

**Ordenação evolutiva de anúncios  
em publicidade computacional**

Marcos Eduardo Bolelli Broinizi

TEXTO APRESENTADO  
AO  
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA  
DA  
UNIVERSIDADE DE SÃO PAULO  
PARA  
OBTENÇÃO DO TÍTULO  
DE  
DOUTOR EM CIÊNCIAS

Programa: Ciência da Computação  
Orientador: Prof. Dr. João Eduardo Ferreira

São Paulo, agosto de 2015

# Ordenação evolutiva de anúncios em publicidade computacional

Esta versão da dissertação/tese contém as correções e alterações sugeridas pela Comissão Julgadora durante a defesa da versão original do trabalho, realizada em 15/06/2015. Uma cópia da versão original está disponível no Instituto de Matemática e Estatística da Universidade de São Paulo.

Comissão Julgadora:

- Prof. Dr. João Eduardo Ferreira (orientadora) - IME-USP
- Prof. Dr. Marco Antonio Pinheiro de Cristo - UFAM
- Prof. Dr. Altigran Soares da Silva - UFAM
- Prof. Dr. Luciano Vieira de Araújo - EACH-USP
- Prof. Dr. Roberto Marcondes Cesar Jr. - IME-USP

# Agradecimentos

Agradeço em primeiro lugar à minha família, em especial à minha esposa Tatiana, minha mãe Tânia e meu pai Luciano, por todo o suporte, auxílio e, principalmente, compreensão por todos os momentos em que estive ausente para concluir este trabalho. Dedico este trabalho à toda a minha família e principalmente à minha filha, Isabella, que nasceu no decorrer deste projeto e que me inspirou a concluí-lo mesmo nos momentos que eu mais desejava estar ao seu lado. Agradeço a orientação e confiança do João Eduardo Ferreira na execução do projeto, assim como dos meus colegas que me auxiliaram, sobretudo, Danilo Mutti e André Casimiro. Por fim, agradeço a paciência e carinho da minha esposa, Tatiana, por todos esses anos que enfrentamos juntos para alcançar esse objetivo.

# Resumo

Broinizi, M. E. B. **Ordenação evolutiva de anúncios em publicidade computacional**. 2015. 102 f. Texto (Obtenção de Título de Doutor em Ciências) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2015.

Otimizar simultaneamente os interesses dos usuários, anunciantes e publicadores é um grande desafio na área de publicidade computacional. Mais precisamente, a ordenação de anúncios, ou *ad ranking*, desempenha um papel central nesse desafio. Por outro lado, nem mesmo as melhores fórmulas ou algoritmos de ordenação são capazes de manter seu status por um longo tempo em um ambiente que está em constante mudança. Neste trabalho, apresentamos uma análise orientada a dados que mostra a importância de combinar diferentes dimensões de publicidade computacional por meio de uma abordagem evolutiva para ordenação de anúncios a fim de responder a mudanças de forma mais eficaz. Nós avaliamos as dimensões de valor comercial, desempenho histórico de cliques, interesses dos usuários e a similaridade textual entre o anúncio e a página. Nessa avaliação, nós averiguamos o desempenho e a correlação das diferentes dimensões. Como consequência, nós desenvolvemos uma abordagem evolucionária para combinar essas dimensões. Essa abordagem é composta por três partes: um repositório de configurações para facilitar a implantação e avaliação de experimentos de ordenação; um componente evolucionário de avaliação orientado a dados; e um motor de programação genética para evoluir fórmulas de ordenação de anúncios. Nossa abordagem foi implementada com sucesso em um sistema real de publicidade computacional responsável por processar mais de quatorze bilhões de requisições de anúncio por mês. De acordo com nossos resultados, essas dimensões se complementam e nenhuma delas deve ser negligenciada. Além disso, nós mostramos que a combinação evolucionária dessas dimensões não só é capaz de superar cada uma individualmente, como também é conseguiu alcançar melhores resultados do que métodos estáticos de ordenação de anúncios.

**Palavras-chave:** publicidade computacional, publicidade contextualizada, publicidade online, análise de componentes principais, análise exploratória de dados, publicidade online, publicidade digital, programação genética.

# Abstract

Broinizi, M. E. B. **Evolutionary ad ranking for computational advertising**. 2015. 102 f. Texto (Obtenção de Título de Doutor em Ciências) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2015.

Simultaneous optimization of users, advertisers and publishers' interests has been a formidable challenge in online advertising. More concretely, ranking of advertising, or more simply *ad ranking*, has a central role in this challenge. However, even the best ranking formula or algorithm cannot withstand the ever-changing environment of online advertising for a long time. In this work, we present a data-driven analysis that shows the importance of combining different aspects of online advertising through an evolutionary approach for ad ranking in order to effectively respond to changes. We evaluated aspects ranging from bid values and previous click performance to user behavior and interests, including the textual similarity between ad and page. In this evaluation, we assessed commercial performance along with the correlation between different aspects. Therefore, we proposed an evolutionary approach for combining these aspects. This approach was composed of three parts: a configuration repository to facilitate deployment and evaluation of ranking experiments; an evolutionary data-based evaluation component; and a genetic programming engine to evolve ad ranking formulae. Our approach was successfully implemented in a real online advertising system that processes more than fourteen billion ad requests per month. According to our results, these aspects complement each other and none of them should be neglected. Moreover, we showed that the evolutionary combination of these aspects not only outperformed each of them individually, but was also able to achieve better overall results than static ad ranking methods.

**Keywords:** computational advertising, contextual advertising, online advertising, principal component analysis, exploratory data analysis, genetic programming, learning to advertising.

# Sumário

<b>Lista de Abreviaturas</b>	<b>vii</b>
<b>Lista de Figuras</b>	<b>viii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Problema . . . . .	4
1.2 Contextualização . . . . .	6
1.2.1 Objetivo Principal . . . . .	8
1.3 Hipótese . . . . .	8
1.4 Justificativa da Hipótese . . . . .	9
1.5 Resumo da proposta . . . . .	11
1.6 Contribuições . . . . .	11
1.7 Organização do texto . . . . .	12
<b>2 Fundamentos</b>	<b>14</b>
2.1 Participantes . . . . .	14
2.2 Tipos de anúncios . . . . .	15
2.3 Modelos de pagamento . . . . .	16
2.4 Anatomia de um anúncio . . . . .	17
2.5 Sistema de publicidade computacional . . . . .	19
2.6 Dimensões e análise multivariada . . . . .	20
2.6.1 Análise de componentes principais . . . . .	21
2.7 Indicadores e função objetivo . . . . .	21
2.8 Ordenação de anúncios . . . . .	24
2.9 Reordenação de anúncios . . . . .	26
2.10 Sistemas de software adaptativos . . . . .	27
2.11 Programação genética . . . . .	28
2.12 Conclusão . . . . .	31
<b>3 Trabalhos relacionados</b>	<b>33</b>
<b>4 Proposta: Abordagem evolutiva em publicidade computacional</b>	<b>39</b>
<b>5 Engenharia e arquitetura</b>	<b>43</b>
5.1 Descrição do sistema de publicidade computacional . . . . .	43
5.1.1 <i>Crawling</i> . . . . .	44

5.1.2	Classificação . . . . .	44
5.1.3	Interesses do usuário . . . . .	44
5.1.4	Gerenciamento de anúncios . . . . .	45
5.1.5	Indexação de anúncios . . . . .	45
5.1.6	Requisição de anúncios . . . . .	46
5.1.7	Consulta de anúncios . . . . .	46
5.1.8	Seleção de anúncios . . . . .	46
5.1.9	Reordenação de anúncios . . . . .	47
5.1.10	Cálculo de custo por clique . . . . .	47
5.1.11	Resposta . . . . .	48
5.2	Ambiente . . . . .	48
5.3	Arquitetura . . . . .	49
5.4	Implementação da capacidade adaptativa . . . . .	51
5.5	ReConf: repositório centralizado de configurações . . . . .	53
5.5.1	RESTful Web Service API . . . . .	53
5.5.2	Biblioteca cliente de configuração . . . . .	54
5.6	Arquitetura de reconfiguração . . . . .	57
5.6.1	Demandas de negócio . . . . .	58
5.6.2	Experimentos controlados . . . . .	60
5.7	Conclusão . . . . .	62
<b>6</b>	<b>Implementação da ordenação evolutiva</b> . . . . .	<b>63</b>
6.1	Escopo . . . . .	63
6.2	Restrições de dados . . . . .	63
6.2.1	Função objetivo ajustada . . . . .	64
6.3	Arquitetura evolutiva . . . . .	66
6.4	Avaliação . . . . .	66
6.5	Evolução . . . . .	67
6.5.1	Indivíduos . . . . .	68
6.5.2	Operadores genéticos . . . . .	69
6.5.3	Função de <i>fitness</i> . . . . .	71
6.5.4	Seleção . . . . .	71
6.6	Ciclo de evolução . . . . .	72
6.7	Justificativa da implementação . . . . .	73
6.8	Limites e discussões . . . . .	74
6.8.1	Múltiplos objetivos . . . . .	75
6.8.2	Valor escalar por dimensão . . . . .	75
6.8.3	O valor do clique e a dimensão de valor comercial . . . . .	76
6.9	Conclusão . . . . .	77
<b>7</b>	<b>Resultados</b> . . . . .	<b>78</b>
7.1	Mudanças e adaptação . . . . .	78
7.2	Análise dos dados . . . . .	79
7.3	Experimentos com dimensões . . . . .	81

7.4	Experimento 1 - avaliação da abordagem evolutiva . . . . .	86
7.4.1	Restrições . . . . .	86
7.4.2	Geração inicial . . . . .	87
7.4.3	Parâmetros . . . . .	88
7.4.4	Resultados do experimento . . . . .	88
7.4.5	Promovendo a melhor solução do experimento 1 . . . . .	90
7.5	Experimento 2 - evolução por períodos maiores . . . . .	91
7.5.1	Resultados do experimento 2 . . . . .	92
7.5.2	Promovendo a melhor solução do experimento 2 . . . . .	94
7.6	Experimento 3 - evoluções consecutivas . . . . .	95
7.7	Experimento 4 - reavaliação . . . . .	96
<b>8</b>	<b>Conclusão</b>	<b>98</b>
8.1	Síntese do trabalho . . . . .	98
8.2	Trabalhos futuros . . . . .	100
<b>A</b>	<b>Exemplos de Fórmulas</b>	<b>102</b>
	<b>Referências Bibliográficas</b>	<b>104</b>

# Lista de Abreviaturas

CPC	Custo por clique ( <i>cost per click</i> )
CPM	Custo por milhar de impressão, custo por mil visualizações ( <i>cost per mille</i> )
CTR	Taxa de clique ( <i>click-through rate</i> )
eCPM	Custo efetivo por milhar de impressão, custo efetivo por mil visualizações ( <i>effective cost per mille</i> )
ROI	Retorno sobre o investimento ( <i>return on investment</i> )

# Lista de Figuras

1.1	<b>Gastos Mundiais com Publicidade Digital, 2012-2018 bilhões de dólares, % de alteração e % do total gasto com publicidades em todas as mídias</b> Nota: inclui publicidade apresentada em computadores de mesa e laptops assim como telefones e <i>tablets</i> , e inclui todos os diversos formatos de publicidade nessas plataformas; excluindo publicidade de mensagens SMS, MMS e P2P Fonte: eMarketer, Jun 2014 [eMac] . . . . .	1
1.2	<b>5 Maiores Países, Ordenados por Gastos com Publicidade em Todas as Mídias, Mídia Digital e Móvel, 2015 bilhões de dólares</b> Nota: *inclui SMS, MMS e P2P; **excluindo Hong Kong Fonte: eMarketer, Dez 2014 [eMaa] . . . . .	2
1.3	<b>Gastos com Publicidade Digital na América Latina, por País, 2012-2018 bilhões de dólares</b> Nota: inclui publicidade apresentada em computadores de mesa e laptops assim como telefones e <i>tablets</i> , e inclui todos os diversos formatos de publicidade nessas plataformas; excluindo publicidade de mensagens SMS, MMS e P2P; os valores podem não somar o total adequadamente devido a arredondamentos; eMarketer baseou as projeções de gastos com publicidade digital do México com base nos dados do IAB México/PwC para os quais o último ano completo de medição foi 2012. Fonte: eMarketer, Junho 2014 [eMab] . . . . .	2
1.4	<b>Crescimento dos Gastos Mundiais com Publicidade Digital na América Latina, por País, 2012-2018 % de alteração</b> Nota: inclui publicidade apresentada em computadores de mesa e laptops assim como telefones e <i>tablets</i> , e inclui todos os diversos formatos de publicidade nessas plataformas; excluindo publicidade de mensagens SMS, MMS e P2P; os valores podem não somar o total adequadamente devido a arredondamentos; eMarketer baseou as projeções de gastos com publicidade digital do México com base nos dados do IAB México/PwC para os quais o último ano completo de medição foi 2012. Fonte: eMarketer, Junho 2014 [eMab] . . . . .	3
4.1	<b>Abordagem evolutiva - Ciclo de Exploração, Avaliação e Evolução</b> . . . . .	40
4.2	<b>Abordagem evolutiva - A construção da ordenação evolutiva tem como alicerce um sistema de publicidade adaptativo. Para desenvolver a capacidade de adaptação criamos como base uma arquitetura de reconfiguração para o sistema de publicidade computacional.</b> . . . . .	42
5.1	Arquitetura do sistema de publicidade computacional. . . . .	50
5.2	<b>ReConf API - com os verbos HTTP implementados. Essa Figura foi gerada pela aplicação <i>www.api.hub.com/raml/api-designer</i>.</b> . . . . .	55

5.3	Exemplo de interface de configuração utilizando a biblioteca cliente em Java. . . . .	56
5.4	Arquitetura do sistema de publicidade computacional incluindo a capacidade de reconfiguração. . . . .	58
6.1	Arquitetura evolutiva do sistema de publicidade computacional. . . . .	66
6.2	<b>Representação de uma função como uma árvore.</b> . . . . .	70
6.3	<b>Crossover</b> - Os dois indivíduos selecionados para reprodução, à esquerda, têm nós selecionados aleatoriamente como pontos de <i>crossover</i> . A combinação da árvore do primeiro indivíduo substituindo a sub-árvore de <i>crossover</i> pela sub-árvore do segundo indivíduo cria o novo indivíduo, filho da geração anterior. As partes não utilizadas são descartadas. Para gerar o segundo filho, o processo é repetido invertendo os pais e escolhendo-se novos pontos de <i>crossover</i> . Essa Figura foi adaptada da Figura 2.5, pag 16, de [PLM08]. . . . .	71
7.1	Resultados da utilização do <b>ReConf</b> (escala logarítmica) . . . . .	79
7.2	Representação dos eventos avaliados: anúncios clicados são exibidos em vermelho nas coordenadas que representam seus valores normalizados de $BID_a$ , $CTR_a$ e $CTX_{a,c}$ . Anúncios não clicados são representados por pontos pretos. . . . .	80
7.3	<b>Primeira rodada</b> - valores normalizados de CTR e eCPM . . . . .	84
7.4	<b>Segunda rodada</b> - valores normalizados de CTR e eCPM . . . . .	84
7.5	Dimensões projetadas nos planos derivados da análise de componentes principais, PCA. . . . .	85
7.6	<b>Experimento 1</b> - Valores estatísticos da função objetivo calculados com base nos 40 indivíduos (máximo, percentil, quartil, mediana e média) durante os 22 dias de execução do experimento com o objetivo de avaliar a abordagem evolutiva . . . . .	89
7.7	<b>Desempenho da melhor solução encontrada no experimento 1</b> - Comparativo dos indicadores de desempenho normalizados, ECPM e CTR, da nova solução e da solução de controle nos 22 dias da segunda fase de avaliação da melhor solução encontrada no experimento 1 . . . . .	90
7.8	<b>Experimento 2</b> - Valores estatísticos da função objetivo (a), percentual de incremento de CTR (b) e percentual de incremento de ECPM (c). . . . .	93
7.9	<b>Desempenho da melhor solução encontrada no experimento 2</b> - Comparativo dos indicadores de desempenho normalizados, ECPM e CTR, da nova solução e da solução de controle nos 22 dias da segunda fase de avaliação da melhor solução encontrada no experimento 2 . . . . .	94
7.10	<b>Experimento 3</b> - Valores estatísticos da função objetivo calculados com base nos 40 indivíduos (máximo, percentil, quartil, mediana e média) durante os 22 dias de execução do experimento com o objetivo de avaliar o resultado de evoluções consecutivas . . . . .	95
7.11	<b>Experimento 4</b> - Valores estatísticos da função objetivo calculados com base nos 40 indivíduos (máximo, percentil, quartil, mediana e média) durante os 22 dias de execução do experimento 4 (b) com o objetivo de reavaliar o resultado do primeiro experimento (a). . . . .	97



# Capítulo 1

## Introdução

A indústria de publicidade na Internet, publicidade digital ou publicidade *online*, continua em grande e rápida expansão acompanhando e muitas vezes superando o crescimento da própria Internet. Essa publicidade é a principal fonte de receita de muitas das grandes empresas de Internet, como o *Google* e o *Yahoo!*. Os gastos mundiais com publicidade *online* em 2013 superaram 120 bilhões de dólares, um crescimento de 14,8% em relação a 2012. As estimativas são de crescimento de 16,7% e 14,3% para 2014 e 2015 e ultrapassando 160 bilhões em 2015, de acordo com o relatório de junho de 2014 do *www.eMarketer.com*, como podemos observar na Figura 1.1.

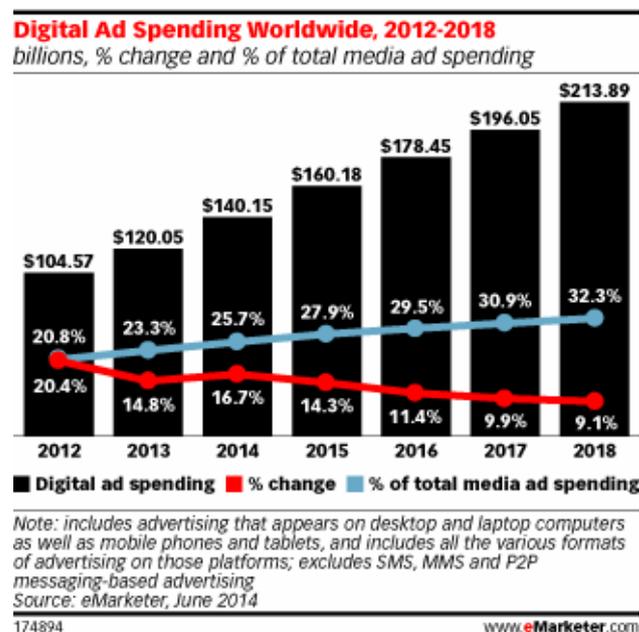


Figura 1.1: Gastos Mundiais com Publicidade Digital, 2012-2018

bilhões de dólares, % de alteração e % do total gasto com publicidades em todas as mídias

Nota: inclui publicidade apresentada em computadores de mesa e laptops assim como telefones e tablets, e inclui todos os diversos formatos de publicidade nessas plataformas; excluindo publicidade de mensagens SMS, MMS e P2P

Fonte: eMarketer, Jun 2014 [eMac]

De acordo com as Figuras 1.2, 1.3 e 1.4, os investimentos totais em publicidade pela Internet no Brasil totalizaram US\$ 2,25 bilhões em 2013, um crescimento de 20% em relação a 2012. O volume pode ser considerado pequeno se comparado com as previsões de gasto dos Estados Unidos, US\$ 58 bilhões em mídia digital e US\$28 bilhões em Internet Móvel. Por outro lado, o Brasil lidera com bastante folga os investimentos na América Latina, mesmo com previsões de crescimento menores que o México, mantendo-se acima do crescimento mundial.

**Top 5 Countries, Ranked by Total Media, Digital and Mobile Internet Ad Spending, 2015**  
billions

	Total media	Digital	Mobile internet
1. US	\$189.06	\$58.61*	\$28.24
2. China**	\$73.13	\$30.81	\$12.14
3. Japan	\$40.19	\$9.61	\$3.37
4. Germany	\$27.71	\$6.67	\$2.11
5. UK	\$25.22	\$12.59*	\$4.67*
<b>Worldwide</b>	<b>\$592.43</b>	<b>\$170.50</b>	<b>\$64.25</b>

Note: \*includes SMS, MMS and P2P messaging-based advertising;  
\*\*excludes Hong Kong  
Source: eMarketer, Dec 2014

182885 www.eMarketer.com

**Figura 1.2: 5 Maiores Países, Ordenados por Gastos com Publicidade em Todas as Mídias, Mídia Digital e Móvel, 2015**  
bilhões de dólares

Nota: \*inclui SMS, MMS e P2P; \*\*excluindo Hong Kong

Fonte: eMarketer, Dez 2014 [eMaa]

**Digital Ad Spending in Latin America, by Country, 2012-2018**  
billions

	2012	2013	2014	2015	2016	2017	2018
Brazil	\$1.87	\$2.25	\$2.88	\$3.31	\$3.87	\$4.41	\$4.94
Mexico*	\$0.50	\$0.66	\$0.90	\$1.10	\$1.31	\$1.54	\$1.82
Argentina	\$0.31	\$0.29	\$0.31	\$0.33	\$0.35	\$0.37	\$0.38
Other	\$0.70	\$0.91	\$1.20	\$1.44	\$1.70	\$1.95	\$2.21
<b>Latin America</b>	<b>\$3.39</b>	<b>\$4.11</b>	<b>\$5.29</b>	<b>\$6.18</b>	<b>\$7.23</b>	<b>\$8.27</b>	<b>\$9.36</b>

Note: includes advertising that appears on desktop and laptop computers as well as mobile phones and tablets, and includes all the various formats of advertising on those platforms; excludes SMS, MMS and P2P messaging-based advertising; numbers may not add up to total due to rounding; \*eMarketer benchmarks its Mexico digital ad spending projections against the IAB México/PwC data for which the last full year measured was 2012  
Source: eMarketer, June 2014

175102 www.eMarketer.com

**Figura 1.3: Gastos com Publicidade Digital na América Latina, por País, 2012-2018**  
bilhões de dólares

Nota: inclui publicidade apresentada em computadores de mesa e laptops assim como telefones e tablets, e inclui todos os diversos formatos de publicidade nessas plataformas; excluindo publicidade de mensagens SMS, MMS e P2P; os valores podem não somar o total adequadamente devido a arredondamentos; eMarketer baseou as projeções de gastos com publicidade digital do México com base nos dados do IAB México/PwC para os quais o último ano completo de medição foi 2012.

Fonte: eMarketer, Junho 2014 [eMab]

**Digital Ad Spending Growth in Latin America, by Country, 2012-2018**  
% change

	2012	2013	2014	2015	2016	2017	2018
Mexico*	34.6%	32.1%	35.3%	22.3%	19.0%	17.5%	18.5%
Brazil	40.2%	20.0%	28.0%	15.0%	17.0%	14.0%	12.0%
Argentina	16.2%	-6.4%	7.0%	7.0%	6.0%	5.0%	4.0%
Other	38.8%	30.2%	31.6%	20.4%	17.5%	15.1%	13.1%
<b>Latin America</b>	<b>36.5%</b>	<b>21.5%</b>	<b>28.5%</b>	<b>17.0%</b>	<b>16.9%</b>	<b>14.5%</b>	<b>13.1%</b>

*Note: includes advertising that appears on desktop and laptop computers as well as mobile phones and tablets, and includes all the various formats of advertising on those platforms; excludes SMS, MMS and P2P messaging-based advertising; \*eMarketer benchmarks its Mexico digital ad spending projections against the IAB México/PwC data for which the last full year measured was 2012*  
Source: eMarketer, June 2014

175109 www.eMarketer.com

**Figura 1.4: Crescimento dos Gastos Mundiais com Publicidade Digital na América Latina, por País, 2012-2018**  
*% de alteração*

*Nota: inclui publicidade apresentada em computadores de mesa e laptops assim como telefones e tablets, e inclui todos os diversos formatos de publicidade nessas plataformas; excluindo publicidade de mensagens SMS, MMS e P2P; os valores podem não somar o total adequadamente devido a arredondamentos; eMarketer baseou as projeções de gastos com publicidade digital do México com base nos dados do IAB México/PwC para os quais o último ano completo de medição foi 2012.*

*Fonte: eMarketer, Junho 2014 [eMab]*

Autores como Shanahan e Kurra se referem a essa área como *Publicidade Digital*, conforme podemos constatar no capítulo *Digital Advertising: An Information Scientist's Perspective* em [MBY11, SK11]. De acordo com Josifovski e Broder [BJ11], o termo *Publicidade Computacional* é uma outra forma de se referenciar a essa área com uma perspectiva científica. Segundo esses autores *Publicidade Computacional* pode ser compreendida como uma sub-disciplina científica na intersecção de busca e análise de texto em larga escala, recuperação de informação, modelagem estatística, aprendizado computacional, teoria dos jogos, teoria dos leilões, classificação, otimização, microeconomia e sistemas de recomendação que visa resolver o seguinte problema:

*encontrar o melhor anúncio dentre um conjunto de anúncios disponíveis para exibir a um usuário em um determinado contexto.*

Analisando os elementos dessa descrição, como apresentado em [BJ11], temos alguns termos-chaves tais como *melhor anúncio*, *conjunto de anúncios disponíveis*, *usuário* e *contexto*, que são definidos como se segue:

- **contexto** - a página onde serão exibidos os anúncios e os demais dados disponibilizados, como palavras-chave da página, categoria da página entre outros;
- **usuário** - a pessoa que está acessando a página por meio de um navegador e as informações sobre essa pessoa, dados demográficos, termos buscados, interesses e comportamento;

- **conjunto de anúncios disponíveis** - anúncios cadastrados por anunciantes no sistema, disponíveis computacionalmente para exibição;
- **melhor anúncio** - qualidade subjetiva do anúncio exibido, que pode variar de acordo com os objetivos e participantes.

Nessa definição, e neste trabalho, utilizamos o termo contexto para referenciar o conjunto de características textuais e sua relação com o página e local onde será exibido o anúncio. Por sua vez, utilizamos o termo usuário para referenciar o conjunto de características associadas à pessoa para qual será exibido o anúncio, como dados demográficos (ex: sexo, faixa etária) e interesses ou comportamentos (ex: usualmente acessa conteúdos sobre economia e esportes), ao invés de usuário poderíamos ter utilizado *contexto de usuário*, mas nesse caso deveríamos utilizar para o primeiro termo algo como *contexto de página*. Dessa forma, preferimos manter apenas os termos **contexto** para características textuais do local de exibição e **usuário** para características associadas à pessoa para qual será exibido o anúncio.

Neste texto, utilizaremos o termo **publicidade computacional** para nos referenciar a esse assunto. Estudaremos a área de publicidade computacional, analisando os interesses dos participantes desse mercado e os modelos existentes, nos aprofundando nos aspectos de seleção e ordenação de anúncios. Para tornar mais claro nosso estudo nas próximas seções definiremos mais precisamente o escopo do trabalho.

## 1.1 Problema

Neste trabalho, assumiremos que o principal desafio de publicidade computacional pode ser compreendido como um problema de adaptação de conjuntos. Temos o conjunto de anúncios e o conjunto de pares usuário e contexto. Desejamos encontrar uma função ou mapeamento entre os elementos dos dois conjuntos. O problema poderia ser abordado dos dois lados, mas usualmente temos como entrada o usuário e o contexto, fornecidos na requisição, e devemos selecionar os elementos associados do conjunto de anúncios. Identificar apenas que existe uma relação, associação ou relevância entre os elementos de conjuntos diferentes não é suficiente, pois, pelo problema original, queremos encontrar o *melhor* elemento, ou seja, o mais relevante ou mais fortemente relacionado a entrada. A partir da observação de que a quantidade de elementos desejada é finita e consideravelmente pequena (menor que uma dezena de anúncios, usualmente de um a três) quando comparada a grande quantidade de elementos disponíveis, podemos restringir o problema que abordaremos neste trabalho à seguinte questão:

*como encontrar uma função de ordenação que permita exibir os melhores anúncios a um usuário em um determinado contexto, considerando o conjunto de anúncios disponíveis?*

A maior subjetividade do problema está na definição do que são os melhores anúncios. De acordo com o problema queremos encontrar uma função de ordenação que identifique e posicione em primeiro lugar o melhor anúncio disponível. A avaliação da função de ordenação, identificando se ela resolve ou não o problema, depende da clareza na definição de qual é o melhor anúncio, o segundo melhor, e assim sucessivamente.

Por outro lado, podemos considerar que os melhores anúncios são aqueles que ao serem exibidos otimizam o valor para os participantes. Dessa forma, podemos avaliar a função de ordenação comparando a utilidade que os participantes alcançaram. Como indicado na seção 2.1, as funções de utilidade dos participantes são possivelmente contraditórias, ou seja, é importante ressaltar que otimizar a utilidade para um participante pode reduzir a utilidade para os demais. Para isso, podemos utilizar uma função objetivo que represente a utilidade total do sistema em um determinado cenário real ou técnicas de otimização de múltiplos objetivos. Dessa forma, podemos redefinir o nosso problema da seguinte maneira:

**como encontrar uma função de ordenação para exibir anúncios a um usuário em um determinado contexto que otimize a utilidade do sistema para seus participantes?**

Precisamos considerar a utilidade de cada participante: o usuário, o anunciante e o publicador. O interesse dos usuários por um anúncio pode ser representado pelo **CTR** (*click-through rate*) do anúncio, a taxa de cliques, como descreveremos na equação 2.1, é a quantidade de cliques obtidos pelo anúncio dividida pela quantidade de impressões registradas para o anúncio. A utilidade do anunciante pode ser representada pelo **ROI** (*return on investment*), retorno sobre o investimento do anúncio, de acordo com a equação 2.3, é a razão do retorno financeiro do anunciante obtido pelo anúncio sobre o investimento gasto com esse anúncio. Já o publicador tem como objetivo rentabilizar a audiência em seus contextos, o **eCPM** (*effective cost per mile*), custo efetivo por milhar de impressão, como descreveremos na equação 2.4, mil vezes o custo do anúncio dividido pela quantidade de impressões. O eCPM pode ser utilizado como um estimador da receita pois, para um eCPM fixado, quanto maior a audiência, maior será o volume de impressões e a multiplicação do eCPM pelo volume de impressões fornece uma estimativa de receita.

A definição desses indicadores pode ser feita para a unidade do anúncio, ou seja, o CTR do anúncio, o ROI do anúncio e o eCPM do anúncio. Podemos ampliar essa definição para todo o

sistema, utilizando os dados de todo o sistema em um determinado período de tempo para calcular os valores, por exemplo, o CTR do sistema em um dia será a quantidade de cliques obtidos por todos os anúncios dividida pela quantidade de impressões registradas para todos os anúncios naquele dia.

## 1.2 Contextualização

Para resolver nosso problema a partir de agora vamos restringir nosso escopo a anúncios de texto para publicidade contextualizada no modelo de **CPC** (*cost per click*), seguindo as definições apresentadas no capítulo 2, *Fundamentos*.

Encontrar uma função de ordenação que resolva o problema é o nosso objetivo geral. Diferentes áreas têm produzido resultados nesse assunto, começando por economia e teoria dos jogos, com os modelos de leilão, passando por recuperação de informação, com funções de relevância e similaridade, por modelos estatísticos e de probabilidade com estimadores da taxa de clique e mais recentemente com utilização de sistemas de recomendação. O aprendizado de máquina tem sido aplicado em diversas dessas áreas obtendo resultados interessantes como em [LCG<sup>+</sup>06], [CMP08] e [SGV11].

O que pudemos perceber durante nosso levantamento bibliográfico é que cada uma dessas iniciativas desenvolveu uma dimensão do problema. Os modelos de leilão na dimensão financeira das apostas e no cálculo do CPC. A recuperação de informação na contextualização do anúncio com a página ou a consulta. Modelos estatísticos e de probabilidade na previsão do CTR. Por fim, sistemas de recomendação no comportamento do usuário.

Individualmente em cada uma dessas áreas os resultados obtidos para ordenação de anúncios são significativos, como em [LCG<sup>+</sup>06, BGJM10]. No melhor da nossa pesquisa bibliográfica não encontramos como combinar essas diferentes soluções em cada uma das dimensões do problema.

Outro aspecto evidenciado é que, pelo que acreditamos com base em nosso levantamento bibliográfico, não é possível, ou será muito difícil, encontrar uma função de ordenação universal que combine todas essas dimensões e seja ótima para qualquer sistema de publicidade computacional com quaisquer conjuntos de anúncios, usuários e contextos.

Observado isso, o objetivo deste trabalho é muito mais modesto. Com nossa experiência em ciência de dados, esperamos descrever uma abordagem que permita construir um sistema de publicidade computacional que encontre e evolua uma boa função de ordenação com base nos dados observados, uma função de ordenação capaz de combinar as diferentes dimensões do problema. Vamos considerar as seguintes dimensões:

- **valor comercial** - medida do potencial de geração de receita do anúncio, fortemente relaci-

onado ao valor da aposta (valor máximo que um anunciante deseja pagar por um clique no anúncio como descreveremos na seção 2.4), mas pode também considerar outros fatores como o orçamento da campanha, ou o saldo do anunciante;

- **desempenho histórico** - medida do resultado do anúncio quando exibido anteriormente, associado ao CTR global do anúncio no sistema, pode eventualmente utilizar CTRs mais específicos como o CTR para o contexto ou o CTR para o usuário, ou outro indicador como o ROI do anúncio;
- **contexto** - medida de quanto o anúncio é relevante para o contexto, usualmente uma medida de relevância ou similaridade entre o anúncio e contexto (página, consulta, etc);
- **usuário** - medida de quanto o anúncio é recomendado para o usuário, uma medida de relevância que pode utilizar características demográficas ou classificação de usuários para recomendar anúncios com características ou comportamentos semelhantes.

Utilizaremos como base a ordenação por ECPM, ordenação histórica por eCPM que apresentaremos na equação 2.7, descrita na seção 2.8. Alguns trabalhos [BF02] abordam sobretudo o valor comercial. Inicialmente vamos fixar essa dimensão na ordenação como primeiro fator de nossas funções, como na fórmula de ECPM. Isso permitirá não só a manutenção da precificação pelo leilão de segundo preço, como também a validade da fórmula de CPC, equação 2.8. Combinaremos as demais dimensões para calcular um valor de qualidade com base nos dados observados, estabelecendo um substituto da probabilidade de clique, com base no que descreveremos em mais detalhes na seção 2.8.

Utilizaremos a dimensão de desempenho histórico para representar o quanto um anúncio foi capaz de atrair o interesse dos usuários quando exibido. Quanto mais cliques o anúncio receber quando exibido anteriormente, maior será nosso valor da dimensão de desempenho histórico, representando o interesse do usuário pelo anúncio por meio do clique. Nessa dimensão utilizaremos o CTR global do sistema.

Diversos trabalhos [NH97, WZCD02] evidenciam que anúncios mais relevantes ou similares ao texto têm maior chance de atrair o interesse do usuário. Nossa dimensão de contexto será maior quanto mais palavras-chave e categorias um anúncio compartilhar com a página, representando o aumento da relevância do anúncio para o texto da página. Trabalhos como [BGJM10, LCG<sup>+</sup>06] utilizaram abordagens que buscavam otimizar essa relevância com o contexto para selecionar e ordenar anúncios.

Trabalhos mais recentes [TLY<sup>+</sup>11, LPAJ12, APJ12, CZA<sup>+</sup>12, LODL12, PDH<sup>+</sup>12, FB12, TPGJ11] têm explorado a dimensão do usuário e seu comportamento, procurando estimar a probabilidade de clique com base no comportamento prévio do usuário e associando o histórico de interação de outros usuários com o anúncio com comportamento semelhante ao usuário para qual se deseja exibir um anúncio, dessa forma, o problema é abordado com uma perspectiva mais próxima dos sistemas de recomendação. Apesar de estarmos conscientes dos bons resultados apresentados por essa linha de pesquisa, iremos abordar essa dimensão sob a perspectiva de recuperação de informação, utilizando assim a mesma abordagem da dimensão de contexto, permitindo uma comparação mais próxima entre ambas.

