0%	0.0%	0.4%	0.0%

Tabela 5.36: Cenário 4 - Anunciantes por Zipf de potência 1

Distribuição de Anunciantes por Zipf - Potência = 0.5								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
0.50	\bar{x}	1977	48632	98.6%	99.1%	99.0%	100.0%	99.8%
	σ	0	0	0.1%	0.1%	0.0%	0.0%	0.0%
1.00	\bar{x}	1977	48632	98.4%	98.9%	98.8%	99.9%	99.4%
	σ	0	0	0.1%	0.1%	0.0%	0.3%	0.1%
1.50	\bar{x}	1977	48632	98.3%	98.8%	98.8%	100.0%	99.1%
	σ	0	0	0.1%	0.1%	0.1%	0.0%	0.1%
2.00	\bar{x}	1977	48632	97.7%	97.9%	97.7%	98.7%	98.0%
	σ	0	0	0.0%	0.0%	0.0%	1.0%	0.0%

Tabela 5.37: Cenário 5 - Anunciantes por Zipf de potência 0.5

Distribuição de Anunciantes por Zipf - Potência = 1								
		Solução Ótima		Algoritmos Online				
Zipf		Exibe	Receita	Guloso	AdWords	P-Dual	Preditivo	Aleatório
0.50	\bar{x}	1978	42235	98.0%	98.9%	96.0%	100.0%	99.8%
	σ	0	0	0.1%	0.0%	0.1%	0.0%	0.0%
1.00	\bar{x}	1978	42235	98.2%	98.7%	95.8%	100.0%	99.8%
	σ	0	0	0.0%	0.1%	0.0%	0.0%	0.1%
1.50	\bar{x}	1978	42235	97.7%	98.6%	95.6%	99.9%	99.6%
	σ	0	0	0.1%	0.1%	0.1%	0.2%	0.1%
2.00	\bar{x}	1978	42120	97.3%	98.1%	95.1%	99.0%	98.5%
	σ	0	11	0.1%	0.1%	0.0%	0.3%	0.1%

Tabela 5.38: Cenário 5 - Anunciantes por Zipf de potência 1