Um outro aspecto, bem descrito em [TPGJ11], é a maior eficiência do modelo de recuperação de informação em um conjunto muito grande de dados de usuário quando comparado com um modelo de sistemas de recomendação. Assim, os interesses do usuário serão mapeados como um conjunto de palavras-chave e categorias. Nossa dimensão de usuário será maior quanto mais palavras-chave e categorias um anúncio e um usuário compartilharem, representando o aumento de relevância do anúncio para os interesses do usuário.

Considerando essas quatro dimensões do problema, as restrições de escala e tempo de resposta que discutiremos na seção 2.5 e a técnica de reordenação de anúncios que descreveremos na seção 5.1.9, podemos descrever nosso objetivo principal.

### 1.2.1 Objetivo Principal

Conscientes das restrições de escala e tempo de resposta impostas aos sistemas de publicidade computacional e que diferentes trabalhos podem apresentar melhores soluções para uma das dimensões descritas, vamos considerar um sistema com pelo menos duas fases de ordenação para alcançar nosso principal objetivo, assim definido:

**estabelecer uma abordagem para encontrar e evoluir uma função de reordenação de anúncios com base nos dados observados que combine as dimensões de valor comercial, desempenho histórico, contexto e usuário para aumentar o valor de utilidade do sistema.**

## 1.3 Hipótese

Temos evidências pelos resultados de trabalhos já citados como [LCG<sup>+</sup>06] e [CMP08] que o aprendizado de máquina é um bom campo a ser utilizado como arcabouço da nossa abordagem.

Além disso, acreditamos que as dimensões descritas anteriormente são necessárias para representar o problema. Como buscamos encontrar e evoluir uma função baseada nos dados, defendemos que uma abordagem evolutiva baseada em aprendizado de máquina como motor de um sistema de publicidade computacional adaptativo será capaz de alcançar nosso objetivo. Podemos, então, definir nossa hipótese da seguinte maneira:

**a abordagem evolutiva permite não apenas encontrar, como também evoluir, funções de reordenação de anúncios com base nos dados observados que combinem as dimensões de valor comercial, desempenho histórico, contexto e usuário para aumentar o valor de utilidade do sistema, permitindo que o sistema se adapte quando modificações no ambiente forem introduzidas.**

## 1.4 Justificativa da Hipótese

O ambiente de publicidade computacional está em constante modificação. Novos contextos, novos anúncios e novos interesses surgem e se modificam em pouco tempo. Mudanças podem ocorrer em poucas horas ou dias e modificações mais significativas no comportamento dos participantes em semanas. Nesse ambiente, nem as melhores funções de reordenação são capazes de resistir por muito tempo. Dessa forma, a utilização de uma abordagem evolutiva em um sistema capaz de se adaptar tem como objetivo encontrar funções de ordenação que sejam adequadas para o ambiente e evoluir essas funções para aumentar a utilidade do sistema e adaptar o sistema às modificações de comportamento dos participantes.

Historicamente o problema de publicidade computacional tem sido abordado com foco nas suas diferentes dimensões. As dimensões de valor comercial, desempenho histórico, contexto e usuário têm sido utilizadas com sucesso para selecionar e exibir anúncios em publicidade computacional. Observamos que cada uma dessas dimensões pode representar características importantes do problema. Optamos por agrupar as características dentro dessas quatro dimensões. Considerado o histórico de utilização, acreditamos que essas dimensões são necessárias para representar o problema. Um dos nossos objetivos é validar essa parte da hipótese analisando o impacto do desempenho comercial de cada uma dessas dimensões, assim como, a correlação entre elas. Apesar de defendermos que essas dimensões são necessárias, nós não podemos afirmar que elas são suficientes. Outras características que influenciam o problema podem ser reconhecidas e talvez não seja adequado agrupá-las dentro dessas dimensões. Por essa razão, a nossa abordagem deve ser construída de forma a permitir que novas dimensões sejam incluídas no modelo quando necessário.

Nossa abordagem é inspirada por técnicas de computação evolucionária. Computação evolucionária são técnicas de aprendizado de máquina inspiradas pela evolução biológica [PLM08], fortemente associadas à mineração de dados. Optamos por utilizar como motor da evolução a técnica de Programação Genética (GP). GP se propõe a resolver automaticamente problemas encontrando soluções otimizadas para certas características do problema.

Uma definição mais abrangente de GP pode ser encontrada em [PLM08]:

“Programação genética é um método *sistemático, independente de domínio* para que computadores resolvam problemas *automaticamente* a partir de uma proposição de alto-nível do que precisa ser feito.”

Programação genética é normalmente utilizada para aproximar relacionamentos complexos e não-lineares. Devido ao seu mecanismo de busca inerentemente paralelo de soluções e sua poderosa capacidade de exploração de soluções globais em um espaço de alta-dimensão, tem sido utilizada para resolver uma grande variedade de problemas difíceis de otimização.

Na sua formulação mais abrangente a programação genética evolui uma população de *programas de computadores*. A cada geração, a programação genética *estocasticamente* transforma populações de programas em novas, potencialmente melhores, populações de programas. Assim como a natureza, é um processo aleatório, sem garantias de resultado. Por outro lado, essa aleatoriedade é o que permite a programação genética escapar de armadilhas que outros métodos determinísticos têm dificuldade de enfrentar, como múltiplos ótimos locais. Apesar da generalidade para programas, a programação genética historicamente obteve melhores resultados na aproximação de funções. Funções podem ser facilmente representadas como árvores e executadas, ou avaliadas, diretamente sobre um conjunto de dados de entrada. Como as respostas costumam ser numéricas, a criação da função de *fitness* não costuma ser um grande problema. Essa função de *fitness* é parte essencial da programação genética como apresentaremos na seção 2.11.

A programação genética já produziu um grande conjunto de resultados nas mais diversas áreas. A literatura, que já cobre mais de 5000 registros de uso de programação genética, relata um enorme número de aplicações nas quais a programação genética tem sido bem sucedida como uma ferramenta de programação automática, uma ferramenta de aprendizado de máquina ou um mecanismo de resolução de problemas de forma automática. Em [PLM08] é apresentada uma lista de propriedades de áreas nas quais a programação genética tem sido bem sucedida, apresentamos uma discussão sobre seis dessas propriedades que acreditamos estar presentes no nosso problema na seção *Programação Genética 2.11*, do capítulo de fundamentos, 2.

Um aspecto de GP foi o principal responsável por nossa opção de utilizá-la. Para que a GP encontre soluções não é necessário que o usuário conheça ou especifique a forma ou estrutura da solução, permitindo assim explorar funções de reordenação sem assumir ou especificar sua forma, como é necessário ao se optar por outras técnicas, como regressões tradicionais. Dessa forma, GP permite descobrir modelos matemáticos de forma livre, não necessariamente formas lineares. Esse método de utilização de GP é conhecido como regressão simbólica, uma alternativa atraente às regressões tradicionais devido sua flexibilidade em criar esses modelos matemáticos de forma livre. A maioria dos algoritmos de aprendizado de máquina foram desenvolvidos para criar modelos lineares e, portanto, possuem menos flexibilidade que GP. A motivação por explorar modelos matemáticos de forma livre vem dos modelos e fórmulas estáticas normalmente utilizados em publicidade computacional, por exemplo, as fórmulas utilizadas como referência (*baseline*) dos experimentos apresentados neste trabalho não são lineares. O recente trabalho [IB13] defende o posicionamento da regressão simbólica por GP como a principal técnica de resolução desse grupo de problemas, propondo melhorias e avaliando alternativas que utilizam modelos determinísticos de aprendizado de máquina, como [McC11]. Assim sendo, esse aspecto foi decisivo na opção por utilizar GP em detrimento de outras técnicas de mineração que possuem convergência mais rápida, como as regressões.

## 1.5 Resumo da proposta

Neste trabalho buscamos desenvolver uma nova abordagem que não apenas fosse capaz de explorar e avaliar soluções candidatas de ordenação de anúncios, mas que também permitisse a evolução e adaptação de soluções ao longo do tempo. Nossa abordagem estabelece um modelo cíclico, composto por um conjunto não-prescritivo de passos que permitem a *Exploração* de experimentos, *Avaliação* dos resultados e *Evolução*, ou mesmo, a criação de novas soluções. A construção da abordagem de ordenação evolutiva teve como alicerce um sistema de publicidade adaptativo. Por sua vez, para desenvolver a capacidade de adaptação criamos como base uma arquitetura de reconfiguração para o sistema de publicidade computacional.

## 1.6 Contribuições

A principal contribuição deste trabalho foi definir e implementar a abordagem evolutiva como um ciclo composto por três passos principais responsáveis por **explorar** funções de reordenação de anúncios, **avaliar** o desempenho dessas funções comparando com a função de controle e **evoluir**, ou

mesmo criar, novas funções com base nas melhores funções anteriormente avaliadas. Para a implementação da abordagem foram necessários aprimoramentos na arquitetura do sistema de veiculação de anúncios e a criação de um modelo de reconfiguração. Dedicamos uma grande parte do projeto e deste texto na implementação e descrição das soluções de sistemas de software que permitiram incorporar a capacidade evolutiva em um sistema real de publicidade computacional. Isso se demonstrou um pré-requisito muito desafiador de nossa proposta, que agregou valor e conhecimento ao nosso projeto.

Como uma segunda contribuição, nós apresentamos uma análise detalhada dos blocos básicos com os quais construímos as funções de reordenação e que sustentam a nossa abordagem, ou seja, uma análise detalhada das diferentes dimensões do problema de publicidade computacional: valor comercial, desempenho histórico, contexto e usuário [CBMF14]. Com base nessa análise, nós podemos afirmar que todas as dimensões do problema de publicidade computacional se complementam mutuamente e nenhuma delas deve ser negligenciada para solucionar o problema de reordenação de anúncios.

Como contribuições adicionais nós desenvolvemos diferentes componentes de software que disponibilizamos para a comunidade como software livre de forma a auxiliar outros grupos na construção de experimentos e no desenvolvimento de sistemas evolutivos em ambientes reais [BMF14]. Entre esses componentes nós destacamos o **ReConf**, uma inovadora solução que disponibiliza um repositório de configurações centralizado como um serviço web RESTful.

## 1.7 Organização do texto

Este texto segue organizado da seguinte forma: no capítulo 2, *Fundamentos*, apresentamos os principais conceitos utilizados no trabalho; no capítulo 3, *Trabalhos relacionados*, apresentamos o levantamento bibliográfico e discutimos a sua relação com este trabalho; no capítulo 4, *Proposta: Abordagem evolutiva em publicidade computacional*, apresentamos a nossa abordagem para resolver o problema proposto. No capítulo 5, *Engenharia e arquitetura*, descrevemos o sistema e o ambiente do projeto e a implementação da capacidade adaptativa, detalhando o ReConf e como ele é utilizado para oferecer reconfiguração ao sistema. No capítulo 6, *Implementação da ordenação evolutiva*, descrevemos a implementação completa de nossa abordagem utilizando o ReConf e um motor de aprendizado para evoluir soluções de ordenação de anúncios no sistema de publicidade computacional. No capítulo 7, *Resultados*, descrevemos os experimentos realizados e os resultados obtidos; concluímos o texto no capítulo 8, *Conclusão*, e descrevemos os pontos de extensão que poderão ser

explorados em trabalhos futuros.

## Capítulo 2

# Fundamentos

Podemos descrever brevemente como a mecânica da exibição de anúncios funciona com base no texto de Weideman [WHS02]. Uma requisição de anúncios ocorre quando um usuário faz uma busca ou acessa uma página com espaços publicitários. Ele obtém uma página com o conteúdo desejado que inclui também anúncios. Os anúncios são claramente distintos dos resultados de buscas ou conteúdo da página, normalmente destacados por marcações visuais, cores e textos que identificam sua característica publicitária. A forma mais comum, mas não a única como veremos mais adiante, de direcionar os anúncios é por meio de palavras-chave. Os anunciantes associam ao anúncio uma palavra, por exemplo, um anúncio de uma empresa aérea associa a palavra *Florianópolis*. Quando essa palavra for pesquisada por um usuário em um mecanismo de busca, ou aparecer no texto de uma página de conteúdo para a qual o usuário navegou, o anúncio (texto, imagem, vídeo, ..) associado à palavra será exibido. Quando um usuário clica em um desses anúncios, ele é redirecionado para a página de destino associada, no exemplo, a página da empresa aérea.

### 2.1 Participantes

A publicidade computacional pode ser descrita como uma relação entre o anunciante, o publicador e o usuário. O **anunciante** possui interesse em anunciar um produto ou serviço; o **publicador** possui contextos e audiência e deseja exibir anúncios para obter receita; e o **usuário** acessa os contextos na Internet para obter informação. Podemos explicitar mais claramente os valores de cada um dos participantes desse mercado:

- o usuário acessa a Internet com a intenção de obter informações, esperando ter uma boa navegação e tem potencial de adquirir algo;

- um anunciante disponibiliza uma informação, bem ou serviço, que deseja fornecer ou vender a um usuário e possui recursos financeiros para investir em publicidade, esperando receber acessos e vendas, ou conversões, a partir dessa publicidade;
- o publicador exerce o papel de intermediário, possuindo audiência em suas páginas ou mecanismos de busca e deseja aumentar a receita exibindo publicidade e aumentar a audiência atraindo mais usuários.

Dessa forma, o problema de exibir anúncios em um contexto é semelhante ao problema de integrar outros tipos de informação. Encontrar o *melhor anúncio* é um tipo de problema de recuperação de informação com múltiplas, possivelmente contraditórias, funções de utilidade. Otimizar diferentes e, possivelmente conflitantes, objetivos é um aspecto de publicidade computacional que continua sendo estudado, como no artigo [OLL<sup>+</sup>14].

## 2.2 Tipos de anúncios

Existem diferentes tipos de anúncios utilizados em publicidade computacional [BJ11], dois deles historicamente ocupam a maior parte do mercado de publicidade:

- **anúncios visuais**, *display ads*, ou anúncios de *display*, são anúncios exibidos em formatos ricos que combinam diferentes aspectos como texto, *link*, imagem, som, movimento, interação, vídeo e outros. Os formatos mais comuns são *banners*, imagens de diferentes tamanhos combinando texto e outras imagens publicitárias. Esses anúncios são responsáveis pela maior parcela da receita publicitária computacional;
- **anúncios de texto**, são anúncios que usualmente possuem texto (título e descrição) e um *link* para o site do anunciante. Esses anúncios se tornaram muito populares nos mecanismos de busca.

Além disso, também se utiliza a divisão entre os tipos de contexto:

- **busca patrocinada**, *sponsored search*, quando a publicidade é exibida em uma página de resultados em um sistema de busca na Internet;
- **publicidade contextualizada**, *contextual advertising*, quando a publicidade é exibida na mídia *online* do publicador, como páginas da Internet com conteúdo de interesse para o usuário, portais e aplicações *online/offline/mobile*.

De modo geral, os anúncios visuais são mais comuns na publicidade contextualizada e anúncios de texto na busca patrocinada. Recentemente pode-se perceber um aumento do mercado para anúncios de texto em publicidade contextualizada, no entanto, a combinação de anúncios visuais com busca patrocinada ainda está começando a ser explorada.

## 2.3 Modelos de pagamento

A monetização dessa publicidade na maioria dos casos é feita de uma das seguintes formas:

- **CPM - custo por mil impressões**, *cost per mille*, o anunciante paga ao publicador um valor fixo para cada mil impressões, a quantidade de impressões pode ser combinada e garantida em um período de tempo, conhecida como *entrega garantida*, ou a entrega pode ser feita sob demanda com o critério de seleção e ordenação definido pelo publicador e o anunciante pagando pelo que for veiculado, conhecida como *entrega não-garantida*;
- **CPC - custo por clique**, *cost per click* ou **PPC**, pagamento por clique (*pay-per-click*), o anunciante paga ao publicador cada vez que o usuário clica no anúncio e é direcionado para o destino, [LCG<sup>+</sup>06] e [BGJM10];
- **CPA - custo por aquisição/ação**, *cost per action*, o anunciante paga ao publicador quando após o clique e o direcionamento, o usuário realiza uma ação ou aquisição no local de destino, como um cadastro ou a compra de um produto.

O modelo de CPM impõe ao anunciante um alto risco de retorno sobre o investimento, ROI (*return on investment*), uma vez que impressões de um anúncio podem não resultar em engajamento ou compras. O modelo de CPC foi empregado pela primeira vez em 1997 pela Goto.com (depois adquirida pela Overture, que foi posteriormente adquirida pelo Yahoo!). Em 2002 o modelo foi revolucionado pelo Google ao adicionar um componente de qualidade [EOS<sup>+</sup>05]. Com isso o clique passou a ser um representante do interesse do usuário pelo anúncio e o risco passou a ser compartilhado entre o anunciante e o publicador. O anunciante não paga por impressões que não geram valor e o publicador só recebe se o usuário interagir com o anúncio. A história nos mostrou que nem todos os cliques geram valor para o anunciante, não se convertendo em ações úteis, como uma compra, e que fraudes são um problema para o ROI do anunciante, pois é fácil gerar cliques fraudulentos. O modelo de CPA vincula o pagamento à ação que gera o valor para o anunciante, eliminando a diferença entre o clique e a geração de valor. Esse modelo dificulta fraudes, mas trans-

fere a maior parte do risco para o publicador, pois os usuários podem não gerar valor apesar de um grande volume de publicidade ser exibido.

Modelos híbridos que combinam restrições como um valor máximo para o CPM médio e uma meta de CPA já foram construídos permitindo que o risco seja melhor compartilhado entre o anunciante e o publicador.

Anúncios visuais são na maioria monetizados no modelo de CPM, mas isto vem se modificando nos últimos anos após o surgimento das redes de anúncios e de *RTB - real-time bidding*, nos quais o modelo de CPC tem sido muitas vezes o utilizado [ZYW14]. Anúncios de texto normalmente utilizam o modelo de CPC. O modelo de CPA é utilizado para os dois formatos e faz muito sucesso em ferramentas como o *AdSense* do *Google* que permite otimizar a entrega buscando melhorar a quantidade de aquisições ou ações. Os modelos híbridos têm sido vistos em sistemas intermediários como *ad networks*, *domain-side platform* (DSPs) e *ad exchanges*.

## 2.4 Anatomia de um anúncio

Em [BGJM10] os autores se concentram na anatomia de anúncios de texto. Anúncios visuais costumam ter um conjunto semelhante de partes, com a inclusão de partes gráficas, por vezes substituindo componentes textuais. Segundo os autores, nos anúncios textuais podemos identificar as seguintes partes visíveis:

- **título** - um título do anúncio, normalmente evidenciado por fontes, estilos ou cores;
- **descrição** - uma ou duas linhas de texto que explicam o anúncio assim como um *snippet* descreve um resultado de uma busca, pode incluir um *call to action*, palavra ou frase de ação como, *Veja aqui!* ou *Compre agora!*;
- **URL de exibição** - URL que representa e identifica o destino ou anunciante.

Existem variações como uma imagem auxiliar, uma área específica para um *call to action*, para um preço, entre outras. As três partes visíveis acima são as mais comuns de serem encontradas. Um anúncio também contém partes invisíveis:

- **URL de destino** - também conhecida como *landing page*, página de destino, é a página para qual o usuário será redirecionado ao clicar no anúncio;
- **palavras-chave** - palavras ou frases para as quais o anunciante deseja que seu anúncio seja exibido, quando informadas em um sistema de busca ou identificadas em um contexto ou página;

- **categorias** - são assuntos identificados em uma taxonomia comercial para os quais o anunciante deseja que seu anúncio seja exibido, em resultados de buscas ou contextos identificados como desse mesmo assunto;
- **restrições de usuário** - outras restrições relacionadas ao usuário que possam ser obtidas no momento da requisição, como sexo ou idade, ou mesmo restrições indiretas como geolocalização ou dispositivo de acesso (celular, navegador, S.O., etc);
- **aposta** - valor máximo pago no modelo de pagamento para uma exibição ou interação com o anúncio, no modelo de CPC, seria o máximo valor que o anunciante pagaria por um clique no anúncio.

Em diversos modelos a aposta está associada a cada palavra-chave ou categoria. Neste trabalho vamos organizar essas partes em dois grupos:

- **descritivo** - representa diferentes formas de anunciar e é composto pelas partes de interação com usuário, todas as partes visuais, **título**, **descrição** e **URL de exibição**, juntamente com a **URL de destino**;
- **segmentação** - restringe o contexto e usuário para o qual quer se anunciar e é composta pelas partes que representam critérios de seleção do anúncio, **palavras-chave**, **categoria** e **restrições de usuário** associadas a um único valor de **aposta** para todo o conjunto de critérios exibirem o anúncio.

Os anúncios são normalmente definidos e organizados como bases de dados estruturadas com diversos tipos de entidades. Cada **anunciante** possui uma ou mais **contas**. Cada conta possui uma ou mais **campanhas**, que têm diferentes orçamentos, temas e durações (ex: ofertas especiais de Natal). Campanhas são compostas por **grupos de anúncios** que agrupam múltiplos **descritivos** e **segmentações**. Qualquer descritivo pode ser pareado com qualquer segmentação do mesmo grupo de anúncio, esse par exibido ao usuário é o que chamamos de anúncio. Os sistemas costumam permitir em um grupo de anúncio dezenas de descritivos e segmentações e milhares de critérios. Nesse modelo limitamos uma aposta para cada segmentação, isso não é uma restrição a modelos que a aposta é associada à palavra-chave, por exemplo, pois podemos representar cada associação palavra-chave e aposta, como uma segmentação diferente.

Assim podemos reformular o problema original de publicidade computacional nessa estrutura da seguinte forma:

*Encontrar o melhor par <descritivo, segmentação> dentre um conjunto de pares disponíveis para exibir a um usuário em um determinado contexto.*

## 2.5 Sistema de publicidade computacional

Um sistema de publicidade computacional deseja selecionar e exibir anúncios em resposta a uma requisição. Na busca patrocinada essa requisição é composta pela consulta do usuário ao sistema de busca e o sistema de publicidade tem como objetivo satisfazer a necessidade de informação do usuário enquanto, paralelamente, entrega a mensagem do anunciante, considerando a publicidade como uma forma de informação. Na publicidade contextualizada o processo é semelhante, mas a necessidade de informação não é fornecida pelo usuário e precisa ser inferida pelo contexto, como o conteúdo e assunto da página, e pelo comportamento usuário, representado por atributos diretos, como sexo e idade, por atributos indiretos, geo-localização, e por interesses, palavras-chave ou categorias, identificadas pelo histórico de navegação ou interação do usuário.

Esses sistemas estão submetidos a fortes restrições de escala e custo, usualmente trabalhando com bilhões de anúncios disponíveis, bilhões/trilhões de requisições/impressões por hora e bilhões de usuários únicos em um mês. Eles precisam responder em tempos inferiores a 100 ms para não prejudicar a experiência dos usuários que *aguardam* enquanto os anúncios são selecionados e exibidos. Servir cada requisição consome recursos de CPU e os dados costumam ficar em memória. Isso tudo agravado pelas baixas taxas de clique, que em publicidade contextualizada chegam a ser inferiores a 0,1%, fazem com que o objetivo de tornar o sistema viável se apresente como um problema desafiador.

Para resolver esse desafio os sistemas de publicidade computacional utilizam teorias e técnicas semelhantes às utilizadas por sistemas de busca. Um sistema de busca tradicional opera na seguinte ordem: *web crawling*, indexação, ordenação dos documentos para as consultas do usuário e geração da página de resultados. Na primeira etapa a página é salva localmente e tem seu texto analisado e as características extraídas, na sequência um índice é construído normalmente utilizando a estrutura de dados de recuperação de informação conhecida como lista invertida, uma estrutura que armazena associada a uma chave (normalmente uma palavra), uma lista de elementos que contem características como o documento e a frequência que a palavra ocorre, como um índice remissivo ao final de um livro. Quando uma consulta é submetida por um usuário, ela é analisada, as características são extraídas e um *score* ou *ranking* é calculado entre a consulta e os documentos indexados. Os documentos são então ordenados por esse valor e os primeiros são então exibidos como resultado

da busca.

A funcionalidade de sistemas de publicidade computacional podem ser fortemente mapeadas para a funcionalidade de um sistema de busca tradicional, diferindo de forma sutil mas desafiadora. As etapas podem ser mapeadas na seguinte ordem: *landing page crawling*, indexação de anúncios, ordenação de anúncios e exibição dos anúncios.

Alguns sistemas utilizam os termos e características da página de destino para enriquecer os dados sobre o descritivo e/ou o grupo de anúncio, nesse caso essa página precisa ser gravada e analisada. No contexto de publicidade computacional podemos generalizar o uso de listas invertidas para indexação, utilizando como chaves mais do que palavras do texto. A chave pode ser estendida para qualquer característica que represente o contexto, o usuário ou o anúncio (ex: categoria do anúncio ou página, a cidade de acesso do usuário, uma palavra que ocorre no título do anúncio, uma palavra-chave selecionada pelo anunciante em uma segmentação, etc). O valor associado a cada chave costuma ser uma lista ligada de elementos. Um elemento pode ser simplesmente uma identificação de um anúncio indicando que a característica ocorre ou foi associada ao anúncio e um determinado peso dessa relação. Esse peso costuma ser um valor de intensidade da relação, como a quantidade de vezes que uma palavra ocorreu no documento (frequência). As características extraídas de um anúncio podem começar com termos encontrados ou gerados a partir do descritivo ou das palavras-chaves associadas pelo anunciante a uma segmentação e qualificados de acordo com o local que foram encontrados, como no título, na descrição ou na palavra-chave. Outras características podem ser obtidas diretamente da segmentação, como categorias, geo-localização ou idade. A exibição do anúncio é semelhante a exibição de resultados de busca. Já a ordenação necessita do cálculo de um *score* ou *ranking* que pode ser baseado em uma função definida manualmente ou em estratégias de aprendizado de máquina.

## 2.6 Dimensões e análise multivariada

Recuperando o que definimos anteriormente iremos considerar as seguintes dimensões:

- **valor comercial** - medida do potencial de geração de receita do anúncio, fortemente relacionado ao valor da aposta, mas pode também considerar outros fatores como o orçamento da campanha, ou o saldo do anunciante;
- **desempenho histórico** - medida do resultado do anúncio quando exibido anteriormente, associado ao CTR global do anúncio no sistema, pode eventualmente utilizar CTRs mais

específicos como o CTR para o contexto ou o CTR para o usuário, ou outro indicador como o ROI do anúncio;

- **contexto** - medida de quanto o anúncio é relevante para o contexto, usualmente uma medida de relevância ou similaridade entre o anúncio e contexto (página, consulta, etc);
- **usuário** - medida de quanto o anúncio é recomendado para o usuário, uma medida de relevância que pode utilizar características demográficas ou classificação de usuários para recomendar anúncios com características ou comportamentos semelhantes.

Como também já comentamos, decidimos por agrupar as características do problema de publicidade computacional nessas quatro dimensões. Considerado o histórico de utilização, acreditamos que essas dimensões são necessárias para representar o problema. Um dos nossos objetivos é validar essa parte da hipótese analisando o impacto do desempenho comercial de cada uma dessas dimensões, assim como, a correlação entre elas.

### 2.6.1 Análise de componentes principais

Para analisarmos a correlação utilizamos a técnica de PCA (*Principal Component Analysis*). PCA [Jol05] é um procedimento estatístico usado em análise multivariada e exploratória de dados que utiliza uma transformação ortogonal para converter um conjunto de observações de variáveis possivelmente correlacionadas em um conjunto de valores de variáveis linearmente não-relacionadas, denominado de componentes principais. A transformação é definida de tal forma que a primeira componente principal tem a maior variância possível, ou seja, representa a direção de maior dispersão dos dados possível. Cada componente subsequente tem, por sua vez, a maior variância possível sob a restrição de que seja ortogonal e, dessa forma, não-relacionada com as componentes definidas anteriormente. Essa técnica tem sido aplicada em análises de dados em diferentes áreas, com propostas de evoluções como RPCA, *Robust Principal Component Analysis*, [SLCC14, SXY13].

## 2.7 Indicadores e função objetivo

Os sistemas de publicidade computacional permitem obter indicadores sobre o desempenho dos anúncios e do sistema. Para otimizar um sistema computacional podemos avaliar seus indicadores, para nos concentrarmos em apenas um único valor de otimização é possível agregar os indicadores por meio de uma função objetivo. Os principais indicadores disponíveis são definidos como se segue:

$$CTR_s = \frac{Cliques_s}{Impressoes_s} \quad (2.1)$$

A taxa de clique do sistema,  $CTR_s$ , é a quantidade de cliques em todos os anúncios do sistema,  $Cliques_s$ , dividida pela quantidade de impressões de anúncios do sistema,  $Impressoes_s$ . Esse valor varia de 0 a 1.

$$Receita_s = \sum_{cs} CPC \quad (2.2)$$

A receita do sistema,  $Receita_s$ , pode ser definida como o valor acumulado de todos os CPCs de anúncios clicados no sistema ( $cs$ ). Dessa forma, o valor gasto pelos anunciantes será igual a receita do sistema,  $Receita_s$ , que também pode ser definida como a receita do publicador no período.

$$ROI_s = \frac{Retorno_s}{Receita_s} \quad (2.3)$$

O retorno sobre o investimento do sistema,  $ROI_s$ , é a razão do retorno acumulado por todos os anunciantes em dinheiro,  $Retorno_s$ , pelo investimento gasto com anúncios exibidos pelo sistema,  $Receita_s$ .

$$eCPM_s = \frac{Receita_s}{Impressoes_s} \times 1.000 \quad (2.4)$$

O custo efetivo por milhar de impressão do sistema,  $eCPM_s$ , é a razão da receita do sistema,  $Receita_s$ , pela quantidade de impressões de anúncios do sistema,  $Impressoes_s$ , multiplicada pela constante de milhar de impressões, 1.000. Esse valor varia de 0 a 1.

A função objetivo pode então ser definida como uma combinação linear dessas utilidades:

$$Fobj_s = CTR_s + ROI_s + eCPM_s$$

Como esses valores não são de mesma ordem de grandeza,  $ROI_s$  não é limitado enquanto  $CPM_s$

e  $CTR_s$  estão entre 0 e 1, mas mesmo assim podendo ser muito diferentes, o ideal é trabalhar com valores normalizados. Como utilizaremos a função objetivo para comparar diferentes instâncias das funções de ordenação, ou seja, diferentes sistemas, podemos definir para duas instâncias  $s1$  e  $s2$  do sistema:

$$\bar{X}_{s1,s2} = \frac{X_{s2} - X_{s1}}{X_{s1}}$$

Assim, temos as definições de  $\overline{CTR}_{s1,s2}$ ,  $\overline{ROI}_{s1,s2}$  e  $\overline{eCPM}_{s1,s2}$ , que por simplicidade utilizaremos como  $\overline{CTR}$ ,  $\overline{ROI}$  e  $\overline{eCPM}$  daqui em diante, representando a diferença percentual de um sistema tendo um outro sistema como base. Assim, podemos definir a função objetivo de um sistema com relação a um sistema base previamente fixado como:

$$Fobj = \overline{CTR} + \overline{ROI} + \overline{eCPM}$$

Idealmente a utilidade de cada participante deveria ser igualmente importante, por outro lado, observamos na realidade que a importância de cada utilidade pode variar por interesses comerciais diversos. Assim, para tornar nossa representação mais realista devemos ponderar cada utilidade de acordo com o cenário comercial que será representado:

$$Fobj = \alpha \overline{CTR} + \beta \overline{ROI} + \gamma \overline{eCPM}$$

$$1 = \alpha + \beta + \gamma$$

Para fins acadêmicos podemos considerar os parâmetros  $\alpha$ ,  $\beta$  e  $\gamma$  iguais. Em situações reais esses valores devem ser ajustados para representar a importância comercial de cada um dos participantes no equilíbrio do sistema. Além disso, não queremos que as utilidades sejam mais distantes entre si do que os parâmetros determinam. Assim, vamos descontar a variância dos valores assim calculada:

$$\sigma^2 = \frac{(\overline{CTR} - \mu)^2 + (\overline{ROI} - \mu)^2 + (\overline{eCPM} - \mu)^2}{3} \quad (2.5)$$

$$\mu = \frac{\overline{CTR} + \overline{ROI} + \overline{eCPM}}{3}$$

O valor  $\mu$  é a média dos  $\overline{CTR}$ ,  $\overline{ROI}$  e  $\overline{eCPM}$ , ou seja, a sua soma dividida por três. Por se tratar de três valores, o número três também está presente no denominador como quantidade de elementos  $n$  no cálculo da variância. Mais uma vez para representar adequadamente na realidade a importância do equilíbrio para o sistema um parâmetro livre,  $\delta$ , será associado a variância, concluindo assim nossa função objetivo:

$$F_{obj} = \alpha \overline{CTR} + \beta \overline{ROI} + \gamma \overline{eCPM} - \delta \sigma^2 \quad (2.6)$$

$$1 = \alpha + \beta + \gamma$$

$$\delta \geq 0$$

## 2.8 Ordenação de anúncios

O número de anúncios que podem ser exibidos a cada requisição é limitado e a ordem na qual os anúncios são exibidos é importante, por exemplo, três anúncios serão exibidos antes dos resultados de busca, a primeira posição, mais próxima da caixa de busca costuma ser considerada a mais interessante, dessa forma o *melhor anúncio* ocupará a primeira posição e assim por diante, sendo exibidos apenas os três melhores anúncios. Voltando ao exemplo do anúncio da empresa aérea, sendo um mercado acirrado, imagine que outras cinco empresas também associaram seus anúncios à mesma palavra-chave, *Florianópolis*. Assim temos seis anúncios para exibir em três posições e ainda devemos ordenar os três selecionados de forma que o melhor dentre eles seja exibido em primeiro lugar e assim por diante.

Considerando o modelo de CPC, para que um anúncio seja exibido é necessário que todas as partes sejam definidas: o descritivo; a segmentação com os critérios de seleção e o valor máximo de CPC, aposta (*bid*); e as restrições da campanha como o orçamento total e/ou diário e datas de início e fim.

Um dos maiores desafios para um sistema de publicidade computacional é estabelecer o valor de um clique de forma que o publicador obtenha um valor justo de receita e que todos os anunciantes obtenham um bom ROI. Esse tipo de decisão adentra às áreas de economia e teoria dos jogos. Os mecanismos mais utilizados para o modelo de CPC são baseados em leilões GSP, *Generalized Second Price* [EOS<sup>+</sup>05], ou leilões de segundo preço generalizados. Ao associar uma palavra-chave a uma segmentação e a um grupo de anúncios os anunciantes estabelecem também um valor de aposta que representa o valor máximo que o anunciante deseja pagar por um clique nesse anúncio.

Quando uma requisição é recebida o sistema realiza um leilão selado (ex: apenas pelo valor da aposta) entre todos os anúncios que possuem as características da consulta (ex: a palavra-chave). Esse leilão induz uma ordenação (ex: pela aposta) e estabelece o preço por clique dos anúncios vencedores. Utilizando apenas a ordenação pela aposta, o usuário irá visualizar os anúncios em ordem decrescente de aposta. Nesse caso, o custo por clique, CPC, do primeiro anúncio será o valor da aposta do segundo anúncio, o CPC do segundo anúncio será o valor da aposta do terceiro anúncio e assim por diante. Esta foi a estratégia de ordenação adaptada pelo Goto.com/Overture no primeiro modelo de publicidade por CPC em 1997. Vários problemas foram identificados nesse modelo pois anunciantes podem abusar de seus limites para prejudicar outros anunciantes ou os usuários, por exemplo, fazendo uma espécie de publicidade *spam*, apostando um valor alto em muitas palavras, aparecendo em qualquer requisição e gastando pouco pelos critérios de precificação. Com o tempo, os usuários deixariam de clicar, reduzindo assim a receita do sistema e a satisfação do usuário.

Para superar essas limitações o Google desenvolveu um modelo que busca otimizar tanto a receita (buscando CPCs maiores) quanto a relevância dos anúncios. Essa ordenação utiliza o produto do valor da aposta por um *score* da qualidade do anúncio que corresponde ao comportamento de cliques dos usuários naquele anúncio. Esse produto é uma forma de calcular o CPM efetivo teórico (para o qual utilizaremos ECPM) adaptado da definição de Shanahan e Kurra, no capítulo 9 de *Advanced Topics in Information Retrieval*, [MBY11, SK11] :

$$ECPM(a) = CPCbid_a \times P(Clique(a)|c, a, u) \times 1.000 \quad (2.7)$$

em uma requisição para exibição no contexto (página)  $c$ , do anúncio  $a$ , para o usuário  $u$ .  $CPCbid_a$  é o valor máximo que o anunciante deseja pagar por um clique no anúncio  $a$ . O *score* ou valor de qualidade é representado por  $P(Clique(a)|c, a, u)$ , podendo ser compreendido como um substituto para a probabilidade do anúncio  $a$  exibido no contexto  $c$  ser clicado pelo usuário  $u$ , uma probabilidade desconhecida no momento da seleção e ordenação de anúncios. A constante 1.000 corresponde a 1.000 impressões, seguindo a definição de CPM. O ECPM induz uma ordenação e determina o posicionamento dos anúncios, o anúncio com maior ECPM será exibido na primeira posição disponível, o anúncio com segundo maior ECPM na segunda posição, até que todas as posições disponíveis possuam anúncios. Para determinar os preços, ou seja, o valor do CPC de cada anúncio nessa ordenação podemos assumir um leilão de segundo preço, e o CPC será o valor mínimo que um anúncio precisa pagar para manter sua posição na ordenação por ECPM. Como exemplificado em [MBY11, SK11], podemos considerar dois anúncios  $ad_1$  e  $ad_2$ , com apostas  $Bid_1$

e  $Bid_2$  e valor de qualidade  $Q_1$  e  $Q_2$  respectivamente. Assumindo, sem perda de generalidade, que o ECPM do anúncio  $ad_1$  é maior que do anúncio  $ad_2$ , teremos que:

$$Bid_1 \times Q_1 \geq Bid_2 \times Q_2$$

Para manter a ordenação por ECPM temos que:

$$Bid_1 \geq \frac{Bid_2 \times Q_2}{Q_1}$$

Dessa inequação obtemos a definição do CPC para o anúncio  $ad_1$ , adicionando um pequeno valor, usualmente um centavo:

$$CPC_1 = \frac{Bid_2 \times Q_2}{Q_1} + 0,01 \quad (2.8)$$

É importante ressaltar que o valor do CPC nunca poderá ser superior ao valor da aposta. A necessidade de utilizar o valor de qualidade como um substituto para a probabilidade de clique existe pois não conhecemos essa probabilidade antes da exibição do anúncio. O desafio de estabelecer o valor de um clique e/ou uma função de ordenação tem como raiz conhecer ou estimar essa probabilidade da melhor forma possível. O cálculo do valor de qualidade pode utilizar heurísticas, aproximações, aprendizados de máquina, modelos estatísticos e de recomendação que consideram as características dos anúncios, usuários e contextos, assim como históricos e estatísticas de cliques.

## 2.9 Reordenação de anúncios

Em situações reais, calcular a ordenação para todos os documentos é impraticável devido às restrições de volume e tempo de resposta às quais são impostos os sistemas de publicidade computacional. Uma abordagem normalmente utilizada para superar essas restrições é utilizar um processo de recuperação e ordenação com múltiplas fases, como o utilizado em [BGJM10, AG12]. Esses processos podem ser constituídos de duas ou mais fases, cada uma similar à ordenação descrita na seção 2.8, mas considerando um diferente conjunto de características (algumas dessas características podem ser dependentes da distribuição das fases anteriores). Dessa forma, os anúncios são reordenados em cada fase carregando alguma informação sobre as ordenações anteriores, esse processo de reordenação é normalmente referenciado como *re-ranking*.

Podemos considerar um exemplo para tornar mais clara a ideia de reordenação. Vamos supor

que em um sistema de publicidade computacional os anúncios são indexados e recuperados utilizando uma estrutura de lista invertida como índice, nesse processo de seleção os  $k$  anúncios mais relevantes são retornados. Nessa ordenação poderíamos utilizar o modelo vetorial e uma função de similaridade tradicional com base no cosseno para retornar os anúncios, esse modelo é descrito em detalhes em [BYRN11]. Sua adaptação para publicidade computacional é utilizar os anúncios como documentos a serem recuperados. O vetor que representa um documento é construído utilizando as palavras do documento e estatísticas como *tf* (*term frequency*), frequência, e *idf* (*inverse document frequency*), frequência inversa nos documentos, uma medida da raridade das palavras. Para representar um anúncio as palavras utilizadas podem ser dos diferentes campos do anúncio, por exemplo, do título, da descrição e das palavras-chave. Para representar um vetor de consulta, as palavras são extraídas da requisição, ou seja, do contexto e do usuário, como já discutimos. A similaridade de cossenos é calculada para todos os anúncios do índice e os que  $k$  que obtiverem os maiores valores serão retornados ordenados pela similaridade.

Técnicas de recuperação de informação permitem que essa operação seja eficiente e atenda às restrições de volume e tempo de resposta. Por outro lado, lógicas de seleção e ordenação podem necessitar de operações além das tradicionais de recuperação de informação, como a aplicação de um leilão ou um cálculo de um modelo probabilístico. Lógicas mais complexas podem não ser aplicáveis para todo o conjunto de anúncios de forma eficiente. Além disso, algumas abordagens utilizam informações calculadas no momento da seleção, como a quantidade de anúncios que têm pelo menos uma palavra da consulta ou o valor da similaridade. Retomando nosso exemplo, poderíamos estar interessados em ordenar os anúncios de acordo com a equação 2.7. Para isso, poderíamos utilizar o valor da similaridade como a qualidade e reordenar pela multiplicação da qualidade pela aposta. Como esse cálculo final só será executado para os  $k$  anúncios retornados da primeira fase, essa segunda fase pode aplicar lógicas mais complexas e utilizar informações previamente calculadas, como a similaridade e a quantidade de anúncios que poderiam ser retornados, respeitando as restrições de volume e tempo de resposta.

## 2.10 Sistemas de software adaptativos

Sistemas de software de aplicações distribuídas em ambientes com mudanças frequentes exigem intervenção humana para manter seu bom funcionamento ao longo do tempo. Essas (re)configurações, desenvolvimentos e implementações de modificações se tornam processos caros e que consomem muito tempo de especialistas. Sistemas de software adaptativos [ST09, AdLMW09] são uma res-

posta a esse problema. Esses sistemas possuem ciclos internos de realimentação com o objetivo de ajustar e adaptar de forma autônoma seu próprio funcionamento. Um sistema de software adaptativo deve monitorar a si mesmo e ao contexto do seu ambiente, detectando modificações significativas, decidindo como reagir e agir para executar tais decisões. Esses processos usualmente dependem de propriedades adaptativas, características do domínio (informações do contexto e modelos) assim como preferências dos responsáveis pelo sistema. De forma mais geral, sistemas de software adaptativos devem ser capazes de se modificar durante a sua execução em resposta a mudanças no sistema, nos requisitos, ou no ambiente.

## 2.11 Programação genética

Como indicamos utilizaremos a técnica de programação genética. Os passos básicos de uma programação genética estão descritos no *Algoritmo 1*. Uma população inicial de indivíduos é criada, fornecendo indivíduos como entrada e/ou aleatoriamente criando esses indivíduos (linha 1). Descobrimos a qualidade dos resultados fornecidos por um indivíduo executando seu programa e comparando seu resultado com outro programa por meio da função de adaptação, *fitness*, capaz de quantificar numericamente o sucesso do programa em obter bons resultados no ambiente considerado (linha 3). Os indivíduos que se saírem bem são selecionados para se reproduzir (linha 4) e gerar novos indivíduos programas para a próxima geração (linha 5). As principais operações genéticas que são utilizadas para criar novos indivíduos a partir de outros já existentes são:

- **reprodução**, ou *reprodução assexuada*, ou **clonagem** - um programa adaptado é copiado de uma geração para a próxima;
- **crossover** - a criação de um programa filho é feita combinando partes aleatoriamente escolhidas de dois programas pais;
- **mutação** - a criação de um programa filho é feita alterando aleatoriamente uma parte aleatoriamente escolhida de um programa pai.

---

**Algoritmo 1:** Programação genética
 

---

- 1 Aleatoriamente criamos uma *população inicial* de indivíduos programas;
  - 2 **repita**
  - 3     *Execute* cada programa e calcule seu *fitness* (valor de adaptação);
  - 4     *Selecione* um ou mais indivíduos programas da população com probabilidade determinada pelo *fitness* para participar de operações genéticas (reprodução, cruzamento, mutação, etc);
  - 5     Criar novos indivíduos programas aplicando as operações genéticas com as probabilidades específicas com os indivíduos programas selecionados;
  - 6 **até** que uma solução aceitável seja encontrada ou alguma condição de parada seja alcançada (como um número máximo de gerações);
  - 7 **retorna** o melhor indivíduo programa encontrado até o momento
- 

Os indivíduos programas são possíveis soluções para o problema, usualmente são representados por estruturas de dados complexas, como árvores, listas ligadas ou pilhas. O seu genótipo, ou cromossomos, é descrito nessa estrutura e é nessa estrutura que ocorrem as operações genéticas. O fenótipo, é a manifestação do genótipo no ambiente, obtido pela execução do problema para um conjunto de dados que representa o ambiente. A seleção é feita com base na adaptação do indivíduo ao ambiente, quantificada pela função de *fitness* sobre o fenótipo. Os indivíduos melhor adaptados sobrevivem e poderão se reproduzir, transferindo parte de seu genótipo para a próxima geração que tem o potencial de ser mais evoluída que a anterior. Esse processo se repete e cada geração caminha na evolução para uma melhor solução do problema.

Em [PLM08] é apresentada uma lista de sete propriedades de áreas nas quais a programação genética tem sido bem sucedida. Vamos apresentar as seis propriedades que acreditamos estar presentes no nosso problema.

**O relacionamento entre as variáveis relevantes é desconhecido ou pouco compreendido.** Na publicidade, sabemos que existem relações entre a ordenação, o valor da aposta, o grau de contextualização, o grau de afinidade do usuário, a taxa de clique, a receita, o retorno do investimento e outros. Alguns trabalhos, por exemplo, conseguiram demonstrar que anúncios com segmentos alvo melhor definidos são mais efetivos [NH97] ou que anúncios apresentados quando o usuário não tem interesse são irritantes [WZCD02]. Apesar dos indícios, quantificar e modelar esses relacionamentos continua um problema em aberto. Uma das principais vantagens de programação genética é explorar soluções em domínios pouco compreendidos, auxiliar na identificação

de quais variáveis e operações são importantes; fornecer novas soluções para problemas específicos; revelar relacionamentos inesperados entre variáveis; e até descobrir novos conceitos que podem ser generalizados para uma ampla variedade de circunstâncias.

**Encontrar o tamanho e forma da melhor solução é uma grande parte do problema.**

A programação genética permite explorar não apenas as inclusões e relações entre as variáveis, mas toda a estrutura da solução. Ao procurar uma função de ordenação para anúncios poderíamos fixar uma forma, por exemplo soma ponderada das variáveis, mas um dos principais desafios é justamente encontrar a melhor maneira de combinar essas diferentes variáveis sem limitações de forma ou estrutura.

**Uma vasta quantidade de dados de teste esta disponível computacionalmente.** Os dados formam a base para o aprendizado da programação genética. Um grande volume de dados permite aprender os nuances do problema. Dados de impressão e cliques costumam estar facilmente disponíveis nos sistemas de publicidade.

**Existem bons simuladores para avaliar o desempenho das soluções, mas não bons métodos para obter diretamente boas soluções.** A publicidade computacional desenvolveu um conjunto de indicadores que permitem avaliar o desempenho do sistema para cada um dos participantes. Por outro lado, esses indicadores só são disponíveis no ambiente real, com usuários reais. Dessa forma, podemos testar e avaliar soluções no ambiente real comparando seu desempenho com as outras soluções. A programação genética se utiliza dessas avaliações e simuladores para *minerar* uma boa solução.

**Uma solução aproximada é aceitável.** A programação genética não é capaz de garantir que a melhor solução, ou a solução exata, será obtida, funcionando melhor em domínios onde uma boa aproximação é possível e aceitável. A função de ordenação que buscamos deve melhorar o resultado do sistema e ser capaz de se adaptar a mudanças do sistema, não é nosso objetivo encontrar a função de ordenação que maximiza o resultado do sistema, até porque, não sabemos se tal função existe.

**Pequenos incrementos no desempenho são mensuráveis e muito valorizados.** Em domínios com alto interesse econômico, o estado da arte costuma ser bem avançado. Nesses domínios, pequenas melhorias podem ser extremamente valiosas. A programação genética pode, em alguns casos, descobrir pequenos, mas valiosos, relacionamentos. O negócio de publicidade computacional é um ramo bilionário, no qual pequenos incrementos nos indicadores apropriados podem significar aumentos financeiros milionários.

Em [PLM08] os autores também listam as principais áreas de aplicação de programação genética. Nosso problema pode se enquadrar em duas delas.

A primeira é **regressão simbólica**. Em muitos problemas o objetivo é encontrar uma função cuja saída possua alguma propriedade desejada e/ou produza os valores observados. A regressão simbólica vai além da ideia geral de regressão, encontrar coeficientes para funções pré-definidas de forma que a função se adapte melhor ao conjunto de dados, como funções lineares na regressão linear ou funções logarítmicas na regressão logística. A regressão simbólica tem como objetivo encontrar uma função que se adapte aos dados sem pressupor nada sobre a estrutura dessa função. Como a programação genética não faz nenhum tipo de suposição, é uma técnica bem apropriada para essa tarefa de exploração de funções. O nosso problema pode ser descrito como um problema de regressão simbólica onde buscamos uma função que ordene anúncios e tenha a propriedade de melhorar a utilidade do sistema. Regressão simbólica foi uma das primeiras aplicações de programação genética [Koz92], e continua obtendo ótimos resultados [CPVSY06, LSS+06, GBK05, Kei04] até o momento [IB13, SDV+13].

A outra área é a **modelagem econômica e financeira**. A programação genética tem sido vastamente utilizada em economia e finanças. A teoria dos jogos tem sido uma ferramenta padrão utilizada por economistas para tentar entender os mercados, mas tem sido cada vez mais suplantada por simulações computadorizadas. A programação genética tem sido utilizada como parte das simulações de sistemas sociais, por ser capaz de capturar e reproduzir parte desses comportamentos. Para exemplificar essa área podemos nos focar em um autor, apenas Chen já publicou mais de 60 artigos utilizando programação genética nessa área: previsão do índice Heng-Sheng [CWZ99], teoria dos jogos [CDY02], evoluindo regras comerciais para o S&P 500 [YC04] e modelando agentes no mercado de ações [CL05]. O nosso problema é próximo ao problema de modelar o comportamento do mercado de publicidade computacional, onde os participantes podem ser considerados jogadores com objetivos possivelmente conflitantes.

## 2.12 Conclusão

A publicidade computacional procura resolver o problema de exibir o melhor anúncio em um contexto para um usuário, um tipo de problema de integração e recuperação de informação com múltiplas, possivelmente contraditórias, funções de utilidade para os três participantes: anunciante, publicador e usuário. Esse problema tem sido abordado de diferentes formas e continua sendo estudado para os principais tipos de anúncios: visuais e texto, tanto em contextos de busca patrocinada

como de publicidade contextualizada. Os sistemas de publicidade computacional buscam resolver eficientemente esse problema submetidos a fortes restrições de escala e custo, utilizando processos de seleção, ordenação e reordenação com base nas características do anúncio como título, descrição e palavras-chave, e em modelos comerciais como leilões de segundo preço. Ao mesmo tempo, esses sistemas devem considerar o modelo de pagamento, por impressões, cliques ou aquisição, juntamente com a ordenação, para estabelecer a precificação da publicidade. A seleção, ordenação e precificação devem satisfazer, ao mesmo tempo e da melhor forma possível, os interesses do anunciante, publicador e usuário. Esses interesses podem ser representados por indicadores numéricos como CTR, ROI e eCPM, respectivamente para as utilidades do usuário, anunciante e publicador. Acreditamos que uma abordagem evolutiva aplicada a um sistema de publicidade computacional adaptativo permitirá resolver o problema de seleção e ordenação de anúncios em um ambiente real, buscando o melhor resultado para os três participantes. Nosso problema pode ser representado como um problema de regressão simbólica, no qual buscamos uma função de ordenação que melhore a utilidade do sistema combinando as dimensões de valor comercial, desempenho histórico, contexto e usuário. Devemos avaliar o impacto de cada dimensão na utilidade do sistema assim como a correlação entre as dimensões, o que será feito utilizando a técnica de PCA. Finalmente, pelos aspectos discutidos na seção 1.4 e os resultados já obtidos nas áreas de regressão simbólica e modelagem econômica e financeira, acreditamos que a técnica de programação genética será capaz de encontrar e evoluir funções de reordenação para exibição de anúncios em um ambiente de publicidade computacional.

## Capítulo 3

# Trabalhos relacionados

Para identificar o posicionamento do nosso trabalho no panorama de pesquisa mais atual analisamos os trabalhos apresentados nas últimas edições, de 2012 à 2014, de cinco importantes conferências que possuem trilhas voltadas para publicidade computacional, em ordem alfabética: **CIKM** - ACM Conference on Information and Knowledge Management; **KDD** - ACM international conference on Knowledge Discovery and Data mining; **SIGIR** - international ACM conference on Research and development in Information Retrieval; **WSDM** - ACM international conference on Web Search and Data Mining; e **WWW** - international conference on World Wide Web. Selecionamos todos os artigos que foram apresentados em trilhas específicas da área de publicidade computacional dessas conferências, obtendo um total de 38 artigos.

Os 38 artigos podem ser distribuídos da seguinte forma: treze tiveram como foco a publicidade de anúncios visuais (*display ads*); seis artigos abordavam publicidade de busca patrocinada; quatro artigos podem ser aplicados a qualquer forma de publicidade computacional; quatro artigos aplicaram aprendizado supervisionado a sub-problemas de publicidade computacional como, por exemplo, previsão de cliques [MHS<sup>+</sup>13]; dois artigos discutiram alternativas e melhorias nos modelos de leilões [AGWY14, NNY14]; dois artigos abordaram a automação da definição do valor das apostas para leilões em tempo real, RTB ou *real-time bidding* para publicidade de anúncios visuais [ZYW14, YWC<sup>+</sup>14]. Os demais artigos abordaram outros temas de interesse nessa área como, por exemplo, problemas relacionados a publicidade em dispositivos móveis [OLL<sup>+</sup>14].

Por esse panorama observamos que o principal interesse em contextualização de anúncios está voltado para anúncios visuais, refletindo também o que ocorre no mercado publicitário, onde os anúncios visuais estão em destaque. O interesse publicitário e comercial pode ser explicado pelos resultados comerciais alcançados: anúncios visuais são mais atraentes aos usuários e possuem va-

lores comerciais mais altos em comparação com anúncios texto. Ao mesmo tempo, contextualizar anúncios visuais é um problema relevante, pois identificar características de contexto em imagens apresenta um desafio adicional. O problema de texto já foi vastamente estudado pelas áreas de processamento de linguagem natural e recuperação de informação. Algumas vezes o problema de contextualizar anúncios visuais é simplificado para um problema de contextualizar texto, quando se utiliza palavras-chaves associadas aos anúncios como metadados que não são exibidos aos usuários. Tratar o problema diretamente exige novas abordagens como a proposta de utilizar características multimídias de [CZA<sup>+</sup>12, AZZ<sup>+</sup>12].

Em outra linha de estudo, os modelos de leilões e, sobretudo, leilões em tempo real têm sido muito discutidos recentemente para anúncios visuais. Esses estudos têm aproximado anúncios visuais dos demais modelos. Usualmente os anúncios visuais utilizavam o modelo de CPM, agora passaram a utilizar também leilões como já ocorria em busca patrocinada e publicidade contextualizada. Por exemplo, o recente artigo [ZYW14] busca encontrar o valor ótimo de aposta para impressões, com base em informações coletadas no momento da execução associadas a dados históricos. Esse trabalho afirma que existe uma relação não-linear entre o valor ótimo de aposta e indicadores de desempenho como a taxa de clique, CTR, ou o retorno sobre o investimento, ROI. Uma forma de análise e implementação muito semelhante as utilizadas em busca patrocinada e publicidade contextualizada, e utilizada por nós neste trabalho.

A maioria dos artigos avaliados consideraram a contextualização na dimensão do usuário ou de seu comportamento, normalmente referenciada como *targeted advertising* ou *behavioral targeting advertising*, buscando apresentar ao usuário anúncios que conseguiram atingir outros usuários que têm comportamento semelhante ao do usuário para o qual irá ser exibido o anúncio. A maioria das abordagens utiliza aprendizado supervisionado com base em dados coletados de outras interações (*feedback*) aproximando-se mais de sistemas de recomendação [TLY<sup>+</sup>11, LPAJ12, APJ12]. Outras utilizam a mesma fonte para estimar a probabilidade de clique ou conversão, algumas vezes com o objetivo de estimar ou otimizar a aposta para sistemas de *real-time bidding*, leilão em tempo real, [CZA<sup>+</sup>12, LODL12, PDH<sup>+</sup>12, FB12]. Um trabalho aborda o problema de selecionar os usuários de comportamento semelhante como um problema de recuperação de informação sobre um índice de usuários conhecidos, uma pequena amostra de usuários que clicaram na publicidade é utilizada para construir uma consulta que retorna os perfis de usuários com maior ranqueamento, a principal vantagem apresentada é a eficiência em um conjunto muito grande de dados de usuário [TPGJ11].

O artigo [XZG12] apresenta um modelo Bayseano hierárquico de regressão que utiliza inferência estocástica para prever o CTR e recomendar campanhas de publicidade computacional, esse

trabalho apresenta o método geral e utiliza a publicidade computacional como um exemplo de aplicação. Apesar de apresentar diversas semelhanças de objetivos, não aborda as diferentes dimensões do problema, mas parece ser possível adaptar o nosso problema para esse método. Como discutimos anteriormente nosso trabalho tem características de regressão simbólica, tanto a técnica de programação genética como modelos Bayesianos têm sido aplicados nesse contexto e obtidos bons resultados. O artigo [YW12] apresenta o problema de *exploration and exploitation* na seleção e ordenação de anúncios de texto utilizando POMDP (*partially observable markov decision processes*), uma abordagem estocástica que permite usar a correlação entre anúncios para aumentar o lucro do publicador ao longo do tempo. O problema de *exploration and exploitation*, como descrito em [LWZ<sup>+</sup>10], é inerente à natureza de fluxo dos dados *online*, que inevitavelmente faz com que um sistema de publicidade tenha que optar entre maximizar a receita de acordo com seu conhecimento atual (*exploitation*) ou tentar aprender mais sobre o desconhecido para aumentar seu conhecimento (*exploration*), uma vez que esse último pode aumentar sua receita no futuro. O artigo [AG12] apresenta uma abordagem em dois estágios para selecionar e ordenar a publicidade, como já discutimos na seção 5.1.9, e discutiremos novamente mais adiante nesta seção. Após essa avaliação inicial de trabalhos recentes, aprofundamos nossa pesquisa nos conceitos principais do problema.

O sucesso da publicidade computacional nos diferentes tipos de anúncios, têm motivado a pesquisa em muitos tópicos a ela relacionados. O tipo de anúncio no qual se concentra nosso trabalho, anúncios textuais de publicidade contextualizada, não é tão tradicional ou pesquisado quanto anúncios de busca patrocinada ou anúncios visuais como listamos anteriormente, mas também têm atraído muito interesse da comunidade científica pelos desafios adicionais que apresenta, como discutiremos agora. Muito do que é desenvolvido em recuperação de informação para mecanismos de busca tem sido diretamente aplicado em busca patrocinada, porém, a aplicação dessas inovações em publicidade contextualizada não é tão direta e nem sempre produz o mesmo impacto nos resultados. O mesmo pode ser dito sobre sistemas de recomendação, as técnicas para recomendação de itens também têm sido utilizadas para recomendar conteúdo ou publicidade contextualizada. Apesar dos resultados interessantes em escopos reduzidos, como recomendação de anúncios de produtos e *retargeting*, no caso geral de identificar o melhor anúncio para ser exibido ainda não conseguiu reproduzir o mesmo sucesso.

Programação genética tem sido aplicada em diferentes tópicos de recuperação de informação obtendo bons resultados, por exemplo, em problemas de classificação de documentos. Nosso trabalho foi inspirado nos muitos trabalhos que aplicaram programação genética para descobrir funções de ordenação (*ranking*) de documentos, como os trabalhos de Fan [FPZ09, FPW06, FGPP05, FFPW04].

Apesar de nos inspirar, nos distinguimos em vários aspectos importantes. Ao tentar encontrar funções de ordenação para publicidade contextualizada nos defrontamos com diferentes características e dimensões que não existem no modelo clássico de recuperação de informação, desde características estruturais, da hierarquia de campanhas e grupos de anúncio, às dimensões de valor comercial, desempenho histórico e usuário, observando que o problema clássico é usualmente restrito à dimensão de contexto.

Por outro lado, a origem do nosso problema é descobrir ou estimar qual a probabilidade de um anúncio ser clicado pelo usuário, se exibido em um determinado contexto, ou mais profundamente o que leva um usuário a clicar em um anúncio [GYX+10]. A abordagem mais direta seria estimar ou modelar diretamente a probabilidade de clique [GCBH10, ZCM+10], mas como observamos anteriormente, não precisamos de uma solução exata como a probabilidade de clique, mas apenas de uma ordenação que pode ser fornecida por uma solução aproximada. Outro aspecto é que apesar de inicialmente buscarmos apenas a parte da função de ordenação que representa um estimador para a probabilidade de clique, como discutiremos em 6.8.3, também queremos explorar a função de ordenação com toda a estrutura e dimensões livres, incluindo a dimensão de valor comercial, o que vai além da estimativa ou modelagem da probabilidade de clique.

Outras formas de aprendizado de máquina, além de programação genética, têm sido aplicadas para encontrar funções de ordenação. A utilização de regressões [XZG12] continua sendo explorada mas também evoluiu para utilizar alternativas como algoritmos estocásticos [KLZM11] a fim de explorar diferentes formas de relacionamento. A nossa opção pela programação genética é fortemente inspirada pelo que descrevemos na justificativa da hipótese 1.4, mas muito provavelmente, outras técnicas de aprendizado podem e poderão ser utilizadas nesse problema.

O trabalho mais próximo que encontramos na pesquisa é *Learning to Advertise* [LCG+06], do ano de 2006. Nesse trabalho, os autores descrevem a base de utilização de programação genética para combinar características de anúncios de publicidade contextualizada para obter uma função de ordenação. Como apresentaremos na seção 6.8.2, esse trabalho se concentra na dimensão de relevância entre o anúncio e o contexto. As características utilizadas são agrupadas por nós nessa dimensão. Dessa forma, o resultado, ou função de ordenação por contexto, de *Learning to Advertise*, poderia ser utilizado para obter o valor da nossa dimensão contexto na primeira fase da nossa abordagem. A diferença entre os trabalhos está justamente no fato de utilizarmos uma segunda fase para combinar as outras dimensões que não foram contempladas. Na conclusão do artigo *Learning to Advertise*, os autores indicam como um dos planos futuros expandir o modelo para encontrar funções de ordenação que combinem a relevância com aspectos monetários do problema.

Sob essa perspectiva, podemos considerar que nosso trabalho evolui essa solução. Um trabalho mais recente( [DKR11]) revisitou de forma simples esse artigo sem fazer menção a esse tipo de evolução. De acordo com nossa avaliação esse trabalho mais recente não adicionou nenhum valor ao trabalho original, podendo ser desconsiderado de qualquer outra discussão.

O artigo [THT<sup>+</sup>14] abrange exatamente a mesma área de concentração do nosso trabalho, abordando publicidade contextualizada com uma visão de recuperação de informação e *learning-to-rank*. O trabalho elabora modelos com base em realimentação, especificamente *click-feedback*. Por outro lado, esse trabalho tem como objetivo resolver um problema diferente do que o proposto por nós. O trabalho de Tagami *et al.* propõe um método de tradução para resolver o problema de vocabulários diferentes entre os anúncios e o contexto, inclusive a página. Outras soluções para esse problema exigem a criação de categorias ou classes e a construção de índices e processamento específico para esse tratamento. O artigo apresenta um método considerado pelos autores muito mais fácil de se implementar e que não cria a necessidade de modificar o atual sistema de recuperação de anúncios. Além disso, os autores indicam que o método é capaz de recuperar anúncios mais adequados mantendo o desempenho do sistema de recuperação de anúncios. Em resumo, o método é capaz de aprender o mapeamento entre as informações textuais do contexto e as características textuais dos anúncios, utilizando os dados coletados de interações de cliques. Na nossa abordagem, utilizamos esses dados para avaliar o desempenho das soluções candidatas e permitir a sua seleção e evolução.

O artigo [AG12] apresenta uma abordagem em duas fases para selecionar e ordenar a publicidade: primeiramente recupera os *top-k* anúncios como um problema tradicional de recuperação de informação no qual cada anúncio é representado por um vetor esparsos de características e um *score* de relevância entre o anúncio e a consulta é calculado pelo produto vetorial. O segundo nível avalia um modelo de aprendizado de máquina na força bruta apenas para os *k* anúncios recuperados. Essa abordagem permite utilizar a eficiência de recuperação de informação para grandes volumes de anúncios e requisições, necessária para um sistema real de publicidade computacional, sem perder a qualidade dos resultados produzidos pelo modelo de aprendizado de máquina. Esse trabalho inspirou nossa abordagem em duas fases, que também utilizamos para unir as informações das diferentes dimensões.

O que podemos concluir desse levantamento é o interesse da comunidade em pesquisas envolvendo anúncios visuais, comportamento de usuário e aprendizado de máquina. Dessa forma, nosso trabalho pode ser visto como: 1 - um trabalho próximo ao trabalho [AG12], pois queremos combinar aprendizado de máquina e recuperação de informação para obter eficientemente anúncios ordenados;

2 - um trabalho em uma área que continua interessante do ponto de vista comercial e de pesquisa, como ilustrou o recente trabalho [THT<sup>+</sup>14] que também envolve publicidade contextualizada no formato de anúncios de texto; 3 - um trabalho que tem como meta otimizar diferentes objetivos dos participantes do sistema, da mesma forma que o trabalho de busca patrocinada [OLL<sup>+</sup>14]; e, por fim, 4 - uma evolução do trabalho [LCG<sup>+</sup>06]. Por outro lado, também nos aproximamos de outros trabalhos avaliados por considerar a dimensão de comportamento do usuário como fator relevante da estimativa de probabilidade de clique. O fato de utilizarmos no trabalho e nos experimentos anúncios de texto em publicidade contextualizada nos permite utilizar formas tradicionais de relacionar o contexto da página ao anúncio, mas abordagens semelhantes a [CZA<sup>+</sup>12] poderiam ser utilizadas para se obter o valor de contexto de anúncios visuais permitindo estender nossa abordagem diretamente para esse tipo de publicidade.

A principal contribuição do nosso trabalho está na combinação das diferentes dimensões do problema. Trabalhos anteriores já trataram de combinar diferentes dimensões, como relevância e *click-feedback* [CAJ08]. Outros combinaram modelos de regressão e ordenação [Scu10]. Nosso trabalho não apenas combina as diferentes dimensões do problema e utiliza realimentação com base nos indicadores originados da interação do usuário (impressões, cliques e receita), como também combina o modelo de ordenação com aprendizado orientado a esses indicadores.

## Capítulo 4

# Proposta: Abordagem evolutiva em publicidade computacional

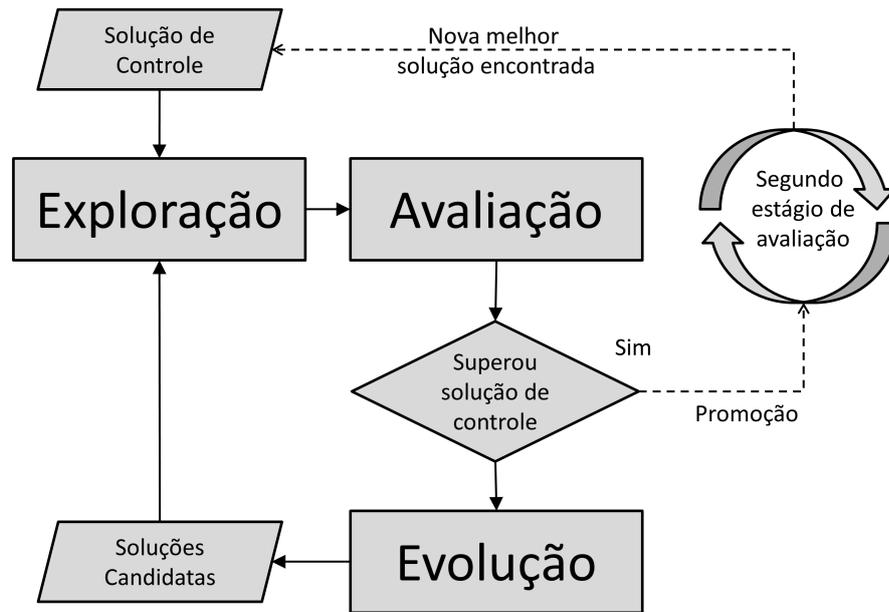
A abordagem evolutiva que desenvolvemos para resolver o nosso problema de publicidade computacional foi denominada em nossos artigos como *Evolutionary Ranking for Online Advertising*<sup>1</sup>. Nos inspiramos fortemente no ciclo de evolução da programação genética, descrito na seção 2.11. Nossa abordagem também é muito semelhante a diversas outras abordagens, como os experimentos de estratégias usualmente aplicados em problemas do tipo *multi-armed bandit* [HM14], técnicas de *exploration and exploitation* [LWZ<sup>+</sup>10] e experimentos *online* controlados (testes A/B) [DLG14a].

Em contrapartida, nosso objetivo foi desenvolver uma nova abordagem que não apenas fosse capaz de explorar e avaliar soluções candidatas, mas que também permitisse a evolução e adaptação de soluções ao longo do tempo. Como já discutimos, por melhor que seja uma solução, função ou fórmula de ordenação, dificilmente essa solução continuará sendo a melhor solução para o sistema durante um longo período de tempo. Para isso, a abordagem estabelece um modelo cíclico, ilustrado na Figura 4.1. O modelo é composto por um conjunto não-prescritivo de passos que permitem a *Exploração* de experimentos, *Avaliação* dos resultados e *Evolução*, ou mesmo, a criação de novas soluções.

O primeiro passo do modelo, *Exploração*, inicia com um conjunto de soluções candidatas para o problema, que no nosso caso é composto por funções de ordenação de anúncios. Esse passo permite que soluções candidatas sejam executadas para um conjunto aleatoriamente selecionado de instâncias do problema. No nosso cenário, cada solução é responsável por ordenar anúncios para um conjunto de requisições durante a experimentação.

---

<sup>1</sup>Ordenação Evolutiva para Publicidade Online



**Figura 4.1:** *Abordagem evolutiva - Ciclo de Exploração, Avaliação e Evolução*

No segundo passo do modelo, *Avaliação*, cada solução candidata pode ser avaliada comparando seus indicadores, ROI, CTR e eCPM, com a **solução de controle**. Essa solução de controle é usualmente a melhor solução conhecida para o problema até o momento. A comparação pode ser feita por meio do cálculo da função objetivo descrita na seção 2.7 pela equação 6.1. Solução de controle é um conceito utilizado em experimentos controlados, como em testes A/B [DLG14a]. Uma solução de controle é utilizada como a referência para a execução de um experimento. Dessa forma, a avaliação do resultado do experimento utiliza como base a solução de controle. Na nossa abordagem, a solução de controle é uma fórmula de ordenação de anúncios, usualmente, a melhor fórmula conhecida até o momento.

Esse fluxo de experimentação e avaliação pode ser executado em ambientes *online* para tornar possível tomada de decisões orientadas a dados, por meio da comparação de diferentes alternativas no mesmo período de tempo. Experimentos descritos no capítulo 7 foram conduzidos exatamente dessa forma. Conforme já apresentamos, isso também é necessário devido a dificuldade de simular a interação dos usuários. Por meio da nossa abordagem, as soluções candidatas são experimentadas por usuários reais, com anúncios reais, durante interações com páginas reais em um ambiente real de publicidade computacional.

Com a inclusão do terceiro passo, *Evolução*, nós aprimoramos esse fluxo incorporando mecanismos para criar automaticamente novas soluções e experimentos. Isso possibilitou ao sistema de

publicidade computacional responder de forma mais eficaz às mudanças no ambiente. Novas soluções são criadas utilizando como base as melhores soluções dos experimentos anteriores. Esse é o passo mais importante da abordagem evolutiva proposta. Um mecanismo de evolução é necessário para executar esse passo, acreditamos que técnicas de aprendizado são adequadas para a implementação desse motor evolutivo.

Por fim, incluímos uma forma de promover a melhor solução encontrada. Quando uma solução supera a melhor solução conhecida até o momento, ou seja, a solução de controle, por um determinado montante (*threshold*) ela é promovida a um segundo estágio de validação. No segundo estágio, um experimento semelhante é executado, comparando apenas a solução promovida com a solução de controle por um período de tempo mais longo que o do primeiro estágio. Caso a solução promovida supere novamente o desempenho da solução de controle durante o segundo estágio de experimentação, ela pode se tornar a nova melhor solução para o problema. O que significa, no nosso ambiente de publicidade computacional, que desse momento em diante a maioria dos anúncios deveriam ser ordenados por essa solução, assim como essa solução deve ser utilizada como a nova solução de controle para a avaliação de novos experimentos e evoluções.

A abordagem evolutiva que descrevemos pode, a princípio, parecer simples, com um ciclo principal e poucos passos. A estruturação da abordagem nesse formato foi feita justamente com o objetivo de tornar mais claro um processo que pode ser muito complexo para se compreender em sistemas como os de publicidade computacional. A implementação de cada um dos passos pode exigir grandes esforços de engenharia de software ao mesmo tempo que apresenta desafios científicos como a escolha e configuração de técnicas de aprendizado ou a modelagem do problema e dos dados.

Na Figura 4.2 resumimos as etapas da criação da abordagem evolutiva. Para construir a ordenação evolutiva, o etapa superior na Figura 4.2, inicialmente precisamos adicionar ao sistema de publicidade computacional a capacidade de adaptação. Essa capacidade de adaptação, etapa intermediária na Figura 4.2 envolve duas habilidades: identificar situações que exigem mudanças e executar mudanças. A primeira é usualmente implementada em sistemas de software como métricas que são calculadas e comparadas com valores de referência. Em sistemas de publicidade podemos utilizar os indicadores descritos na seção 2.7 para isso. Por outro lado, para modificar sistemas de software usualmente é necessário um ciclo de desenvolvimento, teste e implantação de mudanças. Tais ciclos podem envolver muitas pessoas e longos períodos de tempo. Para tornar a modificação do comportamento do sistema de software algo passível de ser parte de um ciclo autônomo, como estabelece a nossa abordagem evolutiva, utilizamos a técnica de reconfiguração para prover ao sistema a capacidade de adaptação. A capacidade de reconfiguração, etapa de base na Figura



**Figura 4.2: Abordagem evolutiva** - A construção da ordenação evolutiva tem como alicerce um sistema de publicidade adaptativo. Para desenvolver a capacidade de adaptação criamos como base uma arquitetura de reconfiguração para o sistema de publicidade computacional.

4.2, foi desenvolvida utilizando um repositório remoto e centralizado de configurações, o ReConf [BMF14]. Nos próximos capítulos nós descrevemos o sistema de publicidade computacional, apresentamos o ambiente real de publicidade computacional utilizado e descreveremos a implementação da abordagem.

## Capítulo 5

# Engenharia e arquitetura

Nesse capítulo descrevemos a construção da abordagem evolutiva proposta por meio de um conjunto de soluções de software. Inicialmente, descrevemos o sistema de publicidade computacional que utilizamos no projeto e o ambiente de veiculação de publicidade no qual ele está inserido. Na segunda parte do capítulo, descrevemos como implementamos a capacidade adaptativa nesse sistema por meio da criação de uma solução de reconfiguração.

### 5.1 Descrição do sistema de publicidade computacional

O sistema de publicidade computacional que trabalhamos é executado de acordo com os seguintes passos:

1. *Crawling* das páginas onde serão exibidos anúncios para identificar palavras-chave relevantes.
2. Classificação por especialistas que atribuem categorias de uma taxonomia comercial para essas páginas.
3. Identificação de interesses dos usuários associando palavras-chave e/ou categorias com base no histórico de navegação e classificação/agrupamento de usuários com interesses semelhantes.
4. Criação e gerenciamento de anúncios pelos anunciantes.
5. Indexação dos anúncios no modelo vetorial, *tf-idf*, e *bag-of-words*, como descrito em [BYRN11] e com identificação do campo de origem dos termos como explicamos em 2.5.
6. Composição da requisição com as informações de palavras-chave e categorias da página, interesses do usuário, dados de geo-localização da requisição e dados de usuário identificado.

7. Transformação da requisição de anúncios para um determinado contexto e usuário em vetores de consulta, da mesma forma que na indexação dos anúncios.
8. Seleção dos anúncios relevantes utilizando recuperação de informação e ordenação por similaridade de cossenos dos vetores do anúncio e da consulta.
9. Reordenação dos anúncios de acordo com a função de ordenação que combina as diferentes dimensões de valor comercial, desempenho histórico, contexto e usuário.
10. Cálculo do CPC de acordo com leilão de segundo preço e a fórmula 2.8.
11. Construção da resposta e retorno para o navegador do usuário.

Além disso, o sistema também registra dados sobre cliques e impressões para permitir análises e cálculos de indicadores.

### 5.1.1 *Crawling*

O sistema tem como entrada uma lista de endereços de páginas que precisam ser processadas. Ele obtém o conteúdo da página diretamente da Internet e utiliza técnicas de análise e processamento de texto e linguagem, como lematização e dicionários, para identificar as palavras-chaves mais relevantes. O sistema armazena internamente a lista de palavras-chave associada a cada endereço para ser utilizada futuramente na seleção de anúncios.

### 5.1.2 *Classificação*

Cada domínio, subdomínio ou parte de um site que possa ser especificada por um endereço pode ser associada a um conjunto de categorias de uma taxonomia comercial que também é utilizada como segmentação dos anúncios. A associação dessas categorias é feita pelos donos dos domínios em interfaces do sistema específicas para esse fim. O sistema armazena internamente a lista de categorias associada a cada endereço para ser utilizada futuramente na seleção de anúncios.

### 5.1.3 *Interesses do usuário*

O sistema armazena de forma anonimizada o histórico de endereços que exibem anúncios de publicidade que foram visitados pelos usuários. Por meio de agrupamentos e heurísticas, o sistema identifica quais categorias da taxonomia comercial são de interesse para o usuário associando o usuário a um grupo de usuários que possuem interesses semelhantes <sup>1</sup>. Esses interesses, por sua vez,

---

<sup>1</sup>A associação do usuário aos grupos é armazenada por meio de dados gravados no navegador do usuário em arquivos de texto denominados *cookies*.

são representados por uma lista de categorias da taxonomia comercial. Dessa forma, nas próximas visitas o sistema terá acesso a uma lista de categorias que o usuário pode ter interesse, podendo utilizar essa lista de categorias na seleção de anúncios.

#### 5.1.4 Gerenciamento de anúncios

As contas, campanhas e anúncios são criados, editados e excluídos por meio de uma interface *web* própria. Nessa interface, os anunciantes definem todos os atributos dos anúncios, seus descritivos e suas segmentações. Por meio dessa interface também é possível gerenciar as apostas e orçamentos, assim como obter relatórios detalhados da entrega e dos resultados de impressões e cliques dos anúncios.

Os seguintes campos são fornecidos pelo anunciante por meio da interface de gerencialmente e são utilizados na indexação, seleção e ordenação dos anúncios:

- título;
- descrição;
- palavras-chave;
- categorias;
- restrições de usuário;
- aposta.

#### 5.1.5 Indexação de anúncios

A indexação de anúncios utiliza o produto cartesiano de todos os descritivos (título e descrição) e segmentações (palavras-chave, categorias, restrições de usuário e aposta) de um grupo de anúncio. Cada par  $\langle \text{descritivo}, \text{segmentação} \rangle$ , ou simplesmente anúncio, é indexado no modelo vetorial ponderado por *tf-idf* utilizando o conceito de *bag-of-words* dentro de cada campo de origem. Dessa forma, existem diferentes *bags*, uma para as palavras-chave, outra para categorias, outra para as palavras do título, outra para as palavras da descrição e assim por diante. Isso permite que o modelo vetorial seja configurado com diferentes pesos para diferentes partes do anúncio, assim como permite tornar um casamento, ou *match*, em um campo obrigatório ou não para filtrar a seleção do anúncio.

Além dos dados disponibilizados pelo gerenciamento de anúncios são utilizados indicadores calculados, como o CTR do anúncio. O CTR do anúncio é calculado a partir da quantidade de cliques e impressões registradas para o descritivo e para a segmentação.

### 5.1.6 Requisição de anúncios

Quando uma página é exibida e contém o código para requisitar anúncios ao sistema de publicidade computacional, uma requisição de anúncios chega ao sistema. O sistema obtém todas as informações disponíveis para essa requisição. Com base no endereço da página requisitante, o sistema obtém a lista de palavras-chave previamente computada, assim como a lista de categorias associadas. Com base nos *cookies* do usuário o sistema obtém a lista de categorias de interesse do usuário e dados adicionais do usuário quando ele estiver autenticado em um sistema integrado ao sistema de anúncios (dados como o sexo e a idade do usuário, por exemplo). O sistema obtém também dados de geo-localização baseados em IP ou dispositivos móveis. Todos esses dados são então reunidos e organizados para construir uma consulta de  $k$  anúncios.

### 5.1.7 Consulta de anúncios

O sistema transforma os dados obtidos para construir uma consulta de anúncios no mesmo modelo vetorial utilizado para a indexação. Essa consulta é composta de um vetor com as características da página e um vetor com as características do usuário. A consulta é então enviada ao mecanismo de recuperação de informação associado ao índice de anúncios.

### 5.1.8 Seleção de anúncios

O sistema executa a consulta de anúncios no índice de anúncios, com o objetivo de recuperar uma quantidade de anúncios maior do que os  $k$  anúncios requisitados para exibição. A seleção de anúncios utiliza técnicas de recuperação de informação e uma primeira fase de ordenação dos anúncios pela similaridade de cossenos dos vetores dos anúncios e da consulta. Nessa fase é calculado o valor das dimensões de contexto e usuário. O valor da dimensão de contexto é o valor da similaridade de cossenos calculado entre o vetor da página e o vetor do anúncio. De forma análoga, o valor da dimensão de usuário é o valor da similaridade de cossenos calculado entre o vetor dos dados do usuário e o vetor do anúncio. Esses valores de similaridade passam a ser disponíveis para as demais fases.

### 5.1.9 Reordenação de anúncios

O principal ponto de interesse deste trabalho ocorre na segunda fase de ordenação, ou seja, na reordenação de anúncios, quando o sistema utiliza uma função de reordenação para combinar as diferentes dimensões de valor comercial, desempenho histórico, contexto e usuário. Os anúncios são então reordenados com base no valor calculado pela função de reordenação com base nos valores das dimensões para todos os anúncios retornados pela primeira fase. Os  $k$  anúncios melhor ordenados serão retornados para exibição.

Para permitir a reordenação é necessário que todos os indicadores de cada dimensão estejam disponíveis nessa fase:

- **valor comercial** - Aposta, disponível pelo gerenciamento e indexado junto com os demais dados do anúncio;
- **desempenho histórico** - CTR, computado durante a indexação;
- **contexto** - Similaridade de cossenos dos vetores do anúncio e do contexto (*tf-idf*), calculado na primeira fase de seleção e ordenação;
- **usuário** - Similaridade de cossenos dos vetores do anúncio e do usuário (*tf-idf*), calculados na primeira fase de seleção e ordenação.

Cada um desses itens do sistema ou indicadores poderiam ser diferentes e são oportunidades de melhoria. Poderíamos, por exemplo, utilizar um modelo probabilístico para mensurar a relação entre um usuário e um anúncio. Utilizar mais características para modelar os anúncios e requisições, como palavras-chave da página de destino, características estruturais da campanha ou comerciais como o orçamento. Diversos trabalhos podem ser utilizados para aprimorar esse sistema descrito, o ponto importante é que se for possível mensurar um valor numérico que represente a dimensão, esses trabalhos poderão ser agrupados na nossa abordagem obtendo resultados cada vez melhores. Utilizamos cada dimensão representada pelos atributos mais simples e modelos mais tradicionais para facilitar a compreensão e avaliação da nossa abordagem.

### 5.1.10 Cálculo de custo por clique

O sistema utiliza a abordagem de leilão de segundo preço para calcular o valor do custo por clique com base na fórmula 2.8 e descrito na seção 2.8:

$$CPC_1 = \frac{Bid_2 \times Q_2}{Q_1} + 0,01$$

O valor da qualidade é calculado com base na função de reordenação isolando-se o valor da aposta.

### 5.1.11 Resposta

O sistema utiliza os dados do anúncio para construir o retorno e a exibição de anúncios, retornando ao navegador do usuário que fez a requisição de anúncios para que os anúncios sejam exibidos na página responsável pela requisição.

## 5.2 Ambiente

Este trabalho tem à disposição um ambiente real de publicidade computacional, o ambiente de publicidade computacional do portal brasileiro UOL, Universo Online - *www.uol.com.br*, e o sistema utilizado para veicular anúncios de texto nas páginas do portal. O UOL é um dos maiores portais da Internet brasileira, com 16 anos de existência, que de acordo com a página de audiência do próprio portal <sup>2</sup>, alcançou em 2011, segundo o Ibope, uma média de 24,183 milhões de visitantes únicos domiciliares mensais e 31,511 milhões de visitantes únicos em ambientes corporativos. O volume de páginas vistas mensalmente é em média de 4,769 bilhões. Em comparação com outros portais brasileiros o UOL se posiciona como líder em visitantes únicos, páginas vistas e tempo total *online*. Na média mensal de 2011 o tempo de permanência no portal foi de 3,111 milhões de minutos. Dados de 2014, segundo a Omniture, disponibilizados pelo portal na página <sup>3</sup> indicam que o portal possui mais de 7,4 bilhões de páginas vistas todos os meses e sua *home page* recebe mais de 50 milhões de visitantes únicos por mês. Além disso, o portal informa que possui mais de 1.000 canais de jornalismo e atrai sete em cada dez internautas brasileiros.

O UOL é um portal com páginas e notícias sobre os mais variados temas, integrante do Grupo Folha, que inclui o maior jornal brasileiro, A Folha de S. Paulo. Além de conteúdo, o portal possui produtos próprios como e-mail, bate-papo e comparador de preços. O sistema de publicidade fornece anúncios para todos esses diferentes contextos, o que permitirá o estudo de contextos bem variados. Nem todas as páginas exibem anúncios, diversas páginas fazem mais de uma requisição de anúncios,

---

<sup>2</sup><http://sobreuol.noticias.uol.com.br/historia/audiencia.jhtm>, visitada em fevereiro de 2015

<sup>3</sup><http://sobreuol.noticias.uol.com.br>, visitada em fevereiro de 2015

cada requisição pode exibir uma diferente quantidade de anúncios, variando de um a mais de uma dúzia, sendo a média entre dois e três anúncios. Utilizando esses dados e a audiência do portal podemos estimar a quantidade de impressões de anúncios como duas vezes a quantidade de páginas vistas, 7,4 bilhões, ou seja, 14,8 bilhões de impressões de anúncios por mês. Considerando 30 dias no mês, teremos um volume diário de 493 milhões de impressões de anúncios. Como não podemos divulgar a taxa de clique real do sistema, podemos considerar um piso arbitrário de taxa de clique de 0,1%, baixa se comparada a busca patrocinada. Por outro lado, esse valor é aceito pelo mercado de publicidade online como uma taxa de clique média geral para anúncios visuais. Utilizando essa taxa de clique, temos a estimativa de 493 mil cliques por dia.

Os anúncios de publicidade contextualizada são comercializados pelo portal no modelo de CPC, por meio do site UOL Cliques<sup>4</sup>, o valor mínimo sugerido para aposta é de 15 centavos de real. Caso esse seja o valor médio do CPC do sistema, temos como piso da receita estimada 0,15 vezes 493 mil cliques por dia, 73.500 reais por dia, ou 2,218 milhões de reais por mês.

Neste trabalho, teremos acesso a uma pequena parcela do total de requisições e cliques do sistema. Nessa parcela, estabeleceremos o grupo de controle com a função de ordenação de referência e realizaremos os testes e evoluções no restante da amostra. Todo o restante do sistema pode ou não utilizar alguma função de ordenação exibida neste trabalho. Coletaremos e compararemos os dados de cliques, impressões e receita das funções e exporemos os incrementos ou decrementos percentuais dos indicadores calculados, mas não os valores absolutos observados.

Acreditamos que esses valores comparativos percentuais são suficientes para avaliarmos os resultados. A não divulgação dos dados absolutos se faz necessária pelo sigilo comercial e estratégico acordado com o portal, que disponibilizou o ambiente e compartilhou o risco dos testes. Esse ambiente oferece uma oportunidade única de estudar o problema na sua aplicação real, com todas as restrições de escala e tempo de resposta dos sistemas de publicidade computacional. Os resultados obtidos, positivos ou não, serão reais e a abordagem será avaliada sem a proteção de modelos abstratos, dados estáticos ou ambientes controlados.

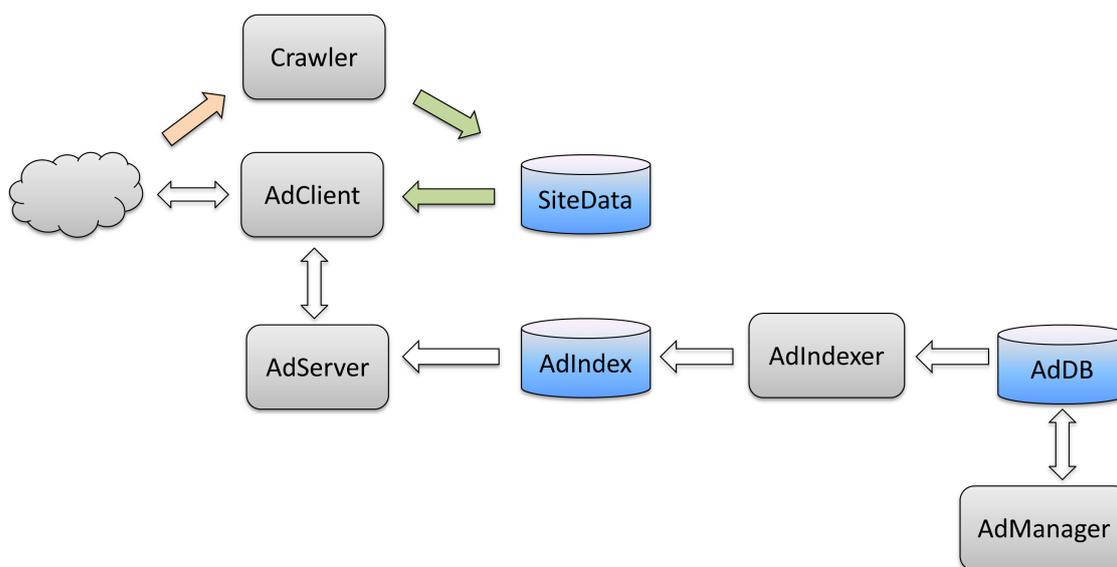
### 5.3 Arquitetura

Para executar todas as funções necessárias e respeitar as restrições de volume e tempo de resposta, um sistema de publicidade computacional usualmente precisa ser construído de forma distribuída. O sistema de publicidade computacional utilizado neste trabalho, denominado SIGA

---

<sup>4</sup>[cliques.uol.com.br](http://cliques.uol.com.br)

(Sistema de Gerenciamento de Anúncios), foi construído na linguagem Java pelas equipes de desenvolvimento do portal UOL, utilizando uma arquitetura distribuída, orientada a serviços, no modelo REST Web Services (Representational State Transfer Web Services) [WPR10]. Na Figura 5.1 apresentamos uma visão simplificada da arquitetura do SIGA e abaixo percorremos novamente os passos de uma recuperação de anúncios detalhando os componentes e serviços envolvidos.



**Figura 5.1:** *Arquitetura do sistema de publicidade computacional.*

Uma requisição de anúncio é feita pelo navegador de um usuário quando ele visita uma página que contém espaços publicitários atendidos pelo sistema. A requisição de anúncio é recebida pelo **AdClient**, serviço responsável por recuperar todos os dados disponíveis para a exibição de anúncios. O **AdClient** obtém dados do usuário pelos *cookies* enviados junto com a requisição de anúncio, calcula dados de geo-localização e recupera dados do site armazenados em repositórios de acesso rápido, **SiteDB**. Esses repositórios contêm dados como listas de palavras-chave previamente extraídas da página pelo **Crawler** e listas de categorias previamente atribuídas ao site pelo seu dono, ou um especialista do UOL, por meio de interfaces específicas.

O *AdClient* constrói uma consulta de anúncios com os dados obtidos e a quantidade de anúncios desejada. Essa consulta é feita ao **AdServer** que é responsável por executar os principais passos da recuperação de anúncio. Inicialmente o **AdServer** transforma a consulta de anúncios em uma consulta vetorial e executa essa consulta no índice de anúncios, **AdIdx**, utilizando técnicas de recuperação de informação no modelo vetorial e similaridade de cossenos ponderada por *tf-idf* para recuperar uma quantidade de anúncios maior do que a quantidade requisitada. Nessa primeira fase,

os anúncios são selecionados e ordenados de acordo com o valor da similaridade de contexto e usuário.

O **AdServer** então executa a segunda fase, reordenando os anúncios de acordo com a função de reordenação. Para todos os anúncios retornados o **AdServer** calcula o valor da função de reordenação utilizando as dimensões de valor comercial, desempenho histórico, contexto e usuário. Por fim, o **AdServer** calcula o valor do custo por clique do anúncio e retorna o resultado da consulta para o **AdClient**, que por sua vez constrói a resposta da requisição com os anúncios recuperados e responde para o navegador que originou a requisição.

O índice de anúncios, **AdIdx**, é atualizado pelo **AdIndexer**, com base nos dados de anúncios da base de anúncios **AdDB**. Os dados de anúncios são gerenciados por meio de interfaces que utilizam o serviço **AdManager** para criar e modificar anúncios e campanhas.

Quando uma requisição é recebida pelo **AdClient** de uma página que não possui uma lista de palavras-chave associada, o endereço é registrado para que o **Crawler** obtenha a página e execute a sua análise, extraindo e armazenando as palavras-chave encontradas no **SiteDB**.

Essa arquitetura distribuída permite que o sistema implemente as funcionalidades de publicidade computacional respeitando as restrições de volume e tempo de resposta. Por outro lado, a construção da ordenação evolutiva exige que novas capacidades sejam introduzidas no sistema. Para a construção da ordenação evolutiva utilizaremos como alicerce a capacidade adaptativa. Para desenvolver essa capacidade de adaptação criamos optamos por desenvolver uma arquitetura de reconfiguração para o sistema de publicidade computacional. A capacidade de reconfiguração do sistema é a chave para que ele possa se tornar adaptativo. Por meio dela é possível descrever os parâmetros e as soluções candidatas de nossa abordagem.

Nosso trabalho tem como objetivo modificar o comportamento da ordenação de anúncios que ocorre dentro do componente **AdServer**. Para que isso fosse possível as modificações que implementamos no sistema não se restringiram a apenas esse único componente.

## 5.4 Implementação da capacidade adaptativa

A arquitetura distribuída descrita na seção anterior associada a uma infra-estrutura adequada permite que o sistema implemente as funcionalidades de publicidade computacional respeitando as restrições de volume e tempo de resposta. Além disso, esse sistema possui alta-disponibilidade e confiabilidade, sendo escalável para atender crescentes volumes de requisições de anúncios. Por outro lado, esse sistema ilustra os problemas tradicionais no desenvolvimento e modificação de

sistemas de software comerciais em produção. Necessidades de negócio fazem com que seja necessário modificar frequentemente o comportamento do sistema, sendo esperado que as modificações possam entrar em operação o mais rapidamente possível, com baixos riscos e sem impactos negativos no funcionamento do sistema. Para que nossa abordagem evolutiva pudesse ser implementada nesse sistema era necessário tornar a inclusão de modificações no sistema menos arriscada e mais ágil, reduzindo seu custo.

Uma das técnicas de engenharia de software que permite reduzir custos e riscos das mudanças é criar configurações que ao serem modificadas alteram o comportamento do sistema dentro de limites estabelecidos. Muitas aplicações, sistemas e bibliotecas de software possuem arquivos de configuração que permitem aos usuários, administradores ou desenvolvedores, modificar as definições que estabelecem os seus comportamentos. Porém, a solução tradicional de incluir múltiplos arquivos de configuração empacotados com o código da aplicação cria uma dependência direta do ciclo de desenvolvimento de software, e dos desenvolvedores, para que qualquer modificação seja aplicada. Por outro lado, disponibilizar arquivos de configurações fora do empacotamento da aplicação, por exemplo, no sistema de arquivos da máquina que executa a aplicação pode causar problemas de inconsistência e dificuldades de gerenciamento em ambientes distribuídos. A alternativa costuma ser a utilização de sistemas de arquivos distribuídos, centralizados ou a sincronização de arquivos. Essas últimas soluções associadas a característica que a maioria das aplicações necessita ser reiniciada para que modificações nas configurações sejam refletidas no seu comportamento aumentariam a chances de falhas e o acoplamento (*coupling*) entre os serviços ou partes do sistema. O baixo acoplamento é justamente uma das propriedades que permitem aos sistemas distribuídos alcançarem alta-disponibilidade, confiabilidade e escalabilidade.

Para superar esses problemas e possibilitar a implementação da abordagem evolutiva nós desenvolvemos o **ReConf** [ReC], um repositório que permite gerenciar as configurações de forma centralizada e as torna remotamente acessível por todos os componentes e serviços do sistema. A solução contém também uma biblioteca cliente opcional para a linguagem Java que permite, além de outras funcionalidades, a construção de aplicações e serviços que não precisam ser reiniciados para modificar seu comportamento quando uma configuração é alterada. O **ReConf** foi desenhado para prover os benefícios de configuração para sistemas de alta-disponibilidade, escaláveis e com baixo acoplamento, de forma a permitir que esses sistemas possam responder mais rapidamente a mudanças por meio de ajustes nas suas configurações.

## 5.5 ReConf: repositório centralizado de configurações

O **ReConf** não apenas desempenha um papel central na implementação da abordagem evolutiva, sendo a chave para tornar o sistema adaptativo, como também foi responsável por difundir a cultura de executar experimentos no ambiente real dentro do UOL. No melhor do nosso conhecimento, o **ReConf** é uma solução inovadora de código aberto [ReC] que torna disponível um repositório centralizado de configurações por meio de uma API de serviço web RESTful, descrita na próxima seção.

### 5.5.1 RESTful Web Service API

A ideia de construir sistemas distribuídos utilizando a arquitetura da *Web* inspirou a criação de abordagens como o estilo arquitetural orientado a recursos REST [WPR10]. Um dos principais conceitos da arquitetura da *Web* é que qualquer conteúdo ou elemento exposto na *Web*, de um documento ou música a um processo de negócio ou serviço, é um recurso. Um recurso *web* é unicamente identificado e endereçável por meio de um URI.

REST descreve a *Web* como uma aplicação hipermídia distribuída cujos recursos conectados se comunicam por meio de trocas de representação dos estados dos recursos. Ao invés de conhecer os possíveis estados previamente, REST defende a noção de HATEOAS (*Hypermedia As The Engine Of Application State*), ou seja, conforme a aplicação atinge um novo estado ela descobre possíveis novas transições de estado.

Leonard Richardson propôs em [RR07] uma classificação para serviços na *Web* que foi utilizada em [WPR10] como um modelo para quantificar discussões sobre maturidade de serviços. O modelo possui quatro níveis (0-3) e é direcionado pelas funcionalidades que o serviço oferece para URIs, HTTP e hipermídia.

Nós defendemos que a implementação de referência da API do **ReConf**, chamada de *ReConf server*, é um serviço RESTful de nível três, provendo as seguintes funcionalidades:

- Diferentes recursos endereçados por URIs distintas.
- Suporte a múltiplos verbos HTTP em cada recurso disponível (ao menos GET e, ou PUT ou DELETE).
- Idempotência, ou seja, GET não causa nenhuma mudança de estado no lado do serviço e múltiplas execuções do mesmo PUT ou DELETE levam ao mesmo estado final.
- Respostas contêm códigos de status HTTP adequados.

- Respostas contêm *links* de URIs para outros recursos ou transições disponíveis, fornecendo uma trilha que pode levar à uma transição de estado da aplicação (HATEOAS).

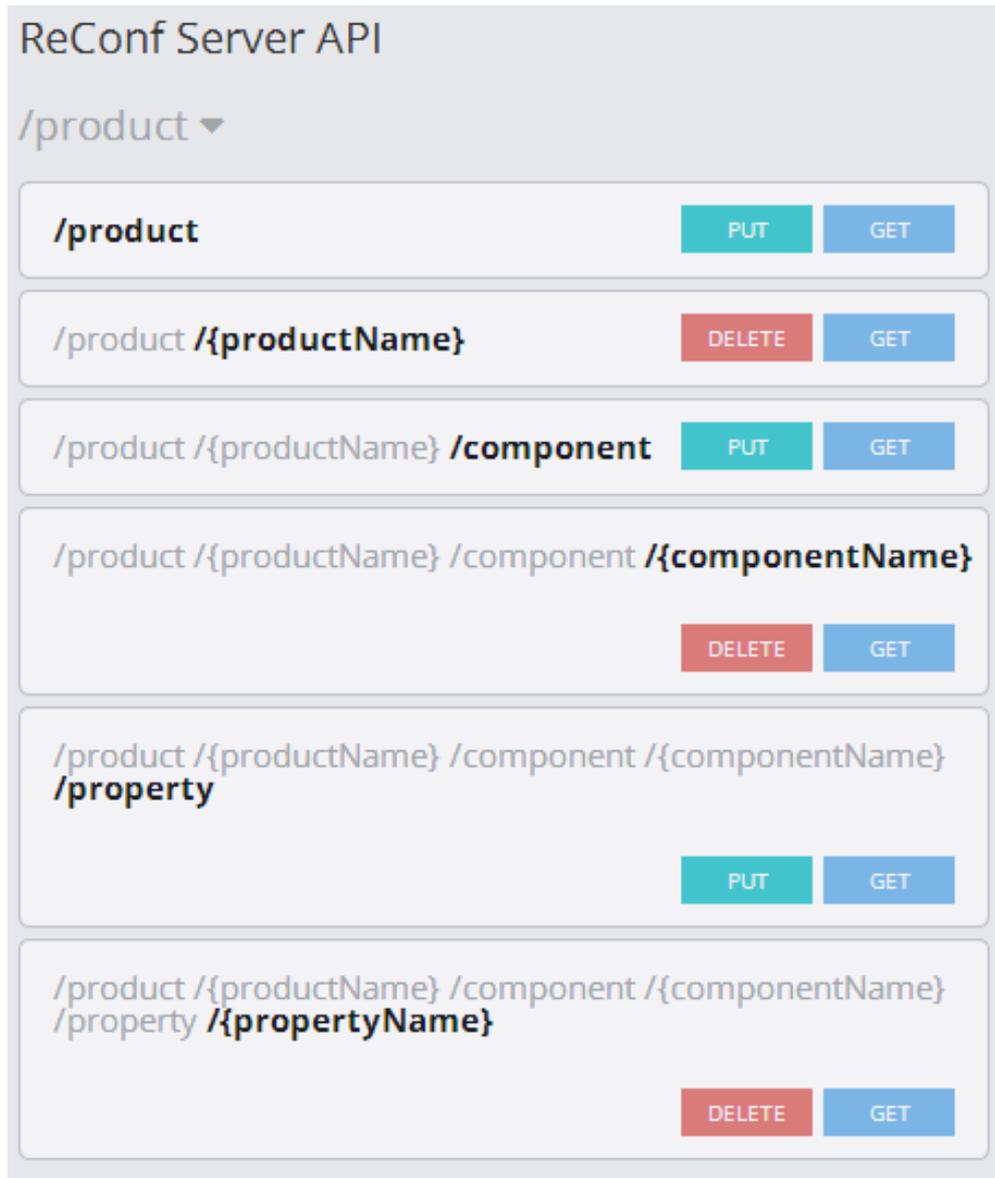
O modelo de configuração do **ReConf** é uma composição de três níveis de elementos, como ilustrado na Figura 5.2. O nível raiz é "Produto", por exemplo, *publicidade*. O nível intermediário permite uma divisão lógica de configurações, denominado "Componente" para representar partes de um produto. Por exemplo, no sistema de publicidade computacional, poderíamos criar componentes como: *ordenacao*, *leilao*, *gerenciador* ou *crawler*. O último nível, "Propriedade", é o nível que realmente armazena os valores das configurações, por exemplo, *funcao*, como uma propriedade do componente de *ordenacao* que armazena o valor  $CPCbid_a \times CTR_a$ . Propriedades são representadas como *Strings*, ou sequências de caracteres. Tipos complexos podem ser armazenados em qualquer formato de representação textual, como CSV ou JSON. A aplicação cliente deve ser capaz de ler e interpretar esses tipos. A biblioteca cliente Java também provê algumas funcionalidades de interpretação dos tipos dessa linguagem. A API básica do **ReConf** exibida na Figura 5.2 permite o gerenciamento dos elementos para criar, recuperar, atualizar e excluir configurações.

### 5.5.2 Biblioteca cliente de configuração

O serviço disponibilizado pelo **ReConf** permite acesso remoto às configurações. Qualquer modificação em um valor de uma configuração se torna visível por qualquer parte do sistema que recupere esse valor de configuração do serviço. Por outro lado, recuperar cada configuração do serviço cada vez que ela for necessária pode resultar em uma sobrecarga desnecessária e degradação do tempo de resposta de todo o sistema. Mecanismos intermediários de *cache* podem auxiliar nessa situação, porém o ideal é que os componentes e serviços do sistema não precisem aguardar uma requisição externa, mesmo *cacheada*, uma vez que a restrição de tempos de resposta, abaixo de 100ms, já é desafiadora sem considerar a recuperação de valores de configurações.

A solução alternativa de recuperar as configurações apenas na inicialização é ainda menos favorável devido ao tempo necessário para que a aplicação modifique seu comportamento com base no novo valor de uma configuração. Sobretudo, uma reinicialização pode ser necessária após cada modificação de algum valor de configuração, reduzindo a disponibilidade do sistema durante o período de reinicialização.

A biblioteca cliente de configurações JAVA, **ReConf JVM**, não apenas provê uma maneira mais fácil de integração com o serviço de configurações **ReConf**, como também disponibiliza a funcionalidade que denominamos de *smart configuration*. Esse tipo de configuração é capaz de



**Figura 5.2:** *ReConf API - com os verbos HTTP implementados. Essa Figura foi gerada pela aplicação [www.api.hub.com/raml/api-designer](http://www.api.hub.com/raml/api-designer).*

automaticamente recuperar novos valores e se recarregar de tempos em tempos, sem a necessidade de reiniciar a aplicação.

Para utilizar a biblioteca **ReConf JVM** a aplicação precisa ser desenvolvida na linguagem Java e as configurações devem ser armazenadas como texto em um formato especialmente desenvolvido para que a biblioteca seja capaz de criar o objeto java apropriado de acordo com o tipo de dado armazenado. A biblioteca depende que um simples arquivo XML de configuração contendo poucos dados sobre o ambiente de execução esteja disponível junto à aplicação, cuja principal informação é o endereço do serviço de configuração.

De maneira mais direta, para desenvolver uma aplicação Java utilizando a biblioteca **ReConf JVM** o desenvolvedor deve:

- Incluir o pacote (*jar*) do **ReConf JVM** no projeto;
- Importar o *reconf.client*;
- Disponibilizar um arquivo de configuração *reconf.xml* no *classpath* da aplicação;
- Criar interfaces tradicionais Java, POJI (*Plain Old Java Interface*), decorando-as com anotações customizadas para as configurações remotas;
- Executar um método de fábrica, *factory* para obter uma instância da interface de configuração.

A Figura 5.3 apresenta um exemplo de uma interface simples de configuração para o produto *advertising*, componente *ranking*. Nesse exemplo as configurações são recarregadas a cada 10 minutos. Existem duas propriedades de configuração: a propriedade *function* que armazena uma função como uma cadeia de caracteres, *String* e a propriedade *minimum.bid* que recupera um número como um objeto Java do tipo *BigDecimal*. Exemplos de valores textos para essas propriedades de configuração seguindo o formato citado poderiam ser, respectivamente: ' $CPCbid_a \times CTR_a$ ' e '0.15'.

```
import reconf.client.annotations.*;

@ConfigurationRepository(product="advertising",
                        component="ranking", pollingRate=10,
                        pollingTimeUnit=TimeUnit.MINUTES)
public interface RankingConfiguration {
    @ConfigurationItem("function")
    String getFunction();

    @ConfigurationItem("minimum.bid")
    BigDecimal getMinBid();
}

public static void main(String[] args) {
    RankingConfiguration rankConf =
        ConfigurationRepositoryFactory.get(
            RankingConfiguration.class);
    System.out.println(rankConf.getFunction());
}
```

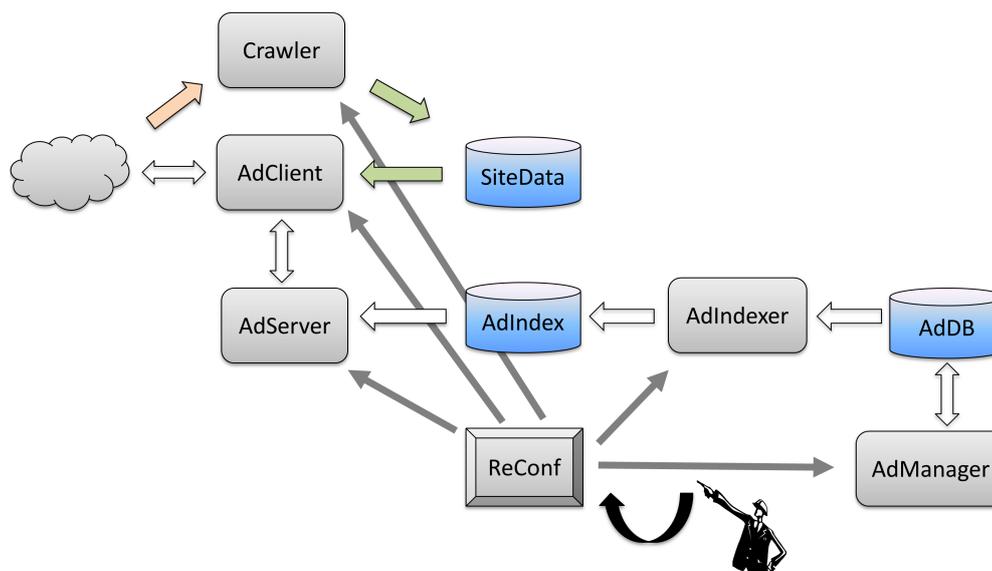
**Figura 5.3:** Exemplo de interface de configuração utilizando a biblioteca cliente em Java.

As principais vantagens da biblioteca cliente são as *smart configurations* e a facilidade para a integração, mas não as únicas funcionalidades fornecidas, entre outras estão:

- **Persistência local** - uma vez que um valor de configuração é recuperado com sucesso ele é armazenado localmente, o que permite à aplicação ter acesso ao último valor de configuração recuperado quando o serviço de configuração não está acessível. Com o valor armazenado localmente ele está disponível para a aplicação mesmo no cenário que a reinicialização da aplicação ocorrer no momento em que o serviço de configuração estiver indisponível;
- **Recuperação atômica** - a biblioteca de configuração considera todas as configurações agrupadas no mesmo produto, componente e interface como uma unidade atômica de recuperação e carregamento. Se por alguma razão um valor não puder ser recuperado, analisado ou instanciado, nenhum dos valores agrupados será atualizado, permitindo assim que o comportamento seja controlado de forma consistente por mais de uma propriedade de configuração quando for necessário;
- **Análise de tipos** - possui um conjunto de analisadores pré-construídos para diversos tipos da linguagem Java, como os tipos primitivos, seus *wrappers*, *arrays* e as principais interfaces no pacote *java.util* como *Collection*, *List*, *Set* e *Map*, assim como qualquer classe Java que possua um construtor que receba uma *String* como parâmetro. O formato de texto foi especialmente desenvolvido para permitir uma especificação legível e precisa de configurações como coleções de valores, ou seja, vetores, listas e mapas.

## 5.6 Arquitetura de reconfiguração

Com o desenvolvimento do **ReConf** foi possível modificar os componentes e serviços do sistema de publicidade computacional SIGA reorganizando a sua arquitetura de forma a utilizar o repositório centralizado para gerenciar as configurações. Essa arquitetura é muito semelhante à apresentada na seção 5.3, incluindo apenas o **ReConf**, como podemos ver na Figura 5.4. A interação com o **ReConf** pode ocorrer diretamente pela API (setas curvas), como no caso da interface para os desenvolvedores e operadores modificarem as configurações, ou diretamente pela biblioteca cliente (setas retas) no caso dos componentes e serviços Java do sistema. Essa modificação foi o primeiro passo em direção a uma arquitetura adaptativa. Apresentaremos nessa seção os benefícios que esse primeiro aprimoramento trouxe ao ambiente de publicidade computacional.



**Figura 5.4:** Arquitetura do sistema de publicidade computacional incluindo a capacidade de reconfiguração.

### 5.6.1 Demandas de negócio

Em ambientes de publicidade computacional diferentes demandas de negócios podem criar a necessidade de realizar diferentes mudanças no comportamento do sistema. Usualmente essas demandas são originadas pelos patrocinadores ou *stakeholders* do negócio como sócios, presidente, diretores, gerentes comerciais entre outros, que financiam o desenvolvimento do sistema e esperam alcançar bons resultados com esse negócio. Exemplos de demandas nesses ambientes poderiam ser que um patrocinador deseja melhorar a receita aumentando o valor mínimo para a aposta do sistema ou em outro momento pode desejar ampliar o número de anunciantes possibilitando apostas de menor valor. Esses exemplos de demandas têm efeitos no gerenciamento dos anúncios, uma vez que restringe a escolha da aposta como também no cálculo do custo do clique, antes da exibição do anúncio.

Em um ambiente tradicional de desenvolvimento, sem utilização de configuração, essa modificação deve ser requisitada ao time de desenvolvimento que irá: codificar a mudança nos dois serviços afetados, **AdServer** e **AdManager**; testar esses serviços; preparar a entrega; e implantar as versões com o comportamento modificado no ambiente de produção. Nas empresas com métodos de desenvolvimento mais evoluídos isso pode demorar menos que uma hora. Porém, na nossa realidade do SIGA, antes da utilização do **ReConf**, este tipo de mudança levaria, na média, algo entre 3 dias e 2 semanas .

## Solução com o ReConf

Utilizando o **ReConf**, esse comportamento poderia ser implementado como no exemplo da Figura 5.3. O desenvolvimento do comportamento dependente da configuração teria um esforço equivalente ao tradicional, porém, mudanças subsequentes nessa característica do sistema podem seguir um fluxo muito mais eficiente. Após a modificação ser requisitada ao time de desenvolvimento, o time irá: analisar se existe algum risco em aplicar a modificação diretamente no ambiente de produção; se necessário, aplicar a modificação no ambiente de homologação e aplicar os testes; e então aplicar a configuração no ambiente de produção. Aplicar essa modificação não envolve mais a codificação da modificação do comportamento, apenas a chamada do serviço web do **ReConf** para atualizar um valor de configuração de forma que todos os componentes e serviços afetados possam acessar o novo valor. A chamada da API pode ser simplificada pela criação de uma interface de usuário, que pode ser facilmente desenvolvida utilizando os serviços da API do **ReConf**.

## Resultados obtidos

Esse tipo de modificação costumava demorar ao menos alguns dias para estar disponível no sistema em produção. Utilizando o **ReConf** na maioria dos casos uma mudança pode estar disponível em alguns minutos. Para modificações mais complexas, como por exemplo uma nova solução de ordenação de anúncios, cuidados adicionais podem ser necessários e o tempo será condicionado ao tempo para testar a modificação, usualmente algumas poucas horas. Para configurações mais simples, que modificam sutilmente o comportamento do sistema e usualmente são configuradas como números, tipos ou caracteres, a responsabilidade por aplicar as modificações pode ser transferida do time de desenvolvimento para os operadores do sistema ou mesmo os patrocinados, permitindo a eles o acesso a uma interface capaz de restringir e controlar as mudanças.

O trabalho para o time de desenvolvimento passa a ser identificar quais demandas devem ser desenvolvidas como comportamentos condicionados a configurações. A regra mais simples e ser utilizada é codificar o comportamento normalmente e sem configurações, quando uma mudança nesse comportamento for necessária, analisar o cenário para identificar se é possível implementar esse comportamento condicionado a configurações, fazendo os refatoramentos necessários, permitindo assim que novas modificações sejam feitas sem a necessidade de codificação e entrega de novas versões dos componentes e serviço, ao mesmo tempo que evita-se o cenário de criação de muitas configurações que podem nunca ser necessárias.

### 5.6.2 Experimentos controlados

Em ambientes comerciais e sobretudo em ambientes científicos a maioria das propostas de mudanças são apresentadas como hipóteses a serem avaliadas, ao invés de uma decisão definitiva. Felizmente, esse modelo tem se tornado cada vez mais frequente com a tendência de aplicar ciência da dados à tomada de decisões gerenciais. Modificar todo o comportamento de um sistema para verificar uma hipótese pode causar efeitos colaterais indesejados ou mesmo problemas irreversíveis. Isso tornam mais valiosas abordagens que aplicam experimentos controlados antes de produzir mudanças globais.

Técnicas como Teste A/B [DLG14b, KDLX14] podem ser utilizadas, tratando parte das instâncias, ou pacientes, com um comportamento/tratamento de controle (A) e uma outra parte equivalente com o comportamento/tratamento da hipótese (B). Dados são rotulados e coletados para cada um dos grupos e o desempenho da hipótese é comparado com o do controle dentro de critérios pré-estabelecidos. Em ambientes como o de publicidade computacional um exemplo poderia ser a proposta de uma nova solução de ordenação de anúncios, *f2*. Poderia essa solução alcançar melhores valores de CTR e eCPM que a solução atual, *f1*? Essa técnica de experimentação inspirou a nossa abordagem evolutiva como descrevemos no capítulo 4, da mesma forma como foi aplicada no experimento das dimensões apresentado na seção 7.3.

Antes da solução de configuração diferentes soluções eram avaliadas desenvolvendo/codificando a nova ordenação *f2* e executando ela por um período de tempo, por exemplo uma semana, e comparando os indicadores desse período com os indicadores do período anterior. Não apenas demorava um longo período para a hipótese ser avaliada (ao menos três dias de desenvolvimento e uma semana de execução), como também expunha todo o sistema a um grande risco. Mesmo que fosse possível identificar que uma nova solução não alcançou bom desempenho, pouco tempo após sua implantação, poderia ser difícil reverter para a antiga solução rapidamente, ao mesmo tempo em que danos significativos podem já ter sido causados.

Uma outra questão é que comparar duas diferentes soluções durante dois intervalos de tempo diferentes poderia levar a uma conclusão incorreta sobre a hipótese. No ambiente real, existe um grande número de variáveis ou parâmetros que podem influenciar o comportamento do sistema. Além disso, não podemos garantir o controle sobre todas essas variáveis. Como consequência, pequenas mudanças locais de comportamento em um dos períodos poderiam ser suficientes para invalidar todo o experimento ou até mesmo validar incorretamente uma hipótese errada.

## Solução com o ReConf

Para utilizar a técnica de Teste A/B as aplicações devem ser devidamente desenvolvidas. Isso pode ser feito utilizando padrões de desenho de software como *Strategy*, *Dispatcher*, *Chain-of-responsibility* entre outros [GHJV95]. Dois ou mais fluxos ou caminhos de execução devem estar disponíveis e podem utilizar diferentes configurações. Por exemplo, diferentes soluções de ordenação de anúncios podem estar disponíveis como *function*, *function-A* e *function-B*. A biblioteca cliente do **ReConf** também auxilia nesse desenvolvimento permitindo o uso de prefixos e sufixos customizados [ReC]. Uma outra configuração, por exemplo, *ABsample*, pode armazenar o tamanho dos grupos para os experimentos, por exemplo, *10*. Dessa forma, o sistema pode ser desenvolvido para processar 10% das requisições de anúncios com a *function-A*, outros 10% com a *function-B* e o restante com a solução principal *function*.

## Resultados obtidos

Desenvolvimento adicional se faz necessário para tornar disponível um ambiente para experimentação controlada. Uma vez que isso seja feito, o **ReConf** permite a definição e início de experimentos tão rápido quanto modificar qualquer outra configuração: em questão de minutos ou poucas horas. Não apenas a exposição e o risco podem ser melhor administrados durante a execução dos experimentos por meio de variáveis de controle (*ABsample*), como também o experimento pode ser interrompido facilmente, por exemplo, atribuindo zero ao tamanho da amostra do experimento.

Esse tipo de ambiente permitiu a disseminação da ideia de experimentação e aumentou consideravelmente o número de hipóteses sugeridas e avaliadas no ambiente de publicidade computacional atendido pelo sistema SIGA. Anteriormente, menos de 10 hipóteses eram avaliadas em um ano, com a maioria delas sendo inconclusivas. Com esse novo ambiente, foi possível alcançar o status de que qualquer ideia pode ser testada sem muita discussão prévia, em questão de horas ou poucos dias. Mais de 40 hipóteses foram avaliadas em um mesmo mês, chegando a mais de 80 no primeiro ano da implementação desse ambiente. É importante ressaltar que ideias disruptivas podem ainda depender de desenvolvimento, pois isso pode levar a novos comportamentos e novos caminhos de execução que ainda não foram mapeados e desenvolvidos.

A execução de experimentos controlados por meio de **ReConf** implementa o primeiro passo da nossa abordagem, **Exploração**, assim como prepara o ambiente para a **Avaliação**, que já é possível de ser executada manualmente nesse experimentos controlados.

## 5.7 Conclusão

Para tornar o sistema adaptativo desenvolvemos, neste trabalho, o **ReConf**, que permite gerenciar configurações que possibilitam aos componentes e serviços modificarem seu comportamento com base nas alterações dos valores de suas configurações. O elemento que falta para complementar a capacidade adaptativa é um serviço ou componente capaz de identificar modificações no ambiente e ajustar os valores das configurações de forma a modificar o comportamento do sistema em resposta a essas modificações no ambiente. No próximo capítulo, **Implementação**, descrevemos como implementamos a abordagem evolutiva para capacitar o sistema de publicidade computacional a adaptar e evoluir a sua ordenação de anúncios.

## Capítulo 6

# Implementação da ordenação evolutiva

Conforme apresentamos no capítulo 4, nossa abordagem evolutiva possui um ciclo principal constituído por três passos: **Exploração**, **Avaliação** e **Evolução**. Nas capítulo anterior descrevemos como modificamos o sistema de publicidade computacional SIGA para possibilitar a exploração de soluções por meio de experimentos controlados, seção 5.6.2, implementando assim o primeiro passo da nossa abordagem. Nesses experimentos o segundo passo, **Avaliação**, era executado manualmente para verificar os resultados. Neste capítulo iremos descrever como implementamos automaticamente o segundo e terceiro passos, assim como toda a nossa abordagem evolutiva, tornando o sistema SIGA capaz de adaptar e evoluir a sua ordenação de anúncios.

### 6.1 Escopo

Conforme fizemos ao definir o objetivo em 1.2, vamos restringir o escopo de publicidade que abordaremos a:

- **anúncios de texto;**
- **publicidade contextualizada;**
- **modelo de CPC**, custo por clique.

### 6.2 Restrições de dados

Para comparar os resultados de diferentes funções de ordenação vamos utilizar a função objetivo descrita em 2.6. Inicialmente, precisamos calcular os indicadores do sistema:  $\overline{CTR}$ ,  $\overline{ROI}$  e  $\overline{eCPM}$ . Observando as definições das raízes desses indicadores de sistema 2.1, 2.3 e 2.4, identificamos que

para calcular esses valores precisamos da receita total do sistema  $Receita_s$ , da quantidade de impressões  $Impressoes_s$ , da quantidade de cliques  $Cliques_s$  e do retorno que os anunciantes obtiveram com a publicidade no sistema  $Retorno_s$  para o período considerado, que será de um dia.

Três desses valores podem ser coletados diretamente pelo sistema:  $Receita_s$ ,  $Impressoes_s$  e  $Cliques_s$ . Infelizmente o quarto valor ( $Retorno_s$ ) é mais difícil de obter no modelo de CPC, o que já não ocorre no modelo de CPA, no qual é importante e interessante para o anunciante fornecer a informação de conversão. No modelo de CPC os anunciantes não têm benefício em oferecer informações de retorno da publicidade, muitos não implementam a coleta desse tipo de informação e outros consideram essa informação sigilosa e não a compartilham com o sistema. Sem esse valor não é possível calcular o indicador  $\overline{ROI}$ , que consideramos o melhor indicador para a utilidade do anunciante. Por outro lado, no modelo de CPC os anunciantes usualmente utilizam o CTR de seus anúncios para avaliar o desempenho da publicidade. Apesar das imperfeições, no modelo de CPC o CTR é usualmente utilizado como um representante da utilidade do anunciante.

### 6.2.1 Função objetivo ajustada

Considerando essas restrições e recuperando a definição da função objetivo:

$$F_{obj} = \alpha \overline{CTR} + \beta \overline{ROI} + \gamma \overline{eCPM} - \delta \sigma^2$$

$$1 = \alpha + \beta + \gamma$$

$$\delta \geq 0$$

E da variância:

$$\sigma^2 = \frac{(\overline{CTR} - \mu)^2 + (\overline{ROI} - \mu)^2 + (\overline{eCPM} - \mu)^2}{3}$$

$$\mu = \frac{\overline{CTR} + \overline{ROI} + \overline{eCPM}}{3}$$

Idealmente, poderíamos utilizar os parâmetros de ponderação,  $\alpha$ ,  $\beta$  e  $\gamma$  com mesmo valor e o peso da variância  $\delta$  como dois, aplicando assim a redução de duas variâncias. Dessa forma, buscaríamos encontrar funções de ordenação mais equilibradas na utilidade de cada participante. Porém, devidos as restrições para obter o componente  $\overline{ROI}$ , vamos desconsiderar netes trabalho o componente  $\overline{ROI}$ , atribuindo zero ao parâmetro  $\beta$ . Em contrapartida, iremos dobrar a ponderação do  $\overline{CTR}$ , utilizando-o uma vez a mais assumindo assim o papel do  $\overline{ROI}$ . Além disso, vamos ajustar o cálculo da variância para considerar apenas os dois indicadores. Com um pouco de álgebra podemos simplificar e obter

a seguinte função objetivo:

$$Fobj = \alpha \overline{CTR} + \beta \overline{ROI} + \gamma \overline{eCPM} - \delta \sigma^2$$

$$1 = \alpha + \beta + \gamma$$

$$\alpha = \frac{2}{3}$$

$$\beta = 0$$

$$\gamma = \frac{1}{3}$$

$$\delta = 2$$

$$Fobj = \frac{2}{3} \overline{CTR} + 0 \overline{ROI} + \frac{1}{3} \overline{eCPM} - 2\sigma^2$$

$$Fobj = \frac{2}{3} \overline{CTR} + \frac{1}{3} \overline{eCPM} - 2\sigma^2$$

$$\sigma^2 = \frac{\sum_{i=1}^N (\overline{X}_i - \mu)^2}{N}$$

$$\mu = \frac{\sum_{i=1}^N \overline{X}_i}{N}$$

$$N = |\overline{X}|$$

$$\overline{X} = \{\overline{CTR}, \overline{eCPM}\}$$

$$N = 2$$

$$\sigma^2 = \frac{(\overline{CTR} - \mu)^2 + (\overline{eCPM} - \mu)^2}{2}$$

$$\mu = \frac{\overline{CTR} + \overline{eCPM}}{2}$$

$$Fobj = \frac{2}{3} \overline{CTR} + \frac{1}{3} \overline{eCPM} - 2\sigma^2 \tag{6.1}$$

$$\sigma^2 = \frac{(\overline{CTR} - \mu)^2 + (\overline{eCPM} - \mu)^2}{2}$$

$$\mu = \frac{\overline{CTR} + \overline{eCPM}}{2}$$

Acreditamos que a abordagem também será válida caso o  $\overline{ROI}$  possa ser calculado, para isso deveríamos utilizar a função objetivo original, ajustando os parâmetros conforme o necessário.

### 6.3 Arquitetura evolutiva

A Figura 6.1 apresenta a inclusão de um novo componente **Avaliador**, que será responsável por executar a avaliação das soluções de ordenação de anúncios com base nos dados coletados durante a exploração, implementando o passo de **Avaliação** da nossa abordagem. Esse componente controla os ciclos da abordagem evolutiva e delega a um motor a implementação do mecanismo de **Evolução**, o terceiro passo da abordagem evolutiva. O **Avaliador** também controla o segundo estágio de avaliação, assim como a promoção de soluções candidatas utilizando o **ReConf** para armazenar e comunicar as modificações a todo o sistema.

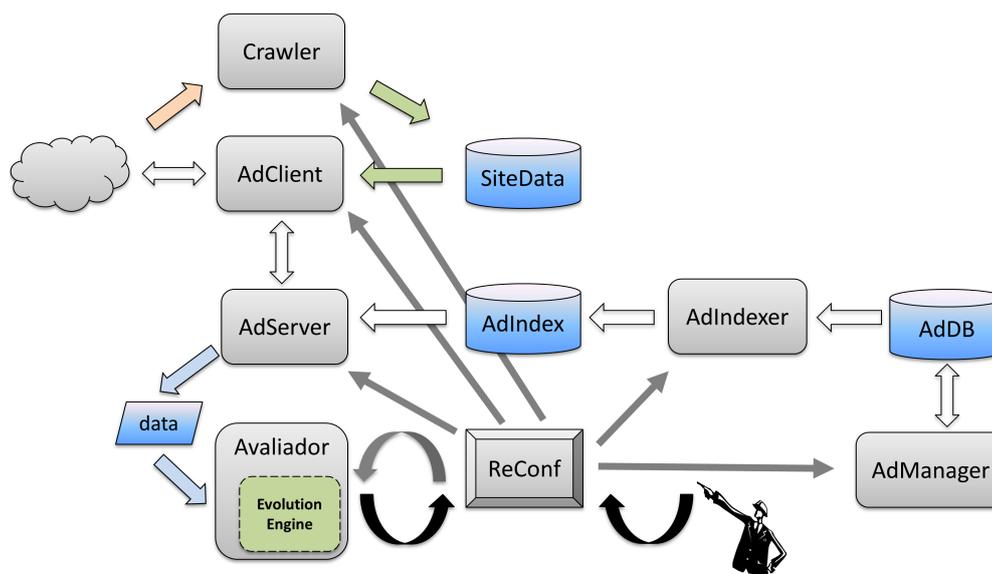


Figura 6.1: Arquitetura evolutiva do sistema de publicidade computacional.

### 6.4 Avaliação

O componente **Avaliador** implementa a maior parte do modelo cíclico de evolução da nossa abordagem. Como descrito anteriormente, um experimento é definido por meio de configurações armazenadas e gerenciadas no **ReConf**. Essas configurações incluem as estratégias de ordenação de anúncios, assim como as fórmulas ou funções de ordenação e os tamanhos das amostras. O **Adserver** é responsável por executar os experimentos e fornecer dados para a avaliação do desempenho.

Em nossa implementação, essas configurações definiram 40 soluções candidatas de ordenação de

anúncios para serem exploradas. Também armazenou configurações para soluções adicionais como a solução de controle, a melhor solução conhecida até o momento e uma solução no segundo estágio de avaliação.

Primeiro, o **Avaliador** recupera os dados disponibilizados pelo **AdServer**. Esses dados contêm a requisição de anúncios, os anúncios exibidos como resposta, o custo por clique de cada anúncio, anúncios que foram clicados pelo usuário após serem exibidos e a solução de ordenação responsável por ordenar esses anúncios.

Segundo, o **Avaliador** calcula o desempenho de cada uma das soluções por meio da função objetivo ajustada, 6.1. Terceiro, o **Avaliador** executa o **EvolutionEngine** para criar um novo conjunto de soluções candidatas de ordenação de anúncios. Por último, o **Avaliador** armazena o novo conjunto de soluções candidatas no **ReConf**, para serem executadas no próximo período de exploração. Também verifica se alguma solução deve ser promovida e então grava a fórmula da solução na adequada propriedade de configuração no **ReConf**. Armazenadas no **ReConf**, as soluções estão disponíveis para que o **AdServer** execute a exploração.

## 6.5 Evolução

O **EvolutionEngine** é responsável por possibilitar a evolução de uma geração de soluções candidatas de ordenação de anúncios para uma nova geração, a fim de executar novos experimentos e avaliações. Cada solução candidata é configurada como uma estratégia de ordenação de anúncios diferente e seu principal atributo é sua fórmula de ordenação. O motor é executado pelo **Avaliador**, recebendo como parâmetros a geração anterior de fórmulas de ordenação e os dados de seu desempenho. O **EvolutionEngine** então retorna como resposta uma nova geração de fórmulas de ordenação de anúncio.

O **EvolutionEngine** foi implementado utilizando a técnica de programação genética. Acreditamos que a utilização de outras técnicas, como por exemplo SVM, pode funcionar adequadamente na nossa abordagem evolutiva. O **EvolutionEngine** tem como alicerce a biblioteca JGAP, *jgap.sourceforge.net*. Para definirmos o modelo de programação genética precisamos definir três componentes principais: os indivíduos, os operadores genéticos e a função de *fitness*. Além disso, discutimos como é feita a seleção e evolução dos indivíduos.

### 6.5.1 Indivíduos

Os indivíduos representam as possibilidades de solução para o problema: funções de ordenação. Para representar funções matemáticas em programação genética, a estrutura mais apropriada é uma árvore, que torna mais fácil sua implementação e interpretação. Essas árvores serão compostas por dois tipos de nós: nós internos e folhas. Os nós internos assumirão valores de funções matemáticas:

- $+$ , adição;
- $-$ , subtração;
- $/$ , divisão, se necessário podemos utilizar a divisão protegida, que resulta em 1 no caso de divisão por zero;
- $\times$ , multiplicação;
- $lg$ , logaritmo na base 10.

Chamaremos essas funções de operadores. Os operadores descritos são binários ou unários e induzirão uma árvore. Essas funções matemáticas foram escolhidas por pertencerem ao repertório clássico de componentes de funções de ordenação de recuperação de informação. Adição e multiplicação para reforçar relações de diferentes graus, divisão para relações inversas, logaritmo para amortizar valores e subtração para redução ou correção.

As folhas representam as constantes e as variáveis disponíveis para o problema, ou seja, as variáveis disponíveis do anúncio  $a$ , para ser exibido no contexto  $c$ , ao usuário  $u$ , considerando cada uma das quatro dimensões (valor comercial, desempenho histórico, contexto e usuário):

- $CPCbid_a$ , valor da aposta, valor máximo que o anunciante pagará por um clique no anúncio;
- $CTR_a$ , taxa de clique, razão da quantidade de cliques pela quantidade de impressões do anúncio;
- $CTX_{a,c}$ , similaridade de cossenos dos vetores do anúncio e do contexto;
- $USR_{a,u}$ , similaridade de cossenos dos vetores do anúncio e do usuário.

Além das variáveis do anúncio, teremos variáveis estatísticas (média, máximo e mínimo) das dimensões, que são calculadas na fase anterior sobre todo o conjunto de anúncios selecionados:

- $CPCbid_{avg}$ , média do valor das apostas;
- $CPCbid_{max}$ , máximo valor das apostas;

- $CPCbid_{min}$ , mínimo valor das apostas;
- $CTR_{avg}$ , média do valor da taxa de clique;
- $CTR_{max}$ , máximo valor da taxa de clique;
- $CTR_{min}$ , mínimo valor da taxa de clique;
- $CTX_{avg}$ , média do valor da similaridade de cossenos de contexto;
- $CTX_{max}$ , máximo valor da similaridade de cossenos de contexto;
- $CTX_{min}$ , mínimo valor da similaridade de cossenos de contexto;
- $USR_{avg}$ , média do valor da similaridade de cossenos de usuário;
- $USR_{max}$ , máximo valor da similaridade de cossenos de usuário;
- $USR_{min}$ , mínimo valor da similaridade de cossenos de usuário.

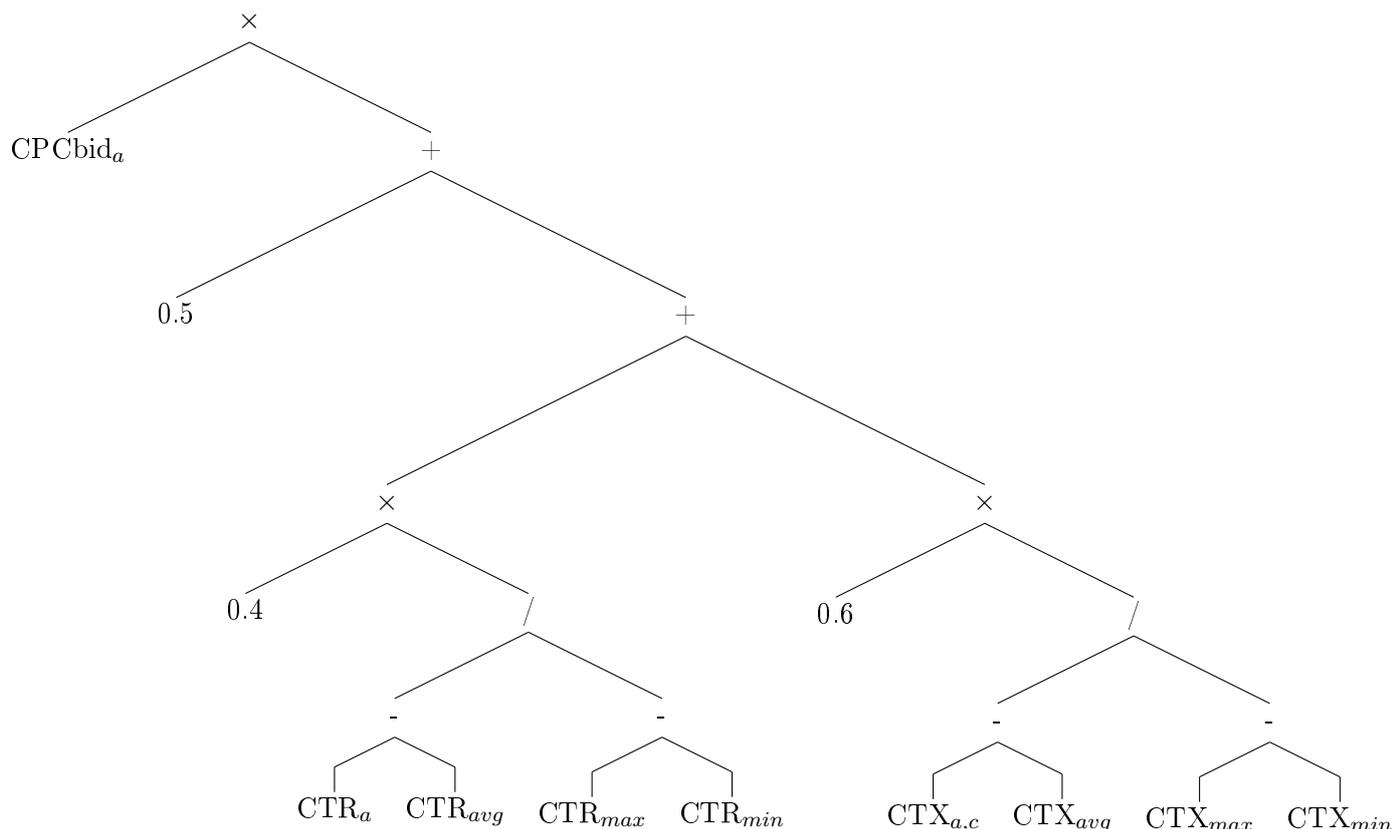
Uma melhoria que poderá ser feita em trabalhos futuros é a inclusão de medidas estatísticas de variância, como o desvio padrão. Constantes aleatoriamente escolhidas (na primeira vez que são necessárias e então fixadas na função) se unem às variáveis para formar o conjunto de folhas ou terminais disponível. Como exemplo, vamos representar a seguinte função como uma árvore na Figura 6.2:

$$CPCbid_a \times \left( 0.5 + \left( 0.4 \times \frac{CTR_a - CTR_{avg}}{CTR_{max} - CTR_{min}} + 0.6 \times \frac{CTX_{a,c} - CTX_{avg}}{CTX_{max} - CTX_{min}} \right) \right)$$

### 6.5.2 Operadores genéticos

Utilizaremos os operadores genéticos tradicionais: **reprodução**, **crossover** e **mutação**, adequados à nossa estrutura de árvore, além da criação de novos indivíduos aleatoriamente. A cada geração uma certa quantidade de novos indivíduos serão gerados aleatoriamente e inseridos no processo. Esses indivíduos não herdarão genes da geração anterior. Na reprodução a árvore que representa o indivíduo será copiada como um novo indivíduo da nova geração, sem a introdução de nenhuma modificação.

No *crossover*, dois indivíduos selecionados como pais irão gerar dois novos indivíduos. Inicialmente se escolhe aleatoriamente um nó na árvore do primeiro pai. Em seguida, é escolhido aleatoriamente um nó na árvore do segundo pai. Um novo indivíduo é formado pela árvore do primeiro

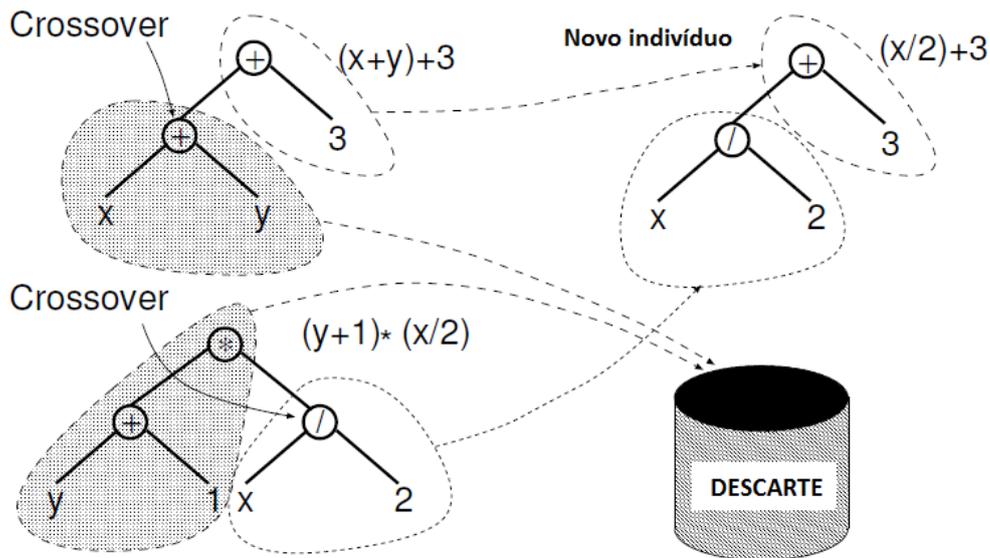


**Figura 6.2: Representação de uma função como uma árvore.**

pai substituindo a sub-árvore do nó escolhido pela sub-árvore do nó escolhido no segundo pai. O processo é repetido utilizando agora o segundo pai primeiro e selecionando novos nós aleatórios para as sub-árvores, gerando assim dois novos indivíduos para a próxima geração que carregam parte dos genes de cada um dos pais.

Na *mutação* um nó da árvore é escolhido aleatoriamente e uma alteração pontual do gene nesse nó poderá ocorrer de acordo com o conteúdo do nó:

- operador  $\times$ , será convertido para operador  $/$ ;
- operador  $/$ , será convertido para operador  $\times$ ;
- operador  $+$ , será convertido para operador  $-$ ;
- operador  $-$ , será convertido para operador  $+$ ;
- constante, será convertida para um novo valor de constante escolhido aleatoriamente;
- variável, será convertida para uma variável escolhida aleatoriamente dentre as variáveis disponíveis.



**Figura 6.3: Crossover** -Os dois indivíduos selecionados para reprodução, à esquerda, têm nós selecionados aleatoriamente como pontos de crossover. A combinação da árvore do primeiro indivíduo substituindo a sub-árvore de crossover pela sub-árvore do segundo indivíduo cria o novo indivíduo, filho da geração anterior. As partes não utilizadas são descartadas. Para gerar o segundo filho, o processo é repetido invertendo os pais e escolhendo-se novos pontos de crossover. Essa Figura foi adaptada da Figura 2.5, pag 16, de [PLM08].

### 6.5.3 Função de *fitness*

A função de *fitness* será a função objetivo ajustada 6.1 definida na seção 6.2.1. Os sistemas, ou funções de ordenação, terão a função objetivo calculada fixando um sistema e uma função de ordenação única como referência. Essa função de referência usualmente será a melhor função de ordenação encontrada antes da execução e avaliação da geração atual.

### 6.5.4 Seleção

Para selecionar os indivíduos para reprodução utilizaremos as técnicas de **elitismo**, **torneio** e **sorteio proporcional à função de *fitness***. Após calcularmos a *fitness* de todos os indivíduos identificamos o indivíduo melhor adaptado, com o maior valor de *fitness* e executamos a reprodução desse indivíduo para a nova geração, o que é conhecido como **elitismo**. Após isso, criamos aleatoriamente a quantidade de novos indivíduos definida para a nova geração. Para complementar a quantidade de indivíduos devemos selecionar os indivíduos da geração anterior e executar operações de reprodução ou *crossover* para criar novos indivíduos com a herança genética da geração anterior.

Inicialmente determinamos qual operação iremos realizar, reprodução ou *crossover*. Para a reprodução devemos selecionar um indivíduo e gerar um novo indivíduo. No *crossover* devemos selecionar dois indivíduos e gerar dois novos indivíduos. Se temos apenas um indivíduo faltando para completar a nova geração devemos optar pela reprodução, caso contrário, sorteamos com base nas

probabilidades definidas qual será a operação realizada.

Escolhida a operação genética devemos fazer a seleção. Inicialmente sorteamos proporcionalmente ao valor da função de *fitness* um indivíduo. Vamos considerar um exemplo com 4 indivíduos, A, B, C e D com valores respectivos de *fitness* de 8, 4, 2 e 2. Nesse cenário o indivíduo A teria 50% de probabilidade de ser sorteado, B teria 25%, enquanto C e D teriam 12,5% cada. Após determinar o primeiro indivíduo, sorteamos novamente de forma proporcional um segundo indivíduo, diferente do primeiro. Executamos então um **torneio** entre os dois, sendo o vencedor aquele que possuir o maior valor de *fitness*. A quantidade de indivíduos no torneio pode ser configurada, caso seja maior do que dois um novo indivíduo é sorteado e disputa com o vencedor do torneio anterior até que a quantidade de indivíduos definida tenha se enfrentado. O campeão do torneio é selecionado para a operação genética. No caso de *crossover* o processo se repete para selecionar o segundo indivíduo. Caso os indivíduos selecionados para um *crossover* sejam iguais, são realizadas duas reproduções desse indivíduo ao invés do *crossover*.

Durante a geração de novos indivíduos ou *crossover*, realizamos um sorteio de acordo com a probabilidade de mutação, caso seja definido que uma mutação deve ocorrer, um nó é aleatoriamente sorteado e a mutação pontual é executada no conteúdo daquele nó de acordo com a definição da seção 6.5.2.

## 6.6 Ciclo de evolução

A nossa abordagem sugere que a população de soluções de uma geração seja executada em uma amostra de requisições no ambiente real. Essa abordagem difere do modelo tradicional que utiliza uma coleta de dados para treinamento, a execução e avaliação de diversas gerações sobre esses dados, até selecionar uma solução satisfatória. De forma contrária, nós sugerimos um processo interativo.

Inicialmente é necessário definir uma parcela dos dados para teste e evolução. No ambiente de publicidade computacional, devemos definir uma parcela das requisições de anúncios. Essa parcela pode ser definida como uma amostra aleatória de requisições, contextos ou usuários. Acreditamos que ao separar os contextos perderíamos informação relevante. Separar usuários aleatoriamente é uma abordagem tradicional para testes A/B, muito comuns para testes de interação com usuário ou marketing, quando um usuário tem uma alternativa fixada à sua sessão de navegação. A separação por requisição é muito mais aleatória, inclusive para uma mesma sessão do usuário, essa aleatoriedade na publicidade pode ser um fator que altere o comportamento do usuário. Dessa forma,

consideramos como melhor alternativa a separação de uma amostra de usuários.

O tamanho da amostra deve ser significativo, mas devemos considerar que se trata do ambiente real e interesses comerciais podem ser afetados se uma grande parte do tráfego estiver direcionada para soluções com baixo desempenho nos indicadores. A maior parte das requisições deve utilizar a função de referência, assim como o grupo de controle.

Como exemplo, vamos supor que queremos testar uma população de 10 indivíduos. Podemos considerar uma amostra de 20% dos usuários que farão requisições de anúncios. Os demais 80% continuam utilizando a função de referência. Assim, com a distribuição uniforme, cada um dos 10 indivíduos irá atender as requisições de 2% dos usuários.

Como segundo passo, definimos um intervalo de tempo para o teste, um dia, uma semana, um mês, etc. Durante esse período acompanharemos indicadores de quantidade de impressão, quantidade de cliques e receita para cada um dos indivíduos e da função de referência.

Transcorrido o intervalo, calculamos o valor da função objetivo para cada indivíduo, *fitness*, aplicamos a seleção e os indivíduos mais adaptados produzirão uma nova geração por meio dos operadores genéticos. A nova geração será então executada durante um novo intervalo de teste. Quando um indivíduo for considerado bom o bastante, a sua função poderá ser promovida à função de referência.

Além das vantagens e desvantagens inerentes da programação genética, essa abordagem tem as suas próprias vantagens e desvantagens. Inicialmente, percebemos que obter uma boa solução pode demorar, uma vez que a maior parte dos indivíduos gerados não é boa, sendo necessárias muitas gerações e com cada geração executando durante um período longo de tempo antes de ser avaliada. Outro aspecto é que não podemos utilizar um número muito grande de indivíduos devido à restrição do tamanho da amostra ou não obteríamos uma avaliação significativa. Por outro lado, alterações no ambiente decorrentes de fatores externos são absorvidas pela evolução assim como a quantificação e avaliação do resultado é feita observando a função inserida no seu real ambiente, para anúncios, contextos e usuários reais.

## 6.7 Justificativa da implementação

Os publicadores são os grandes interessados na melhoria dos sistemas de publicidade computacional. Normalmente, eles também controlam os algoritmos de seleção e ordenação de anúncios. Ao utilizar apenas a ordenação por aposta o publicador pode maximizar a sua receita imediata, mas os anúncios exibidos tendem a perder qualidade e deixar de interessar o usuário. É importante

ressaltar que o interesse do usuário é fundamental, pois a receita só é obtida quando o usuário clica no anúncio e, para isso, o anúncio exibido deve ser relevante para o usuário. Além disso, anúncios demais ou de pouco interesse podem atrapalhar ou frustrar o usuário na sua navegação, o que pode fazer com que o usuário deixe o site do publicador não apenas não gerando receita mas também reduzindo a audiência, o que seria o pior cenário para o publicador.

Historicamente, os publicadores perceberam que esse é um problema complexo. Tentar aumentar diretamente a receita modificando o sistema pode até funcionar para aumentar a utilidade do publicador no curtíssimo prazo, mas o equilíbrio acaba sendo perdido, pois a utilidade do anunciante pode reduzir a ponto de ele deixar o sistema ou a utilidade do usuário reduzir, causando o abandono do conteúdo do publicador pelo usuário. A perda de anunciantes e audiência é considerada um sério problema para os publicadores. Sendo esse equilíbrio delicado, difícil de ser mapeado e que varia muito com fatores externos e com o tempo, percebemos a necessidade não apenas de encontrar uma solução, mas de encontrar uma nova solução a cada modificação desses fatores.

Dessa forma, o risco de utilizar o ambiente real para a evolução pode ser absorvido pelos publicadores com o benefício de se adequar mais rapidamente a mudanças e não mais interferir arbitrariamente no sistema, reduzindo muito a chance de perder anunciantes ou usuários devido a alterações arbitrárias na ordenação de anúncios.

## 6.8 Limites e discussões

Os principais limites da abordagem evolutiva e sua implementação já foram comentados. O primeiro é relacionado ao risco de utilizar o ambiente real que, como descrevemos, também se relaciona diretamente as principais vantagens da nossa abordagem. O segundo é o longo tempo de evolução. A utilização do ambiente real implica na restrição do tamanho da amostra e consequente restrição na quantidade de indivíduos, essa restrição associada ao modelo cíclico da abordagem evolutiva nos fez optar pela exploração e avaliação de cada geração no período de um dia. Isso associado a característica de lenta convergência da técnica de programação genética faz com que sejam necessários longos períodos de evolução. Além desses dois principais limites, que continuarão sendo abordados, existem outros limites que podem ser avaliados para a evolução deste trabalho. Esses outros limites são discutidos nas próximas seções.

### 6.8.1 Múltiplos objetivos

A utilização de uma função objetivo para agregar as utilidades dos diferentes participantes, como descrito na seção 1.2, pode ser reavaliada sob a ótica de múltiplos objetivos, explicitando que a criação da função objetivo é uma forma de combinar diferentes objetivos em um único valor escalar. Cada um dos indicadores, CTR, ROI e eCPM, poderiam ser tratados como objetivos separados. A abordagem de produzir uma função agregada permite controlar mais facilmente a influência de cada um dos objetivos na solução. Por outro lado, podemos criticar a função objetivo por ser arbitrária, de difícil parametrização e não ser uma função dos valores absolutos dos objetivos, mas apenas uma função de comparação.

Acreditamos que os resultados obtidos, mesmo com essa simplificação, serão significativos. Poderíamos modelar nossa avaliação para múltiplos objetivos, considerando assim os valores absolutos dos indicadores. Ao utilizar um modelo de múltiplos objetivos devemos nos preocupar com o fenômeno *bloat* [PLM08], quando os tamanhos dos indivíduos cresce rapidamente sem obter melhorias significativas de *fitness*, dessa forma, um objetivo adicional do modelo seria manter as soluções suficientemente pequenas. Ao utilizar múltiplos objetivos, deixamos de ter uma ordem total, anteriormente fornecida pela função objetivo e passamos a ter apenas uma ordem parcial que pode utilizar, por exemplo, a noção de dominância de *Pareto* [PLM08]. Vamos considerar dois indivíduos A e B. O indivíduo A tem CTR maior do que B e eCPM menor do que B. Dessa forma, A domina B com relação a CTR, mas B domina A com relação a eCPM. Nesse caso, o objetivo do algoritmo de busca de soluções passa a ser identificar o conjunto de soluções que não são dominadas por nenhuma outra. Uma ordenação parcial pode considerar para cada solução quantas outras soluções a dominam, assim a solução menos dominada seria a melhor e a solução mais dominada seria a pior, utilizando o que é conhecido como seleção de *Pareto*.

Essa alternativa utiliza normalmente um maior número de indivíduos e um maior número de interações. Gostaríamos de evoluir nossa abordagem para trabalhar diretamente com múltiplos objetivos. Porém, devido às restrições no tamanho da amostra, que implica em restrições no tamanho da população, e ao longo intervalo de tempo entre as gerações, optamos inicialmente pela agregação dos objetivos por meio de uma única função objetivo, descrita na seção 1.2.

### 6.8.2 Valor escalar por dimensão

Outro aspecto que devemos avaliar de forma crítica é a simplificação de utilizar para cada dimensão um valor escalar. Como já comentamos, essa simplificação permite modularizar a nossa

abordagem. Vamos considerar, por exemplo, a solução apresentada em [LCG<sup>+</sup>06]. Essa solução produz uma ordenação considerando e avaliando profundamente características de relevância e similaridade entre o anúncio e o contexto, ou mais especificamente, a página. Pelos resultados apresentados no trabalho podemos afirmar que essa solução seria melhor do que a similaridade de cossenos dos vetores(*tf-idf*) utilizada na nossa abordagem. A modularização, isolando a dimensão de contexto como um escalar, permite substituir a similaridade de cossenos por essa solução.

Por outro lado, soluções como a descrita em [LCG<sup>+</sup>06] consideram diversas características dos anúncios, contextos e usuários. É possível, e muito provável, que alguma ou várias dessas características possam ser diretamente expostas e utilizadas pela segunda fase de ordenação. O mesmo também pode ser dito das características de todas as outras dimensões. Nesse cenário, a segunda fase teria à sua disposição para reordenação, não apenas um valor escalar de cada dimensão, mas os múltiplos valores de todas as características relevantes de todas as dimensões. Nessa alternativa, a função de ordenação obtida pode superar os resultados de funções evoluídas pela abordagem descrita. Entretanto, a modularização seria mais difícil e o motor de evolução teria que trabalhar com um número muito maior de variáveis, precisando criar funções muito mais complexas, o que potencialmente exigiria uma população maior e um maior número de gerações para evoluir. Mais uma vez, devido às restrições no tamanho da amostra e ao longo intervalo de tempo entre as gerações, optamos pela abordagem inicial que nos traz a vantagem adicional de melhor modularização.

### 6.8.3 O valor do clique e a dimensão de valor comercial

Inicialmente optamos por fixar a dimensão de valor comercial na ordenação como primeiro fator de nossas funções, como na fórmula de ECPM, equação 2.7. Isso permitirá mantermos a precificação pelo leilão de segundo preço e o cálculo do valor do clique pela fórmula 2.8. É importante ressaltar que mesmo fixando esse fator na fórmula de ordenação, o valor do CPC irá variar de acordo com o resultado da ordenação.

Acreditamos que fixando parte da estrutura para garantir o modelo de leilão de segundo preço, concentramos nossa atenção na ordenação. Assim a estrutura da fórmula de ordenação será:

$$CPCbid_a \times g(\dots) \tag{6.2}$$

na qual a função  $g$  não utilizará as variáveis da dimensão de valor comercial ( $CPCbid_a$ ,  $CPCbid_{avg}$ ,  $CPCbid_{max}$  e  $CPCbid_{min}$ ).

## 6.9 Conclusão

Como parte deste trabalho desenvolvemos os componentes de avaliação, **Avaliador**, e evolução, **EvolutionEngine**, que utilizam o **ReConf** para alterar o comportamento do **AdServer** ao selecionar e ordenar anúncios. O único desses componentes que já existia no sistema antes deste trabalho é o **AdServer**, que modificamos o funcionamento para adaptar seu comportamento de acordo com a modificação de valores das configurações do sistema.

Nossa abordagem foi construída para endereçar diretamente o problema de ordenação de anúncios em suas diversas dimensões e utilidades. O modelo estático tradicional de evolução sobre dados coletados poderia alcançar resultados interessantes. Porém, o clique é uma interação com o usuário e, como discutimos no início do texto, descobrir a probabilidade de um anúncio ser clicado é um dos grandes desafios da área. O clique é uma decisão que só é tomada pelo usuário após os anúncios serem exibidos. Por causa disso, qualquer fluxo de clique que coletemos só representa a probabilidade para a função de ordenação que gerou aquelas impressões, não podendo ser utilizado para avaliar uma outra função. Assim, para resolver o dilema, optamos por uma abordagem interativa e diretamente aplicada ao sistema real, de geração à geração.

# Capítulo 7

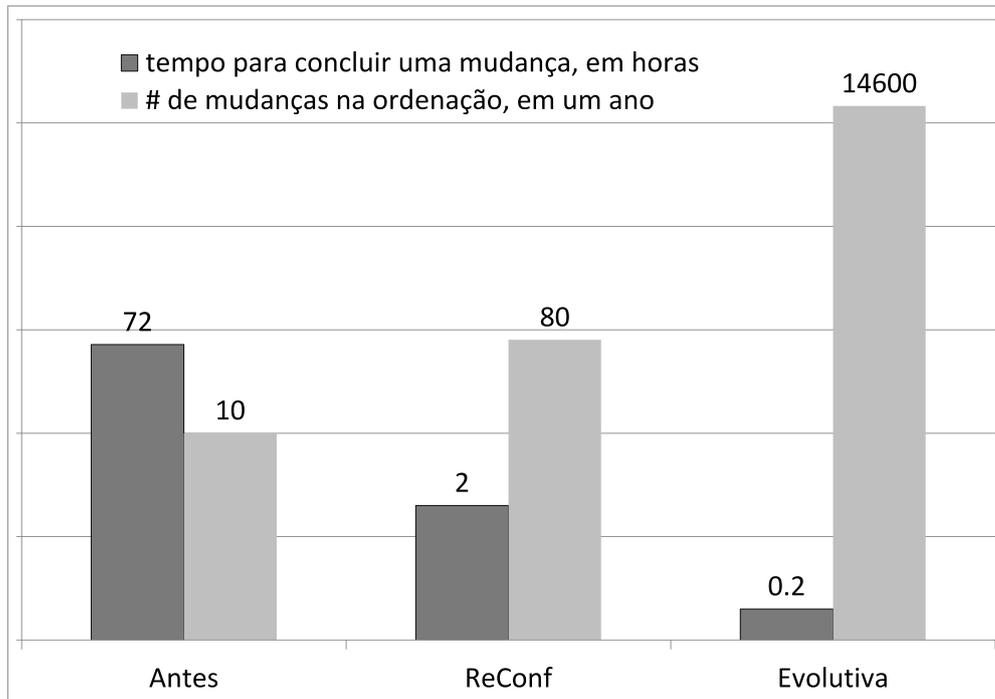
## Resultados

Neste capítulo reunimos todos os resultados obtidos por este trabalho, indicamos as publicações e apresentamos mais detalhes sobre os experimentos realizados.

### 7.1 Mudanças e adaptação

Neste trabalho desenvolvemos e disponibilizamos o **ReConf**, um repositório centralizado de configurações desenvolvido como uma API de serviço web RESTful, como descrevemos na seção 5.5. O **ReConf** é parte central da arquitetura adaptativa do sistema de publicidade computacional SIGA e desempenha um papel fundamental na implementação da abordagem evolutiva, gerenciando as configurações que definem os experimentos. Apresentamos na seção 5.6 como o **ReConf** permitiu resolver problemas, auxiliando a responder melhor e mais rapidamente a mudanças necessárias. Mudanças que se tornam necessárias para atender às demandas de negócio e para a execução de experimentos controlados. Na Figura 7.1, resumimos os resultados alcançados pelo **ReConf** que foram publicados na ICWS 2014, no artigo [BMF14].

Antes da adoção do **ReConf**, rotulado como *Antes*, apenas 10 mudanças na ordenação de anúncios eram propostas e avaliadas em um ano. Qualquer mudança demorava no mínimo 3 dias, ou 72 horas, para ser disponibilizada no ambiente de publicidade computacional pelo sistema. Após a implementação do **ReConf** e as refatorações nos componentes e serviços, coluna *ReConf* na Figura 7.1, a maioria das mudanças ainda eram originadas pelos patrocinadores, *stakeholders*, mas elas passaram a ser disponibilizadas em um tempo bem menor, na verdade, em poucas horas, em média 2 horas. Além disso, os patrocinadores puderam testar diversas hipóteses para a ordenação de anúncios com menor risco e maior confiança na avaliação dos resultados por meio dos experimentos controlados, alcançando 80 mudanças no período de um ano. Por fim, a utilização do **ReConf**



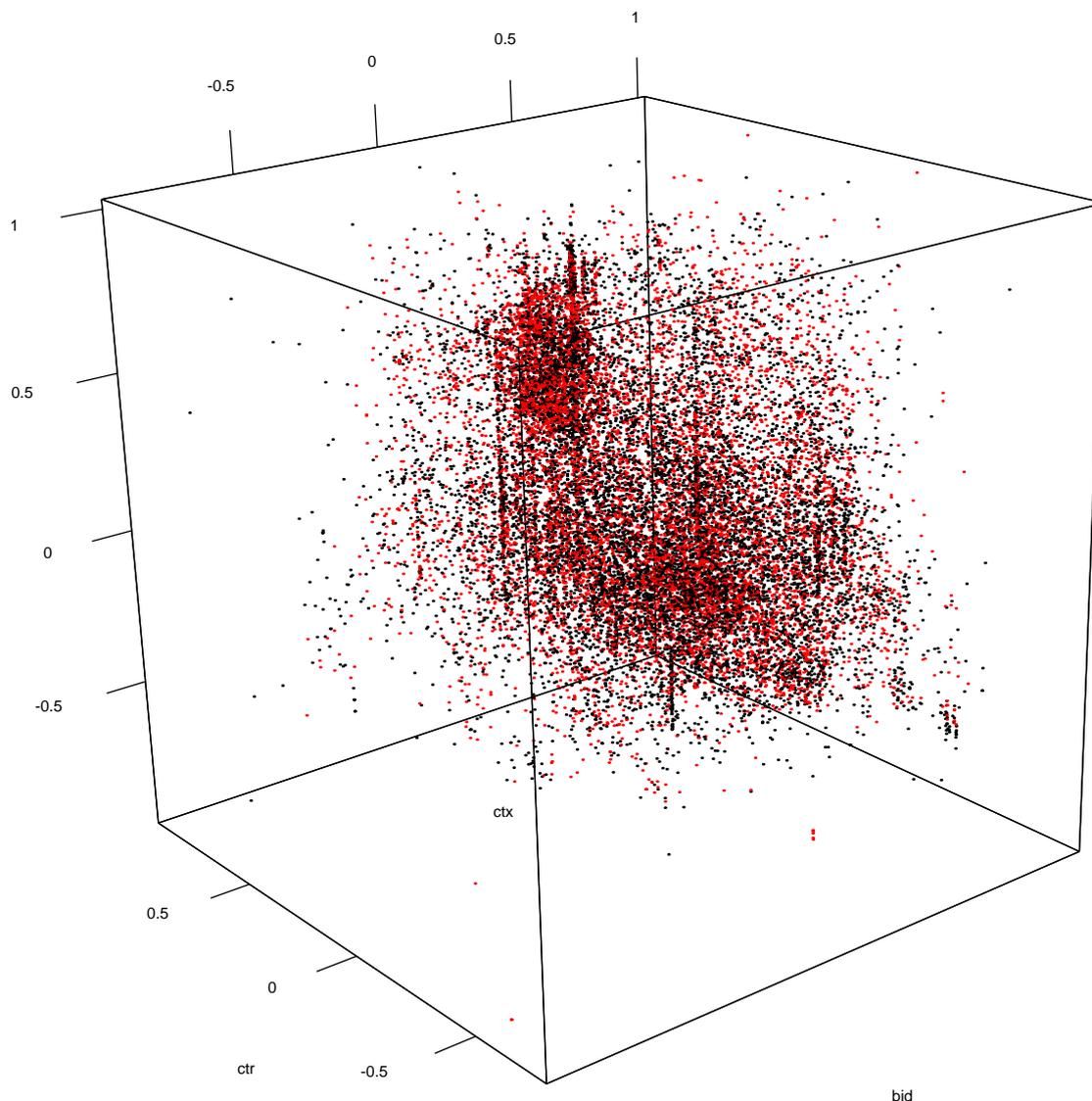
**Figura 7.1:** Resultados da utilização do **ReConf** (escala logarítmica)

para a implementação da abordagem evolutiva, coluna *Evolutiva*, nos levou a explorar diariamente dezenas de mudanças na ordenação de anúncios, atingindo números muito maiores de mais de 14.600 mudanças no período de uma ano. Devido a API de serviço web e a biblioteca cliente, uma mudança se torna disponível no sistema em apenas alguns minutos após ela ter sido elaborada pelo motor de evolução, em média menos de 15 minutos.

## 7.2 Análise dos dados

Inicialmente conduzimos uma análise exploratória dos dados. Coletamos eventos de exibição e clique dos anúncios. Associado a cada evento estão registrados os valores de  $BID_a$ ,  $CTR_a$ ,  $CTX_{a,c}$  e  $USR_{a,u}$  da exibição do anúncio  $a$ , no contexto  $c$  ao usuário  $u$ . Nossa intenção inicial era executar modelos de regressão sobre esses dados. De acordo com nossa formulação do problema avaliamos a possibilidade de que, por exemplo, com base na combinação linear dos valores das dimensões fosse possível classificar se o anúncio seria ou não clicado ou então estimar a probabilidade de clique. A execução direta de regressões se mostrou insatisfatória. Durante nossa investigação utilizamos visualizações dos dados no espaço das dimensões para nos auxiliar na exploração. Uma dessas visualizações é exibida na Figura 7.2.

Na Figura 7.2 exibimos pontos vermelhos indicando anúncios que receberam cliques e pontos pretos para anúncios que não receberam cliques. As três primeiras dimensões normalizadas formam



**Figura 7.2:** Representação dos eventos avaliados: anúncios clicados são exibidos em vermelho nas coordenadas que representam seus valores normalizados de  $BID_a$ ,  $CTR_a$  e  $CTX_{a,c}$ . Anúncios não clicados são representados por pontos pretos.

a base do espaço de visualização. Nessa visualização consideramos apenas dados de requisições de anúncio que retornaram ao menos dois anúncios e, na qual, um dos anúncios retornados recebeu um clique. Devido as taxas de clique extremamente baixas, a utilização de todos os dados tornaria impossível a visualização de anúncios que receberam cliques, ou seja, o espaço de visualização seria dominado por pontos pretos. Por meio dessa visualização restrita, é possível observar que pontos muito próximos ou idênticos possuem eventos de clique e não clique. Em publicidade computacional, isso significa que anúncios com valores equivalentes nas três dimensões foram exibidos e algumas vezes receberam cliques e outras não. Dessa forma, foi possível entender a razão da modelagem

simples do problema e aplicação direta de técnicas de regressão gerar modelos que previam ou estimavam a ação de clique com 50% de probabilidade ou confiança, o equivalente a aleatoriedade ao se considerar que existem apenas duas opções. Acreditamos que as técnicas de regressão possam ser aplicadas com a adequada reformulação do problema.

Recuperando o problema original, compreendemos que não é necessário classificarmos ou estimarmos diretamente a probabilidade de clique. Nosso objetivo é encontrar uma ordenação, obviamente a probabilidade de clique poderia induzir tal ordem, mas não é a única forma de se obter uma ordenação. Utilizando as funções históricas de ordenação, a nossa primeira hipótese é que poderíamos criar uma abordagem iterativa para encontrar os valores dos coeficientes que otimizariam uma função de ordenação composta pela combinação linear das dimensões de desempenho histórico, contexto e usuário com a multiplicação pelo valor comercial. Poderíamos utilizar a abordagem cíclica de exploração, avaliação e evolução para experimentar diferentes valores dos coeficientes. Durante a concepção, percebemos uma desvantagem: que isso fixaria o formato da combinação das dimensões como uma combinação linear. Em contrapartida, perderíamos a possibilidade de encontrar uma possível solução com formato não linear. Dessa forma, optamos por uma técnica capaz de executar a regressão simbólica na estrutura da função, a programação genética.

### 7.3 Experimentos com dimensões

Como já discutimos, uma parte importante da nossa hipótese é encontrar uma solução que combine as diferentes dimensões do problema de ordenação de anúncios em publicidade computacional:

- **valor comercial** - medida do potencial de geração de receita do anúncio, representada no experimento pelo valor da aposta, valor monetário maior do que 1 centavo, sem limite superior;
- **desempenho histórico** - medida do resultado do anúncio quando exibido anteriormente, representada no experimento pelo CTR global do anúncio no sistema, valor entre 0 e 1;
- **contexto** - medida de quanto o anúncio é relevante para o contexto, representada no experimento pela relevância ou similaridade entre o anúncio e contexto (página, consulta, etc) representado como um valor entre 0 e 1;
- **usuário** - medida de quanto o anúncio é recomendado para o usuário, uma medida de relevância que pode utilizar características demográficas ou classificação de usuários para recomendar anúncios com características ou comportamentos semelhantes, representada no experimento como um valor entre 0 e 1.

Apresentamos em detalhes, no capítulo 4, como o sistema de publicidade computacional obtém esses valores e executa o processo de seleção e ordenação de anúncios. Nosso ponto de partida é que a combinação de todas as dimensões mencionadas é capaz de apresentar desempenho superior que a utilização de apenas uma das dimensões individualmente. Para mensurar o impacto de cada dimensão no problema e verificar se a nosso pressuposto estava correta nós conduzimos uma análise dessas dimensões aplicando os passos de experimentação e avaliação descritos no modelo cíclico de nossa abordagem evolutiva, como apresentado no capítulo anterior.

Nossa análise inclui duas rodadas de experimentação comparando diferentes soluções de ordenação de anúncios, também referenciadas como estratégias de ordenação de anúncios, ou *ad ranking strategies* (*st*). Na primeira rodada, nós analisamos cinco soluções diferentes: *stRND*, *stBID*, *stCTR*, *stCTX* e *stBTU*. A solução *stRND* é uma solução de ordenação aleatória, *random*, utilizada como base de comparação para as demais. Cada uma das demais soluções ordena os anúncios utilizando apenas uma dimensão do problema. Valor comercial é utilizado na solução *stBID*, representado pelo valor da aposta, *bid*. Desempenho histórico é utilizado na solução *stCTR*, representado pelo CTR do anúncio no sistema. Contexto é utilizado na solução *stCTX*, representado pela similaridade de contexto. Por fim, usuário é utilizado na solução *stBTU*, representado pela similaridade com comportamento do usuário, *behavior targeting the user*.

Na segunda rodada de experimentação, de maneira equivalente, nós avaliamos as seguintes soluções: *stCTR*, *stBID*, *stTRAD*, *stUSER* e *stMKT*. As duas primeiras são as mesmas utilizadas na primeira rodada. A solução *stTRAD* é uma solução tradicional baseada na fórmula de ordenação utilizada pelo Google em [SK11], a multiplicação direta do CTR pelo valor da aposta, *bid*. A solução *stUSER* é a soma das dimensões que podem ser mais fortemente relacionadas ao usuário: desempenho histórico, contexto e usuário, descartando o valor comercial. Por fim, a solução *stMKT* é uma solução comercial utilizada pelo sistema de publicidade computacional utilizado neste projeto para ordenar anúncios combinando as quatro dimensões do problema.

Cada rodada do experimento executou as soluções por uma semana no sistema de publicidade computacional, aplicando cada solução a uma amostra de mesmo volume de requisições de anúncios. Mais de 600 milhões de anúncios foram exibidos durante esse experimento. Para avaliar os resultados nós coletamos dados para calcular o CTR e o eCPM do sistema durante o período. Além disso, conduzimos uma análise da correlação entre as dimensões por meio da técnica de PCA. Apresentamos os principais resultados alcançados na sequência, resultados que foram publicados no SBBD 2014, no artigo [CBMF14]. Esse artigo recebeu o prêmio de melhor artigo da conferência. Em resumo as soluções avaliadas foram:

- *stRND* - solução de ordenação aleatória, *baseline* de comparação;
- *stBID* - solução de ordenação pelo valor da aposta, *bid*, considerando apenas a dimensão de valor comercial;
- *stCTR* - solução de ordenação pela taxa de clique do anúncio no sistema, CTR, considerando apenas a dimensão de desempenho histórico;
- *stCTX* - solução de ordenação pela similaridade de contexto, considerando apenas a dimensão de contexto;
- *stBTU* - solução de ordenação pela similaridade dos interesses do usuário, considerando apenas a dimensão de usuário;
- *stTRAD* - solução de ordenação tradicional baseada na fórmula de ordenação apresentada pelo Google em [SK11], multiplicação do CTR pelo valor da aposta, *bid*, considerando apenas as dimensões de valor comercial e desempenho histórico;
- *stUSER* - solução de ordenação pela soma das dimensões mais fortemente relacionadas ao usuário: desempenho histórico, contexto e usuário, descartando valor comercial;
- *stMKT* - solução de ordenação comercial utilizada pelo SIGA no período desse experimento, combinando as quatro dimensões, *baseline* forte de comparação.

A Figura 7.3 apresenta os valores normalizados de CTR e eCPM para cada solução durante a primeira rodada de experimentação. A solução com melhor desempenho no CTR foi *stCTR*, enquanto a solução com melhor eCPM foi *stBID*, um comportamento que avaliamos como normal e esperado. As altas taxas de clique apresentadas pela solução *stCTR* possibilitou a essa solução alcançar melhores valores de eCPM, ou seja, melhor desempenho comercial, que a solução de comparação *stRND*. Por outro lado, *stBID* superou consideravelmente todas as outras soluções no valor de eCPM, mesmo possuindo o valor mais baixo de CTR. Enquanto *stCTX* exibiu resultados levemente superiores a *stRND* em ambas as métricas, não podemos dizer o mesmo de *stBTU*, que apresentou comportamento muito similar a base de comparação aleatória.

De forma análoga, a Figura 7.4 apresenta os valores normalizados de CTR e eCPM para cada solução durante a segunda rodada de experimentação. A solução *stUSER* apresentou resultados levemente piores que *stCTR*, em contrapartida, *stTRAD* foi a solução mais equilibrada em ambas as métricas. A solução com melhor valor de eCPM, *stMKT*, superou a segunda melhor solução,

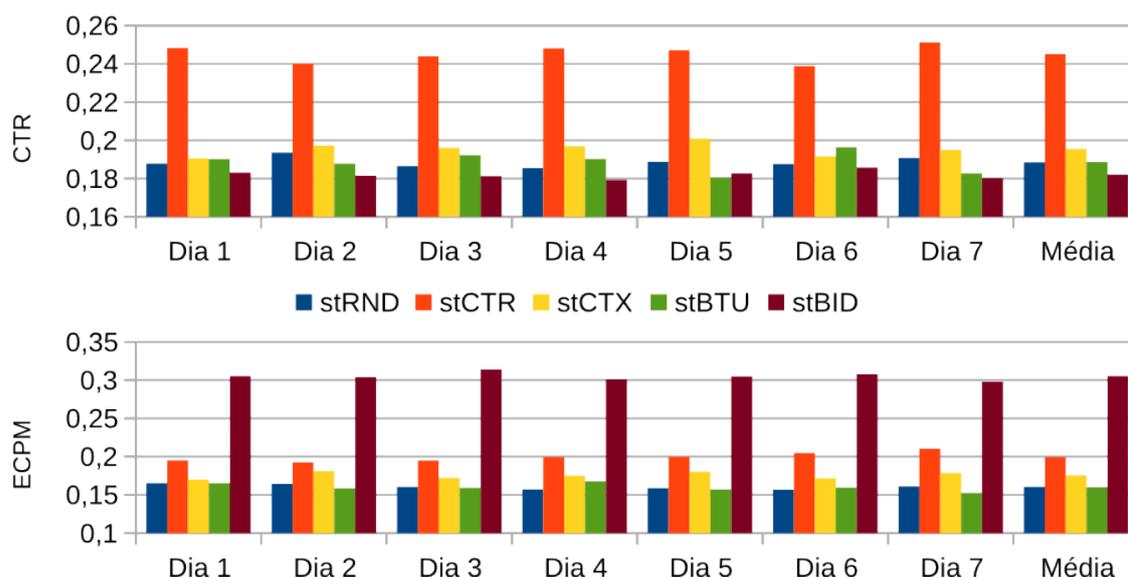


Figura 7.3: *Primeira rodada* - valores normalizados de CTR e eCPM

*stBID*, não apenas no desempenho comercial, como também na capacidade de atrair o interesse dos usuários, alcançando assim maiores taxas de clique que *stBID*.

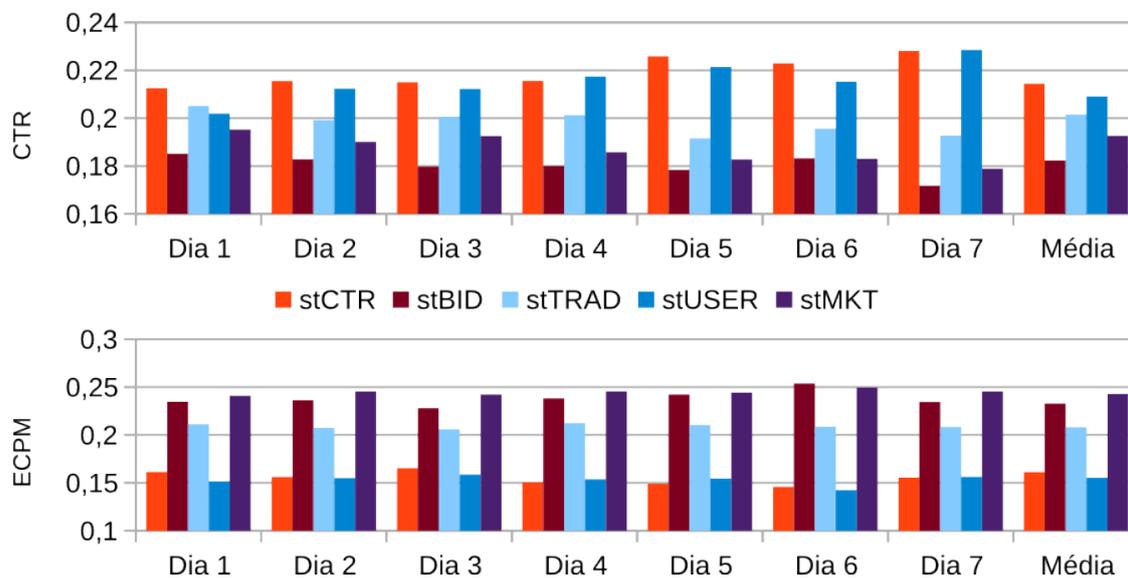
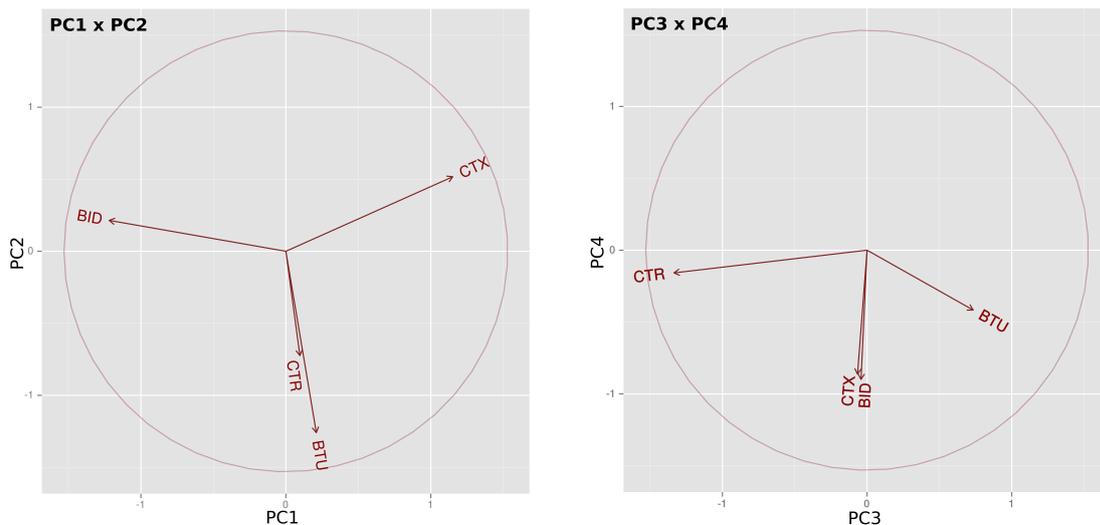


Figura 7.4: *Segunda rodada* - valores normalizados de CTR e eCPM

Esses resultados nos permitem afirmar que a combinação das diferentes dimensões pode simultaneamente melhorar o desempenho geral do sistema de publicidade computacional para os usuários, anunciantes e publicadores. Além disso, nós executamos uma análise da correlação entre as dimensões, como descrito na seção 2.6, aplicando a técnica de análise dos componentes principais, PCA, em amostras de dados das duas rodadas de experimentação.

Para a análise dos componentes principais nós utilizamos apenas dados de requisições de anúncio nas quais algum dos anúncios retornados recebeu clique de um usuário. Isso foi necessário pois o volume de cliques é muito baixo, quando utilizado todo o conjunto de dados as interações de usuário são pouco significativas, não permitindo que a análise seja executada adequadamente. A Figura 7.5 apresenta dois mapas de variáveis (*variable factor maps*), nos quais as dimensões originais do problema representadas como BID, CTR, CTX e BTU, são exibidas como vetores em planos cujas bases são definidas por vetores dos componentes principais identificados pela técnica de PCA: PC1, PC2, PC3 e PC4. No plano PC1 x PC2, podemos observar que BID, CTX e BTU são bem representadas pelas duas primeiras componentes principais PC1 e PC2, uma vez que sua magnitude é quase igual a um. A diferença entre suas direções mostra que elas possuem baixa correlação. Mesmo com CTR possuindo baixa magnitude nesse plano, seu alinhamento com BTU pode fornecer indícios que recuperar e ordenar anúncios de acordo com a dimensão de usuário pode atrair mais cliques para o anúncio.



**Figura 7.5:** Dimensões projetadas nos planos derivados da análise de componentes principais, PCA.

As componentes PC1 e PC2 foram responsáveis por apenas 56,3% de toda a variância dos dados. Portanto, nós devemos também considerar as componentes PC3 e PC4 para descrever completamente a variância dos dados. No plano PC3 x PC4, podemos observar que PC3 é composta sobretudo por CTR. Isto, assim como a diferença nas direções de BID, CTX e BTU no primeiro plano, nos permite concluir que essas dimensões representam diferentes e complementares características do problema. A última componente, PC4, é composta de informação residual das variáveis, como é comum nesse tipo de análise.

Por meio da técnica de PCA, nós descobrimos que existe muito baixa correlação entre essas dimensões. O principal resultado dessa análise é que fomos capazes de confirmar que as dimensões se complementam e nenhuma delas deve ser negligenciada ao se buscar uma solução para o problema de ordenação de anúncios em publicidade computacional. Dessa forma, esse experimento mostrou a importância para o desempenho comercial dessas dimensões do problema de publicidade computacional, permitindo afirmar que a combinação das diferentes dimensões pode simultaneamente melhorar o desempenho geral do sistema de publicidade computacional para os usuários, anunciantes e publicadores.

## 7.4 Experimento 1 - avaliação da abordagem evolutiva

Para avaliar nossa hipótese e colocar em prática nossa abordagem, executamos um experimento simplificando alguns detalhes e apresentaremos a análise de seus resultados. O experimento teve como meta evoluir, no sistema de publicidade computacional descrito, um conjunto de 40 soluções de ordenação de anúncios durante um período de 22 dias consecutivos. Utilizamos como referência de comparação (*baseline*) a melhor função de ordenação conhecida até o momento e utilizada no sistema para ordenar a maioria dos anúncios, *stMKT*, como descrita no experimento com as dimensões na seção 7.3.

Fizemos a opção pelo período de 22 dias pois consideramos um período adequado para a análise, não sendo muito curto e nem tão longo. Os experimentos executados permitiram observar modificações interessantes no intervalo de três semanas. O período não é de exatamente 21 dias por questões técnicas de agenda de implementação e coleta de dados, poderíamos descartar os dados do primeiro ou do último dia do experimento, mas, uma vez que já foram coletados, preferimos incluí-los nas análises.

### 7.4.1 Restrições

Na implementação atual do sistema de publicidade computacional ainda não é possível realizar amostragem por usuários. Possibilitar essa implementação envolve a aprovação de modificações do sistema que não puderam ser obtidas durante o período deste projeto, isso é uma das possíveis extensões do nosso trabalho. Dessa forma, utilizamos amostragem das requisições nos experimentos. Além disso, estamos utilizando a restrição de estrutura das soluções que fixa e limita a dimensão de valor comercial, como discutido na seção 6.8.3. Os indivíduos têm a estrutura definida em 6.2, com o primeiro nó da solução sendo o operador de multiplicação e seu filho esquerdo a variável

$CPCbid_a$ , podendo evoluir livremente a sub-árvore direita. Para permitir a evolução completa da estrutura seria necessário estabelecer uma nova forma de cálculo do valor do clique, o que exige aprovações de modificações no modelo de negócio que também não puderam ser obtidas durante o período deste projeto.

### 7.4.2 Geração inicial

Tradicionalmente, em programação genética a geração inicial é criada aleatoriamente. Cada uma das nossas gerações são compostas por 40 indivíduos. Optamos pela utilização da geração inicial incluindo soluções de ordenação que apresentam resultados satisfatórios, o que permite iniciar a evolução a partir de uma situação melhor conhecida. Dessa forma, a geração inicial foi constituída de 20 soluções aleatoriamente geradas e 20 soluções determinadas da seguinte maneira:

- **função de referência**, cópias da função de referência, exatamente idêntica ou com as sub-árvores espelhadas sempre que o operador for comutativo ( $\times$  e  $+$ );
- **função tradicional**, cópias da função de ordenação tradicional,  $stTRAD$ , seguindo as mesmas variações da função acima;
- **função normalizada**, cópias das funções acima utilizando normalizações diferentes: por log ou por média, máximo e mínimo;
- **função com constantes modificadas**, cópias das funções de ordenação acima, com diferentes valores para as constantes;
- **variáveis puras**, funções cada uma contendo apenas uma variável, apenas  $CTR_a$ ,  $CTX_{a,c}$  e  $USR_{a,u}$ , assim como as combinações de soma simples de duas dessas variáveis e das três em conjunto.

Com essa composição da geração inicial, esperamos avaliar se após algumas gerações já é possível observar um aumento da utilidade do sistema. Por outro lado, inserimos novos indivíduos aleatoriamente criados a cada nova geração, reduzindo o efeito de não ter criado toda a geração inicial de forma aleatória. Utilizar diferentes cópias das funções de referência e da função tradicional é uma estratégia para tentar aproveitar o conhecimento acumulado e acelerar o processo de evolução, sobretudo, ao considerarmos que cada geração demora um dia para ser evoluída. Esse cenário permite analisar a capacidade de evolução da abordagem, partindo de uma população que contém soluções com bons resultados conhecidos.

### 7.4.3 Parâmetros

Devido às restrições de tamanho da amostra, como discutido na seção 6.8, é necessário trabalhar com populações pequenas. Estamos utilizando 40 indivíduos e tamanho da população constante em todas as gerações. Trabalhos com programação genética aplicada a pequenas populações sugerem utilizar uma maior taxa de mutação para permitir que mudanças ocorram com mais frequência. Além disso, para aumentar a diversidade inserimos novos indivíduos gerados aleatoriamente a cada geração.

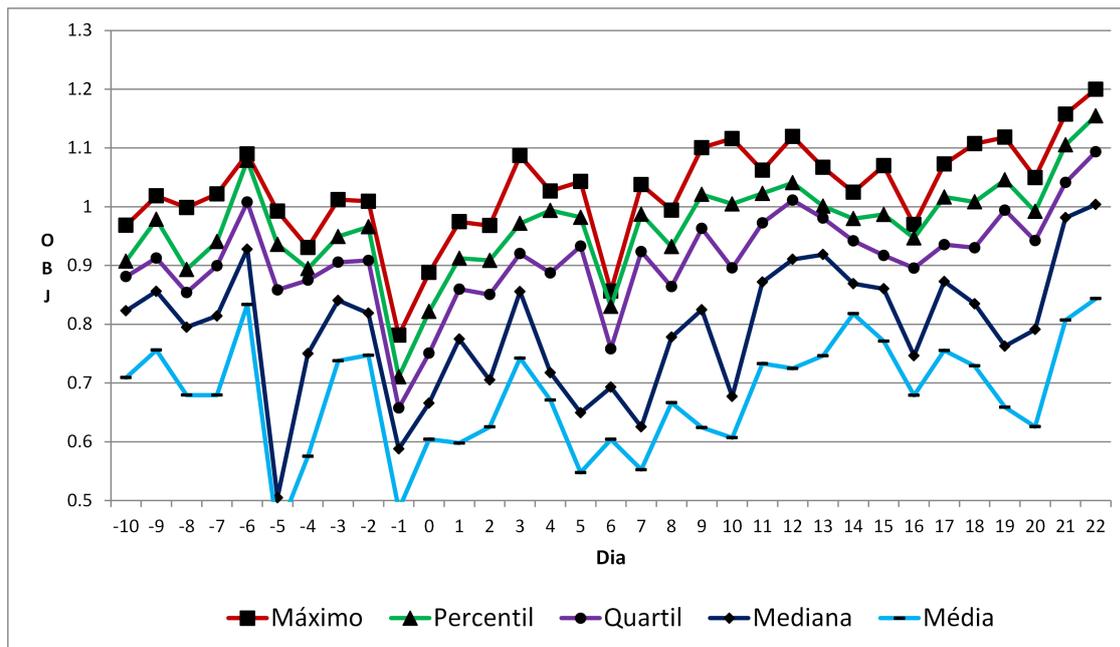
As probabilidades foram assim definidas:

- **reprodução** - 10%;
- **crossover** - 10%;
- **mutação** - 80%;
- **criação** - 15%, porcentagem da nova geração que será preenchida com novos indivíduos gerados aleatoriamente.

Como descrevemos anteriormente, no momento de criar uma nova geração, nossa abordagem utiliza inicialmente elitismo, reproduzindo o indivíduo melhor adaptado. A seguir, nossa abordagem cria aleatoriamente novos indivíduos até atingir a porcentagem definida, no caso 15%, que para uma geração com 40 indivíduos significa a criação de seis indivíduos aleatoriamente. Os outros 33 indivíduos serão gerados por reprodução, mutação ou *crossover* de acordo com as probabilidades definidas.

### 7.4.4 Resultados do experimento

Na Figura 7.6, apresentamos métricas estatísticas no valor da função objetivo para as 40 soluções de ordenação de anúncio evoluídas e avaliadas diariamente por um período de 22 dias. Durante esse período nenhuma mudança foi aplicada na solução de controle, *stMKT*. Do alto para baixo na Figura 7.6, a primeira linha representa o valor máximo obtido pela melhor solução entre as 40 soluções candidatas de cada dia. A segunda linha apresenta o percentil, ou seja, o valor da função objetivo superado por 10% das soluções candidatas. De forma análoga, a terceira linha representa o quartil, 25%, ou um quarto, das soluções candidatas; a quarta linha representa a mediana, valor superado por metade das soluções e a quinta linha representa a média do valor da função objetivo para as 40 soluções naquele dia.



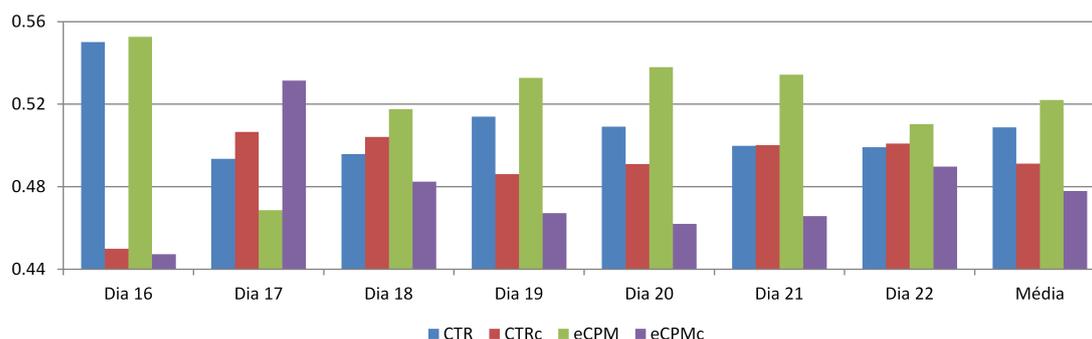
**Figura 7.6: Experimento 1** - Valores estatísticos da função objetivo calculados com base nos 40 indivíduos (máximo, percentil, quartil, mediana e média) durante os 22 dias de execução do experimento com o objetivo de avaliar a abordagem evolutiva

Nós observamos que ocorreu uma melhora geral nos valores de cada métrica do início ao final do período. A Figura 7.6 mostra, por exemplo, que ocorreu um aumento de 21% no valor do percentil. Nós aplicamos um teste de hipótese estatístico, *teste-t*, comparando o primeiro e o último dia. Nós rejeitamos a hipótese nula, concluindo que os dados desses dois dias são significativamente diferentes com um nível de confiança de 98%. Nós também aplicamos testes adicionais, comparando o último dia com cada um dos cinco primeiros dias e, ao contrário, comparando o primeiro dia com cada um dos últimos cinco dias. Sete desses dez testes rejeitaram a hipótese nula com nível de confiança de 90% e quatro dentre os sete com nível de confiança de 98%. Esses dados são fortes evidências da melhora alcançada pelas soluções após 22 dias de evolução.

Com base nesses dados, nós podemos afirmar que o conjunto das melhores soluções avaliado nos últimos dias são capazes de fornecer melhor desempenho em comparação com a solução de controle que o conjunto das melhores soluções disponíveis no início do experimento. Dessa forma, nós podemos concluir que a nossa abordagem evolutiva foi capaz de automaticamente evoluir soluções de ordenação de anúncios durante o período observado, criando assim soluções ainda melhores a partir das melhores soluções das gerações anteriores, validando os passos automáticos de **Avaliação** e **Evolução** da nossa abordagem.

### 7.4.5 Promovendo a melhor solução do experimento 1

No 16º dia, uma solução candidata foi promovida para a segunda fase de validação. Na Figura 7.7, nós apresentamos os indicadores de desempenho da nova solução e da solução de controle. Podemos observar valores superiores aos apresentados pela solução de controle em ambos os indicadores, respectivamente, CTR e eCPM maiores que CTRc e eCPMc. Apesar de possuir apenas sete dias de dados, nós também aplicamos testes de hipótese estatísticos, *teste-t*, comparando os indicadores da solução promovida e da solução de controle. Como era de se esperar, nós não conseguimos rejeitar a hipótese nula com relação ao CTR. Por outro lado, nós ficamos positivamente surpresos por conseguir rejeitar a hipótese nula com relação ao eCPM, com um nível de confiança de 98%. Nós não podemos afirmar que esses últimos resultados possuem significância estatística, porém nós acreditamos que comentar os resultados desses testes estatísticos é importante uma vez que executamos eles e estamos exibindo os dados observados na Figura 7.7.



**Figura 7.7: Desempenho da melhor solução encontrada no experimento 1 - Comparativo dos indicadores de desempenho normalizados, ECPM e CTR, da nova solução e da solução de controle nos 22 dias da segunda fase de avaliação da melhor solução encontrada no experimento 1**

Apesar de não ser possível publicar valores financeiros reais devido a compreensíveis restrições comerciais, nós podemos imaginar a importância de tais aumentos nos indicadores. Vamos considerar que o sistema processe 2 bilhões de requisições de anúncios por mês, vamos supor que o CTR médio do sistema seja 0.5%, ou seja, 10 milhões de cliques por mês. Vamos também supor que U\$ 0,10 é o custo por clique médio do sistema. Isso resultaria em um negócio de um milhão de dólares por mês, com um eCPM de U\$ 0,0005. Na Figura 7.7, o aumento de eCPM foi, em média, maior que 9%. Tal aumento poderia ser capaz de fornecer U\$ 90.000 adicionais por mês. Ao mesmo tempo, o sistema seria capaz de exibir anúncios mais interessantes que poderiam aumentar o número de visitas às páginas dos anunciantes, devido a um aumento médio no CTR de 3,6%.

Portanto, nós também conseguimos observar melhorias originadas pela segunda fase de validação e pelo mecanismo de promoção automático de soluções de nossa abordagem evolutiva. A execução

contínua do ciclo de exploração, avaliação e evolução, trabalhando em conjunto com a segunda fase de avaliação e o mecanismo de promoção, permite reconfigurar automaticamente a ordenação de anúncios do sistema. Além disso, possibilita que o sistema responda melhor às mudanças no ambiente de publicidade computacional, ou mesmo no mercado de publicidade online, por meio da modificação automática da sua ordenação de anúncios.

## 7.5 Experimento 2 - evolução por períodos maiores

Após os resultados do primeiro experimento, executamos um segundo experimento, utilizando um período de tempo maior. O segundo experimento teve como meta evoluir, no sistema de publicidade computacional descrito, um conjunto de 40 soluções de ordenação de anúncios durante um período de dois meses e meio consecutivos. Novamente utilizamos como referência de comparação (*baseline*) a melhor função de ordenação conhecida até o momento e utilizada no sistema para ordenar a maioria dos anúncios, *stMKT*, como descrita no experimento com as dimensões na seção 7.3.

Esse experimento teve as mesmas restrições de amostragem e estrutura do primeiro experimento, como descrevemos na seção 7.4.1. De forma análoga, utilizamos os mesmos parâmetros do primeiro experimento:

- **reprodução** - 10%;
- **crossover** - 10%;
- **mutação** - 80%;
- **criação** - 15%, porcentagem da nova geração que será preenchida com novos indivíduos gerados aleatoriamente.

As gerações iniciais foram construídas da mesma forma que no experimento 1, com 20 indivíduos gerados aleatoriamente e 20 indivíduos determinados. Apesar dos indivíduos determinados serem os mesmos para ambos os experimentos, os demais indivíduos foram gerados de forma independente para cada experimento, fornecendo assim gerações iniciais diferentes. A fim de ilustração, apresentamos exemplos de indivíduos da geração inicial do experimento 2 no Apêndice A, assim como um exemplo de uma fórmula que foi evoluída.

### 7.5.1 Resultados do experimento 2

As Figuras 7.8(a), 7.8(b) e 7.8(c) exibem os dados observados do experimento 2. A Figura 7.8(a) apresenta o gráfico de acompanhamento dos valores estatísticos da função objetivo calculados com base nos 40 indivíduos (máximo, percentil, quartil, mediana e média) durante o período de execução do experimento.

Iniciando com uma análise geral das quatro primeiras linhas da Figura 7.8(a): máximo, percentil, quartil e mediana; podemos ver que elas iniciam em valores entre 0.88 e 0.96 e finalizam em patamares mais elevados, entre 0.94 e 1.15. Essas linhas finalizam com valores mais próximos dos pontos máximos alcançados durante o período do que dos seus respectivos valores iniciais. Outro aspecto que podemos observar é a variância entre os resultados dos indivíduos. Podemos observar esse aspecto pelo distanciamento que ocorre entre as curvas, sobretudo entre a mediana e o máximo. Um padrão pode ser observado, com as curvas se afastando, mas na sequência se reaproximando, na maioria das vezes em patamares maiores de valores de mediana e quartil.

Podemos observar uma maior amplitude de variação nas Figuras 7.8(b) e 7.8(c). A Figura 7.8(b) apresenta os valores percentuais de incremento no CTR, positivo ou negativo, alcançado pelo CTR máximo, percentil, quartil, mediano ou médio, calculado com base nos 40 indivíduos em relação ao CTR da solução de controle. A Figura 7.8(c) apresenta uma visão análoga, com relação ao incremento de ECPM alcançado. De acordo com nossa interpretação, dois fatores podem ser mais relevantes nessa observação: a exploração aleatória e a variância entre os resultados dos indivíduos dentro da mesma geração. Por sua vez, a variância entre os resultados dos indivíduos é influenciada pela inclusão de 6 novos indivíduos gerados aleatoriamente a cada nova geração.

A observação da evolução em uma janela de tempo mais longa nos permite reafirmar que a nossa abordagem é capaz de evoluir soluções ao longo do tempo. Não conseguimos definir qual poderia ser um período ótimo, ou sequer médio, para alcançar soluções interessantes. Dessa forma, ressaltamos o valor da segunda parte da nossa abordagem, a promoção de soluções e modificação da solução de controle, o que permite criar o ambiente para avaliar e modificar o experimento quando for mais indicado, sem pré-definições de tempo.

Assim como no primeiro experimento, aplicamos um teste de hipótese estatístico, *test-t*, comparando o primeiro e último dia. Rejeitamos a hipótese nula, concluindo que os dados desses dois dias são significativamente diferentes com um nível de confiança de 95%.

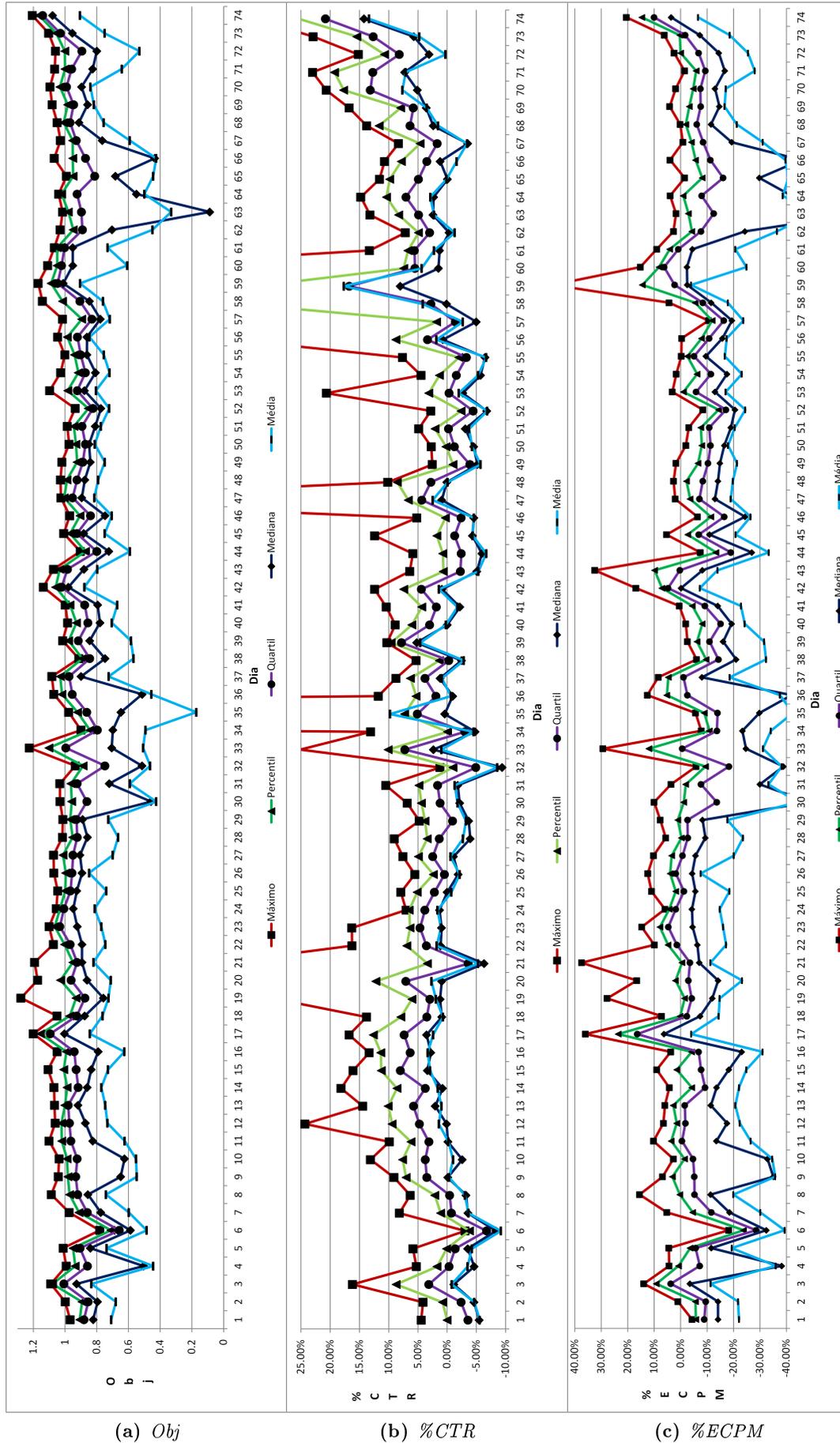
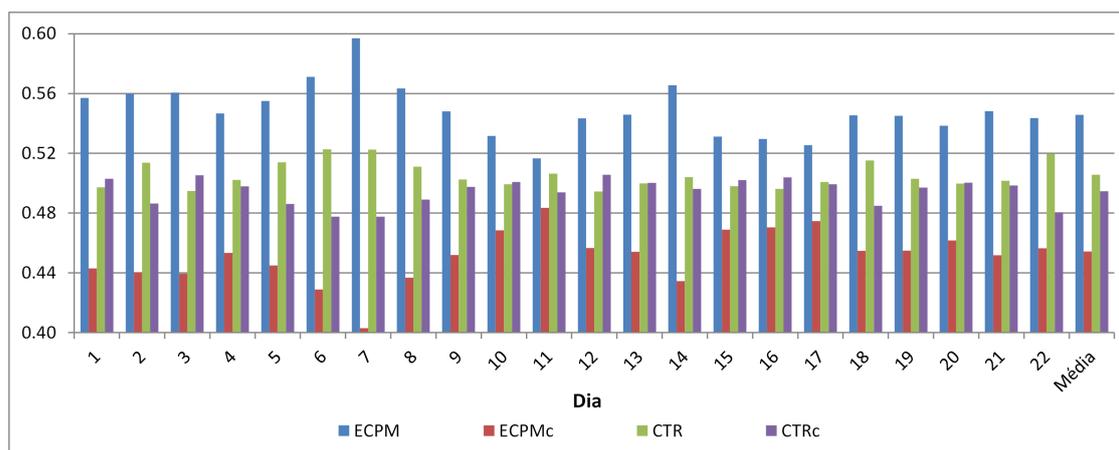


Figura 7.8: Experimento 2 - Valores estatísticos da função objetivo (a), percentual de incremento de CTR (b) e percentual de incremento de ECPM (c).

### 7.5.2 Promovendo a melhor solução do experimento 2

Após o término do experimento 2, promovemos a melhor solução encontrada nesse segundo experimento para a segunda fase de validação. Na Figura 7.9, nós apresentamos o comparativo dos indicadores de desempenho normalizados da nova solução e da solução de controle. A solução de controle é a mesma utilizada durante o experimento 2. Podemos observar, em todo o período, que o valor do indicador de ECPM da nova solução foi superior ao valor de ECPMc da solução de controle. Por outro lado, CTR e CTRc dividem a liderança, cada um sendo maior que o outro em quase metade dos dias observados. A média do período resume o resultado, apresentando valores de CTR e CTRc praticamente iguais e o valor de ECPM significativamente superior que o de ECPMc. Os testes de hipótese estatísticos, *test-t*, reforçam essa observação. Não é possível rejeitar a hipótese nula com relação a diferença de CTR e CTRc. Porém, é possível concluir que ECPM e ECPMc são distintos, com um nível de confiança de 90%.



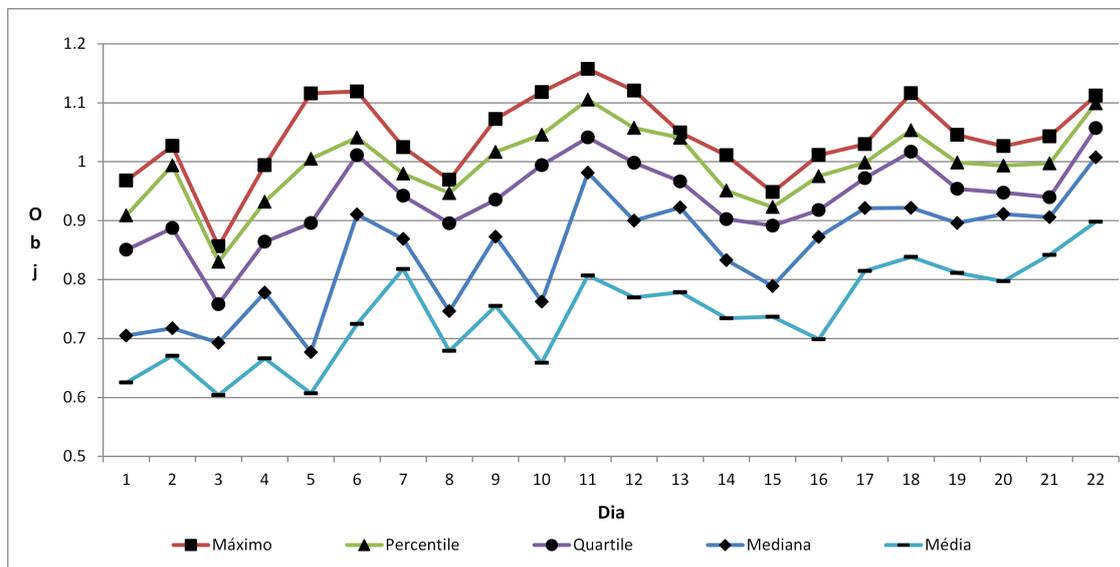
**Figura 7.9:** Desempenho da melhor solução encontrada no experimento 2 - Comparativo dos indicadores de desempenho normalizados, ECPM e CTR, da nova solução e da solução de controle nos 22 dias da segunda fase de avaliação da melhor solução encontrada no experimento 2

Esses resultados indicam que nossa abordagem é capaz de evoluir soluções e, assim, melhorar os resultados do sistema de publicidade computacional utilizado neste trabalho, inclusive em períodos mais longos. Em contraste com outras iniciativas, o objetivo da nossa abordagem não é encontrar soluções fechadas por meio da convergência de soluções, mesmo observando que durante períodos maiores é possível obter algum grau de convergência. Por fim, gostaríamos de avaliar o resultado de evoluções consecutivas dentro desse ambiente. Para isso, elaboramos um novo experimento descrito na próxima seção.

## 7.6 Experimento 3 - evoluções consecutivas

O objetivo deste terceiro experimento é avaliar o cenário de dois experimentos consecutivos. De acordo com nossa abordagem, quando encontramos uma solução que supera os limites esperados, essa solução pode ser promovida para uma segunda fase de avaliação. Uma vez que essa solução se torne bem sucedida na segunda fase de avaliação, ela poderá ser promovida à nova solução principal dos sistema e, conseqüentemente, se tornar também a solução de controle e referência para os novos experimentos.

O experimento 3 possui as mesmas restrições e parâmetros utilizados nos experimentos 1 e 2. Por outro lado, a geração inicial e solução de controle foram definidos de forma diferente. De acordo com o ciclo de evolução, podemos considerar que o experimento 3 ocorre na seqüência do experimento 2. Após o final da primeira fase de avaliação do experimento 2, que durou dois meses e meio, ocorreu a segunda fase de avaliação da melhor solução encontrada, durante um período de 22 dias. Após esse intervalo, essa solução se tornou a nova solução de referência e controle para os experimentos e o terceiro experimento começou com uma geração inicial composta por 20 novas fórmulas geradas aleatoriamente e as 20 melhores fórmulas do último dia do experimento 2.



**Figura 7.10:** Experimento 3 - Valores estatísticos da função objetivo calculados com base nos 40 indivíduos (máximo, percentil, quartil, mediana e média) durante os 22 dias de execução do experimento com o objetivo de avaliar o resultado de evoluções consecutivas

A Figura 7.10, apresenta os valores estatísticos da função objetivo calculados com base nos 40 indivíduos (máximo, percentil, quartil, mediana e média) durante os 22 dias de execução do terceiro experimento. O comportamento dos resultados foi semelhante ao que ocorreu nos experimentos anteriores. As curvas apresentam novamente o padrão de se aproximarem em patamares mais altos

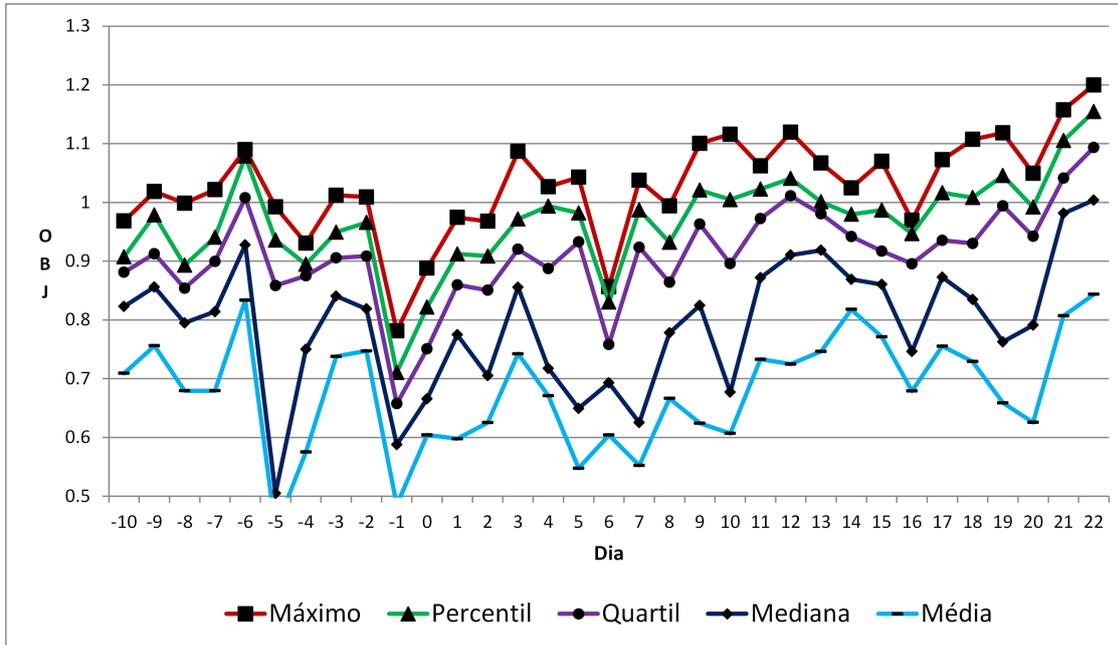
e o resultado do teste de hipótese estatístico confirma a melhora nos valores da função objetivo, com nível de confiança superior a 98%. Neste caso, o que é possível observar como característica distinta são os valores absolutos alcançados. Tanto os valores finais, quanto os máximos observados, são, em geral, inferiores aos exibidos nos experimentos anteriores. Por exemplo, o valor máximo observado não atingiu o valor de 1.2 em nenhum momento, o que ocorreu nos experimentos 1 e 2. Nossa interpretação dessa característica é que nesse experimento foi mais difícil obter melhores resultados do que nos anteriores. Acreditamos que isso seja verdade, uma vez que, utilizamos uma fórmula que já havia sido evoluída como referência e controle do experimento. Em teoria, utilizamos um indivíduo melhor adaptado que a solução de controle dos experimentos anteriores, o que seria uma solução melhor para o sistema e, por sua vez, mais difícil de ser superada.

Os resultados desse experimento apresentam novas vantagens da nossa abordagem evolutiva composta por dois ciclos de avaliação e promoção de soluções. Primeiramente, nos permite supor que a abordagem proposta torna as soluções mais robustas a cada ciclo de promoção. Segundo, se o ambiente estiver suficientemente estável, novos ciclos de evolução terão cada vez mais dificuldade para alcançar os limites (*thresholds*) de promoção. O que é um resultado interessante e desejável pois, em situações estáveis em que o sistema já esteja desempenhando bem, não é necessário aplicar muitas modificações. Por outro lado, alterações no ambiente poderão precipitar o rompimento desse equilíbrio, ao mesmo tempo que facilitam que a evolução alcance os limites para promover novas fórmulas, mais adaptadas ao ambiente modificado.

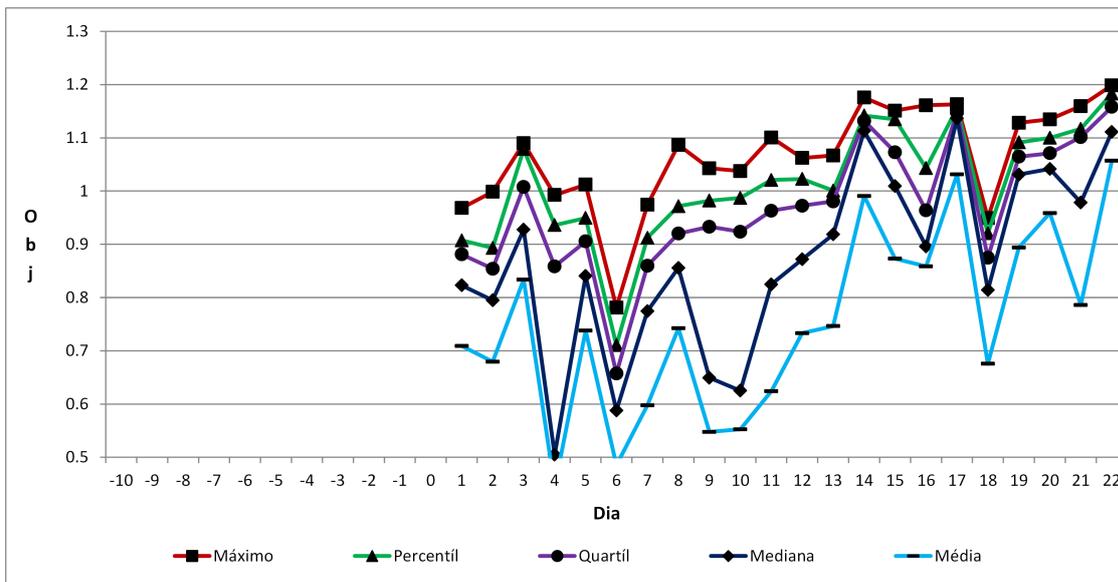
## 7.7 Experimento 4 - reavaliação

O objetivo do terceiro experimento é reproduzir o primeiro experimento em um novo momento com um novo conjunto de dados, verificando se os resultados apresentados nessa segunda reprodução do experimento estão próximos dos obtidos no primeiro experimento. O experimento 4 possui as mesmas restrições e parâmetros utilizados no experimento 1. Além disso, utilizamos a mesma geração inicial, solução de controle e intervalo de execução, 22 dias. Os resultados dos dois experimentos estão apresentados juntos na figura 7.11 para permitir a comparação. Da mesma forma que no primeiro experimento, nós aplicamos um teste de hipótese estatístico, *teste-t*, comparando o primeiro e o último dia. Rejeitamos a hipótese nula, concluindo que os dados desses dois dias são significativamente diferentes com um nível de confiança acima de 98%, assim como ocorreu no experimento inicial. Os resultados, valores e soluções obtidos nesse experimento não são idênticos aos obtidos no primeiro experimento. Isso nos permite concluir que a nossa abordagem é capaz

de evoluir funções para alcançar uma melhor adaptação no ambiente, em momentos e situações distintas.



(a) Exp 1



(b) Exp 4

**Figura 7.11:** Experimento 4 - Valores estatísticos da função objetivo calculados com base nos 40 indivíduos (máximo, percentil, quartil, mediana e média) durante os 22 dias de execução do experimento 4 (b) com o objetivo de reavaliar o resultado do primeiro experimento (a).

## Capítulo 8

# Conclusão

### 8.1 Síntese do trabalho

Neste trabalho estudamos o problema de publicidade computacional que visa

*encontrar o melhor anúncio dentre um conjunto de anúncios disponíveis para exibir a um usuário em um determinado contexto.*

Estabelecemos a hipótese de que

**a abordagem evolutiva permite não apenas encontrar, como também evoluir, funções de reordenação de anúncios com base nos dados observados que combinem as dimensões de valor comercial, desempenho histórico, contexto e usuário para aumentar o valor de utilidade do sistema, permitindo que o sistema se adapte quando modificações no ambiente forem introduzidas.**

A principal contribuição deste trabalho foi definir e implementar a abordagem evolutiva como um ciclo composto por três passos principais responsáveis por **explorar** funções de reordenação de anúncios, **avaliar** o desempenho dessas funções comparando com a função de controle e **evoluir**, ou mesmo criar, novas funções com base nas melhores funções anteriormente avaliadas. Conforme descrevemos, nossa abordagem não só é capaz de descobrir novas funções como também faz isso continuamente por meio da evolução de funções ao longo do tempo, o que permite ao sistema de publicidade computacional responder de forma eficaz a mudanças no ambiente de publicidade. Nossa abordagem evolutiva cria novas funções de reordenação de anúncios combinando as diferentes dimensões do problema e automaticamente avaliando essas funções no ambiente real de publicidade computacional.

Os resultados apresentados no capítulo 7 validaram a nossa implementação. Mostramos como a abordagem evolutiva foi capaz de evoluir funções de reordenação de anúncios a partir de um conjunto de 40 funções iniciais, construindo um novo conjunto de funções que apresentaram desempenho superior ao aferido pelo conjunto inicial, após um período de 22 dias. Além disso, nossos resultados descrevem a segunda etapa de avaliação da melhor função encontrada nos 15 primeiros dias. Esse segundo estágio de avaliação durou uma semana e mostrou que essa função permite um bom potencial de melhoria no eCPM assim como pequenos incrementos também no CTR do sistema. No segundo experimento, avaliamos a abordagem em um período mais longo, dois meses e meio. Os resultados reafirmaram as conclusões do primeiro experimento e indicaram que nossa abordagem é capaz de evoluir soluções e melhorar os resultados do sistema de publicidade computacional também em períodos mais longos. Por fim, o terceiro experimento avaliou nossa abordagem em evoluções consecutivas, reafirmando novamente os resultados dos experimentos anteriores. Além disso, os resultados desse último experimento apresentaram novas características da nossa abordagem evolutiva: as soluções se tornam mais robustas a cada ciclo de promoção; e novos ciclos de evolução têm cada vez mais dificuldade para alcançar os limites (*thresholds*) de promoção. Tais características são desejáveis pois, em situações estáveis em que o sistema já esteja desempenhando bem, não é necessário aplicar muitas modificações. Por outro lado, alterações no ambiente poderão precipitar o rompimento desse equilíbrio, ao mesmo tempo que facilitam que a evolução alcance os limites para promover novas fórmulas, mais adaptadas ao ambiente modificado.

Como uma segunda contribuição, nós apresentamos uma análise detalhada dos blocos básicos com os quais construímos as funções de reordenação e que sustentam a nossa abordagem, ou seja, uma análise detalhada das diferentes dimensões do problema de publicidade computacional: valor comercial, desempenho histórico, contexto e usuário [CBMF14]. Com base nessa análise, nós podemos afirmar que todas as dimensões do problema de publicidade computacional se complementam mutuamente e nenhuma delas deve ser negligenciada para solucionar o problema de reordenação de anúncios.

Como contribuições adicionais nós desenvolvemos diferentes componentes de software que disponibilizamos para a comunidade como software livre de forma que possa auxiliar outros grupos na construção de experimentos e no desenvolvimento de sistemas evolutivos em ambientes reais [BMF14]. Entre esses componentes nós destacamos o **ReConf**, uma inovadora solução que disponibiliza um repositório de configurações centralizado como um serviço web RESTful.

## 8.2 Trabalhos futuros

Nossa abordagem evolutiva determina como obrigatória a utilização nem da nossa implementação do mecanismo de evolução, nem dos cálculos de CTR e relevância dos anúncios para o contexto e usuário. Dessa forma, esses pontos apresentam interessantes oportunidades de extensão ou melhoria que podem ser explorados em trabalhos futuros.

O primeiro ponto de extensão ou melhoria que pode ser abordado em trabalhos futuros é a ampliação das características utilizadas para representar cada dimensão no problema de ordenação de anúncios em publicidade computacional. A nossa abordagem deve funcionar tão bem quanto e, possivelmente, melhor com outras formas e métodos que permitam sintetizar a dimensão como um número real:

- valor comercial - pode ser expandido para incluir outros dados do anúncio ou da campanha, como o saldo disponível ou o orçamento diário;
- desempenho histórico - a noção de CTR do anúncio no período pode ser subdividida para indicar o CTR do anúncio em um determinado subconjunto de contextos ao invés de apenas um único CTR para todos os diferentes contextos;
- contexto - a própria definição de contexto pode deixar de ser apenas a página para incluir também o local, ou área da página, que está exibindo os anúncios. A melhoria pode ser mais significativa ao se utilizar diferentes métodos e características para se calcular a relevância de um anúncio para um contexto, não se limitando a similaridade de cossenos no modelo vetorial clássico de recuperação de informação ou ao conjunto de características que utilizamos neste trabalho;
- usuário - com base nos nossos resultados da análise das dimensões nós acreditamos que aplicar métodos mais avançados para identificar os interesses dos usuários podem fornecer os maiores ganhos no desempenho do sistema de publicidade computacional, dessa forma, essa seria, na nossa opinião, a dimensão com maior potencial para desenvolvimento em novos trabalhos.

O segundo ponto que pode ser explorado é o mecanismo ou motor de evolução. Neste trabalho utilizamos a programação genética pelas diferentes questões já discutidas. Porém, a maior parte dessas questões são suficientemente abertas para permitir que outros métodos ou técnicas de aprendizado possam ser utilizados com nossa abordagem evolutiva. Dependendo da escolha pode ser necessário pré-processar os dados e/ou ajustar a modelagem do problema, como discutimos na seção 7.2 ao considerar outras formas de regressões.

Por fim, listamos alguns pontos menores de melhoria que não conseguimos incluir neste trabalho:

- implementação da amostragem por usuário ao invés de por requisição;
- utilizar otimização multi-objetivo, como indicado na seção 6.8.1;
- análise dos efeitos da sazonalidade, como datas especiais ou mesmo finais de semana;
- análise dos efeitos de diferentes períodos do dia como, por exemplo, manhã, tarde, noite e madrugada;
- ampliar o modelo de evolução para incluir os históricos dos ancestrais de um indivíduo no cálculo da sua adaptação, por exemplo, utilizando os resultados dos ancestrais nas últimas 4 gerações juntamente com o resultado do indivíduo na geração atual;
- expansão da regressão simbólica para incluir também a combinação livre da dimensão de valor comercial.

## Apêndice A

# Exemplos de Fórmulas

Neste apêndice apresentamos um exemplo de fórmula que foi evoluída por meio da aplicação da nossa abordagem no experimento 2, seção 7.5. Na próxima página apresentamos dez fórmulas que fizeram parte da geração inicial do experimento 2.

$$CPCbid_a \times \left( \frac{\frac{(USR_{avg} - USR_{a,u})^{CTR_a * USR_{min}}}{(CTR_a + 0.3) + \frac{CTX_{a,c}}{CTR_{avg}}} + \frac{lg(lg(USR_{max}))}{lg(CTX_{avg}) + (USR_{max} + CTX_{a,c})}}{lg(lg(\frac{CTR_{min}}{CTX_{a,c}})) - (lg((USR_{max} + CTR_{avg})) * lg((CTX_{min}^{CTR_{max}})))} \right)$$

$$CPCbid_a \times CTR_a \quad (A.1)$$

$$CPCbid_a \times CTX_{a,c} \quad (A.2)$$

$$CPCbid_a \times USR_{a,u} \quad (A.3)$$

$$CPCbid_a \times CTX_{a,c} + CTR_a \quad (A.4)$$

$$CPCbid_a \times USR_{a,u} + CTX_{a,c} \quad (A.5)$$

$$CPCbid_a \times CTR_a + USR_{a,u} \quad (A.6)$$

$$CPCbid_a \times CTX_{a,c} + CTR_a + USR_{a,u} \quad (A.7)$$

$$CPCbid_a \times \left( 0.5 + \left( 0.34 \times \frac{CTX_{a,c} - CTX_{avg}}{CTX_{max} - CTX_{min}} + 0.33 \times \frac{USR_{a,u} - USR_{avg}}{USR_{max} - USR_{min}} + 0.33 \times \frac{CTR_a - CTR_{avg}}{CTR_{max} - CTR_{min}} \right) \right) \quad (A.8)$$

$$CPCbid_a \times \left( 0.5 + \left( 0.8 \times \frac{CTR_a - CTR_{avg}}{CTR_{max} - CTR_{min}} + 0.2 \times \frac{CTX_{a,c} - CTX_{avg}}{CTX_{max} - CTX_{min}} \right) \right) \quad (A.9)$$

$$CPCbid_a \times \left( 0.5 \times \frac{\log(1 + CTR_a)}{\log((1 + CTR_{max})^{0.1})} + 0.5 \times \frac{\log(1 + CTX_{a,c})}{\log((1 + CTX_{max})^{0.1})} \right) \quad (A.10)$$

# Referências Bibliográficas

- [AdLMW09] Jesper Andersson, Rogério de Lemos, Sam Malek e Danny Weyns. Modeling dimensions of self-adaptive software systems. Em Betty H.C. Cheng, Rogério de Lemos, Holger Giese, Paola Inverardi e Jeff Magee, editors, *Software Engineering for Self-Adaptive Systems*, volume 5525 of *Lecture Notes in Computer Science*, páginas 27–47. Springer Berlin Heidelberg, 2009. 27
- [AG12] Deepak Agarwal e Maxim Gurevich. Fast top-k retrieval for model based recommendation. *Proceedings of the fifth ACM international conference on Web search and data mining - WSDM '12*, página 483, 2012. 26, 35, 37
- [AGWY14] Deepak Agarwal, Souvik Ghosh, Kai Wei e Siyu You. Budget pacing for targeted online advertisements at linkedin. Em *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, páginas 1613–1619, New York, NY, USA, 2014. ACM. 33
- [APJ12] Deepak Agarwal, Sandeep Pandey e Vanja Josifovski. Targeting converters for new campaigns through factor models. *Proceedings of the 21st international conference on World Wide Web - WWW '12*, página 101, 2012. 8, 34
- [AZZ<sup>+</sup>12] Javad Azimi, Ruofei Zhang, Yang Zhou, Vidhya Navalpakkam, Jianchang Mao e Xiaoli Fern. Visual appearance of display ads and its effect on click through rate. Em *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, páginas 495–504, New York, NY, USA, 2012. ACM. 34
- [BF02] Hemant K. Bhargava e Juan Feng. Paid placement strategies for internet search engines. Em *Proceedings of the 11th international conference on World Wide Web*, WWW '02, páginas 117–123, New York, NY, USA, 2002. ACM. 7
- [BGJM10] Michael Bendersky, Evgeniy Gabrilovich, Vanja Josifovski e Donald Metzler. The anatomy of an ad: Structured indexing and retrieval for sponsored search. Em *Proceedings of the 19th International World Wide Web Conference*, April 2010. 6, 7, 16, 17, 26
- [BJ11] Andrei Broder e Vanja Josifovski. *Introduction to Computational Advertising*. [http://www.stanford.edu/class/msande239/lectures-2011/Lecture 01 Intro.pdf](http://www.stanford.edu/class/msande239/lectures-2011/Lecture%2001%20Intro.pdf). Presented in Stanford University, Autumn 2011. 3, 15
- [BMF14] Marcos Eduardo Bolelli Broinizi, Danilo Mutti e Joao Eduardo Ferreira. Application configuration repository for adaptive service-based systems: Overcoming challenges in an evolutionary online advertising environment. Em *Proceedings of the 2014 IEEE International Conference on Web Services*, ICWS '14, páginas 670–677, Washington, DC, USA, 2014. IEEE Computer Society. 12, 42, 78, 99
- [BYRN11] R. Baeza-Yates e B. Ribeiro-Neto. *Modern Information Retrieval: The Concepts and Technology Behind Search*. Addison Wesley, 2011. 27, 43

- [CAJ08] Deepayan Chakrabarti, Deepak Agarwal e Vanja Josifovski. Contextual advertising by combining relevance with click feedback. Em *Proceedings of the 17th international conference on World Wide Web*, WWW '08, páginas 417–426, New York, NY, USA, 2008. ACM. 38
- [CBMF14] André Casimiro, Marcos Broinizi, Danilo Mutti e Joao Eduardo Ferreira. Principais componentes na ordenação de anúncios: um experimento em ambiente real de publicidade computacional. Em *Proceedings of the Brazilian Symposium on Databases*, SBBD'14, páginas 147–156, Curitiba, PR, Brazil, 2014. 12, 82, 99
- [CDY02] Shu-Heng Chen, John Duffy e Chia-Hsuan Yeh. Equilibrium selection via adaptation: Using genetic programming to model learning in a coordination game. *The Electronic Journal of Evolutionary Modeling and Economic Dynamics*, 15 January 2002. 31
- [CL05] Shu-Heng Chen e Chung-Chih Liao. Agent-based computational modeling of the stock price-volume relation. *Inf. Sci.*, 170(1):75–100, Fevereiro 2005. 31
- [CMP08] Massimiliano Ciaramita, Vanessa Murdock e Vassilis Plachouras. Online learning from click data for sponsored search. Em *Proceedings of the 17th international conference on World Wide Web*, WWW '08, páginas 227–236, New York, NY, USA, 2008. ACM. 6, 8
- [CPVSY06] Weihua Cai, Arturo Pacheco-Vega, Mihir Sen e K. T. Yang. Heat transfer correlations by symbolic regression. *International Journal of Heat and Mass Transfer*, 49(23-24):4352–4359, November 2006. 31
- [CWZ99] Shu-Heng Chen, Hung-Shuo Wang e Byoung-Tak Zhang. Forecasting high-frequency financial time series with evolutionary neural trees: The case of Hang-Seng stock index. Em Hamid R. Arabnia, editor, *Proceedings of the International Conference on Artificial Intelligence, IC-AI '99*, volume 2, páginas 437–443, Las Vegas, Nevada, USA, 28 Junho-1 Julho 1999. CSREA Press. 31
- [CZA<sup>+</sup>12] Haibin Cheng, Roelof Van Zwol, Javad Azimi, Eren Manavoglu, Ruofei Zhang, Yang Zhou e Vidhya Navalpakkam. Multimedia features for click prediction of new ads in display advertising. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*, página 777, 2012. 8, 34, 38
- [DKR11] R. Delfianto, M.L. Khodra e A. Roesli. Content-targeted advertising using genetic programming. Em *Electrical Engineering and Informatics (ICEEI), 2011 International Conference on*, páginas 1–5, July 2011. 37
- [DLG14a] Alex Deng, Tianxi Li e Yu Guo. Statistical inference in two-stage online controlled experiments with treatment selection and validation. Em *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, páginas 609–618, New York, NY, USA, 2014. ACM. 39, 40
- [DLG14b] Alex Deng, Tianxi Li e Yu Guo. Statistical inference in two-stage online controlled experiments with treatment selection and validation. Em *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, páginas 609–618, New York, NY, USA, 2014. ACM. 60
- [eMaa] Advertisers will spend nearly \$600 billion worldwide in 2015, dez 2014. Disponível em <http://www.emarketer.com/Article/Advertisers-Will-Spend-Nearly-600-Billion-Worldwide-2015/1011691>. Acessado em Fevereiro de 2015. viii, 2

- [eMab] Brazil wins at something: Digital ad spending in latin america, jul 2014. Disponível em <http://www.emarketer.com/Article/Brazil-Wins-Something-Digital-Ad-Spending-Latin-America/1011023>. Acessado em Fevereiro de 2015. viii, 2, 3
- [eMac] Global ad spending growth to double this year, jul 2014. Disponível em <http://www.emarketer.com/Article/Global-Ad-Spending-Growth-Double-This-Year/1010997>. Acessado em Fevereiro de 2015. viii, 1
- [EOS<sup>+</sup>05] Benjamin Edelman, Michael Ostrovsky, Michael Schwarz, Thank Drew Fudenberg, Louis Kaplow, Robin Lee, Paul Milgrom, Muriel Niederle e Ariel Pakes. Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. *American Economic Review*, 97, 2005. 16, 24
- [FB12] Ayman Farahat e Michael C. Bailey. How effective is targeted advertising? *Proceedings of the 21st international conference on World Wide Web - WWW '12*, página 111, 2012. 8, 34
- [FFPW04] W. Fan, E.A. Fox, P. Pathak e H. Wu. The effects of fitness functions on genetic programming-based ranking discovery for web search. *Journal of the American Society for Information Science and Technology*, 55(7):628–636, 2004. 35
- [FGPP05] W. Fan, M.D. Gordon, P. Pathak e P. PATHAK. Genetic programming-based discovery of ranking functions for effective web search. *Journal of Management Information Systems*, 21(4):37–56, 2005. 35
- [FPW06] W. Fan, P. Pathak e L. Wallace. Nonlinear ranking function representations in genetic programming-based ranking discovery for personalized search. *Decision Support Systems*, 42(3):1338–1349, 2006. 35
- [FPZ09] W. Fan, P. Pathak e M. Zhou. Genetic-based approaches in ranking function discovery and optimization in information retrieval a framework. *Decision Support Systems*, 47(4):398–407, 2009. 35
- [GBK05] Steven Gustafson, Edmund K. Burke e Natalio Krasnogor. On improving genetic programming for symbolic regression. Em *Congress on Evolutionary Computation*, páginas 912–919, 2005. 31
- [GCBH10] Thore Graepel, Joaquin Quiñonero Candela, Thomas Borchert e Ralf Herbrich. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. Em *Proc. 27th Internat. Conf. on Machine Learning. Morgan Kaufmann, San Francisco, CA*, 2010. 36
- [GHJV95] E. Gamma, R. Helm, R. Johnson e J. Vlissides. *Design Patterns*. Addison-Wesley, 1995. 61
- [GYX<sup>+</sup>10] Siyu Gu, Jun Yan, Zhanyong Xiao, Ning Liu, Dou Shen, Weiguo Fan, Lei Ji, Zheng Chen e Ying Chen. What are driving users to click ads? user habit, attitude, and commercial intention. 2010. 36
- [HM14] Patrick Hummel e Preston McAfee. Machine learning in an auction environment. Em *Proceedings of the 23rd International Conference on World Wide Web, WWW '14*, páginas 7–18, New York, NY, USA, 2014. ACM. 39
- [IB13] Ilknur Icke e Joshua Clifford Bongard. Improving genetic programming based symbolic regression using deterministic machine learning. Em *Evolutionary Computation (CEC), 2013 IEEE Congress on*, páginas 1763–1770. IEEE, 2013. 11, 31

- [Jol05] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2005. 21
- [KDLX14] Ron Kohavi, Alex Deng, Roger Longbotham e Ya Xu. Seven rules of thumb for web site experimenters. Em *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, páginas 1857–1866, New York, NY, USA, 2014. ACM. 60
- [Kei04] Maarten Keijzer. Scaled symbolic regression. *Genetic Programming and Evolvable Machines*, 5(3):259–269, Setembro 2004. 31
- [KLZM11] Maryam Karimzadehgan, Wei Li, Ruofei Zhang e Jianchang Mao. A stochastic learning-to-rank algorithm and its application to contextual advertising. Em *Proceedings of the 20th international conference on World wide web*, WWW '11, páginas 377–386, New York, NY, USA, 2011. ACM. 36
- [Koz92] John R. Koza. *Genetic programming: on the programming of computers by means of natural selection*. MIT Press, Cambridge, MA, USA, 1992. 31
- [LCG<sup>+</sup>06] Anísio Lacerda, Marco Cristo, Marcos André Gonçalves, Weiguo Fan, Nivio Ziviani e Berthier Ribeiro-Neto. Learning to advertise. Em *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, páginas 549–556, New York, NY, USA, 2006. ACM. 6, 7, 8, 16, 36, 38, 76
- [LODL12] Kuang-chih Lee, Burkay Orten, Ali Dasdan e Wentong Li. Estimating Conversion Rate in Display Advertising from Past Performance Data Categories and Subject Descriptors. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*, páginas 768–776, 2012. 8, 34
- [LPAJ12] Yandong Liu, Sandeep Pandey, Deepak Agarwal e Vanja Josifovski. Finding the Right Consumer : Optimizing for Conversion in Display Advertising Campaigns Categories and Subject Descriptors. *Proceedings of the fifth ACM international conference on Web search and data mining - WSDM '12*, páginas 473–482, 2012. 8, 34
- [LSS<sup>+</sup>06] T. L. Lew, A. B. Spencer, F. Scarpa, K. Worden, A. Rutherford e F. Hemez. Identification of response surface models using genetic programming. *Mechanical Systems and Signal Processing*, 20(8):1819–1831, November 2006. 31
- [LWZ<sup>+</sup>10] Wei Li, Xuerui Wang, Ruofei Zhang, Ying Cui, Jianchang Mao e Rong Jin. Exploitation and exploration in a performance based contextual advertising system. Em *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, páginas 27–36, New York, NY, USA, 2010. ACM. 35, 39
- [MBY11] M. Melucci e R. Baeza-Yates. *Advanced Topics in Information Retrieval*. The Information Retrieval Series. Springer, 2011. 3, 25
- [McC11] Trent McConaghy. Ffx: Fast, scalable, deterministic symbolic regression technology. Em *Genetic Programming Theory and Practice IX*, páginas 235–260. Springer, 2011. 11
- [MHS<sup>+</sup>13] H. Brendan McMahan, Gary Holt, D. Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, Sharat Chikkerur, Dan Liu, Martin Wattenberg, Arnar Mar Hrafnkelsson, Tom Boulos e Jeremy Kubica. Ad click prediction: A view from the trenches. Em *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, páginas 1222–1230, New York, NY, USA, 2013. ACM. 33

- [NH97] Thomas P. Novak e Donna L. Hoffman. New metrics for new media: toward the development of web measurement standards. *World Wide Web J.*, 2(1):213–246, Abril 1997. 7, 29
- [NNY14] Gali Noti, Noam Nisan e Ilan Yaniv. An experimental evaluation of bidders' behavior in ad auctions. Em *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, páginas 619–630, New York, NY, USA, 2014. ACM. 33
- [OLL<sup>+</sup>14] Richard J. Oentaryo, Ee-Peng Lim, Jia-Wei Low, David Lo e Michael Finegold. Predicting response in mobile advertising with hierarchical importance-aware factorization machine. Em *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, páginas 123–132, New York, NY, USA, 2014. ACM. 15, 33, 38
- [PDH<sup>+</sup>12] Claudia Perlich, Brian Dalessandro, Rod Hook, Ori Stitelman, Troy Raeder, New York e Foster Provost. Bid Optimizing and Inventory Scoring in Targeted Online Advertising. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*, páginas 804–812, 2012. 8, 34
- [PLM08] Riccardo Poli, William B. Langdon e Nicholas Freitag McPhee. *A Field Guide to Genetic Programming*. Lulu Enterprises, UK Ltd, 2008. ix, 10, 29, 31, 71, 75
- [ReC] ReConf. Disponível em <https://github.com/reconf>. Acessado em Fevereiro de 2015. 52, 53, 61
- [RR07] Leonard Richardson e Sam Ruby. *Restful Web Services*. O'Reilly, first edição, 2007. 53
- [Scu10] D. Sculley. Combined regression and ranking. Em *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, páginas 979–988, New York, NY, USA, 2010. ACM. 38
- [SDV<sup>+</sup>13] N. Staelens, D. Deschrijver, E. Vladislavleva, B. Vermeulen, T. Dhaene e P. Demeester. Constructing a no-reference h.264/avc bitstream-based video quality metric using genetic programming-based symbolic regression. *Circuits and Systems for Video Technology, IEEE Transactions on*, PP(99):1, January 2013. 31
- [SGV11] Rodrigo Silva, Marcos André Gonçalves e Adriano Veloso. Rule-based active sampling for learning to rank. Em *ECML/PKDD (3)*, páginas 240–255, 2011. 6
- [SK11] JamesG. Shanahan e Goutham Kurra. Digital advertising: An information scientist's perspective. Em Massimo Melucci e Ricardo Baeza-Yates, editors, *Advanced Topics in Information Retrieval*, volume 33 of *The Information Retrieval Series*, páginas 209–237. Springer Berlin Heidelberg, 2011. 3, 25, 82, 83
- [SLCC14] Fanhua Shang, Yuanyuan Liu, James Cheng e Hong Cheng. Robust principal component analysis with missing data. Em *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, páginas 1149–1158, New York, NY, USA, 2014. ACM. 21
- [ST09] Mazeiar Salehie e Ladan Tahvildari. Self-adaptive software: Landscape and research challenges. *ACM Trans. Auton. Adapt. Syst.*, 4(2):14:1–14:42, Maio 2009. 27
- [SXY13] Qian Sun, Shuo Xiang e Jieping Ye. Robust principal component analysis via capped norms. Em *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, páginas 311–319, New York, NY, USA, 2013. ACM. 21

- [THT<sup>+</sup>14] Yukihiro Tagami, Toru Hotta, Yusuke Tanaka, Shingo Ono, Koji Tsukamoto e Akira Tajima. Filling context-ad vocabulary gaps with click logs. Em *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, páginas 1955–1964, New York, NY, USA, 2014. ACM. 37, 38
- [TLY<sup>+</sup>11] Jian Tang, Ning Liu, Jun Yan, Yelong Shen, Shaodan Guo, Bin Gao, Shuicheng Yan e Ming Zhang. Learning to rank audience for behavioral targeting in display ads. *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11*, página 605, 2011. 8, 34
- [TPGJ11] Sarah K. Tyler, Sandeep Pandey, Evgeniy Gabrilovich e Vanja Josifovski. Retrieval models for audience selection in display advertising. *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11*, página 593, 2011. 8, 34
- [WHS02] Melius Weideman e Timothy Haig-Smith. An investigation into search engines as a form of targeted advert delivery. Em *Proceedings of the 2002 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology*, SAICSIT '02, páginas 258–258, Republic of South Africa, 2002. South African Institute for Computer Scientists and Information Technologists. 14
- [WPR10] J. Webber, S. Parastatidis e I. Robinson. *REST in Practice: Hypermedia and Systems Architecture*. Theory in practice series. O'Reilly Media, 2010. 50, 53
- [WZCD02] Chingning Wang, Ping Zhang, Risook Choi e Michael D'Eredita. Understanding consumers attitude toward advertising. Em *In: Eighth Americas Conference on Information Systems.*, páginas 1143–1148, 2002. 7, 29
- [XZG12] Tianbing Xu, Ruofei Zhang e Zhen Guo. Multiview hierarchical bayesian regression model and application to online advertising. Em *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, páginas 485–494, New York, NY, USA, 2012. ACM. 34, 36
- [YC04] Tina Yu e Shu-Heng Chen. Using genetic programming with lambda abstraction to find technical trading rules. *Computing in Economics and Finance 2004 200*, Society for Computational Economics, 2004. 31
- [YW12] Shuai Yuan e Jun Wang. Sequential selection of correlated ads by pomdps. Em *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, páginas 515–524, New York, NY, USA, 2012. ACM. 35
- [YWC<sup>+</sup>14] Shuai Yuan, Jun Wang, Bowei Chen, Peter Mason e Sam Seljan. An empirical study of reserve price optimisation in real-time bidding. Em *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, páginas 1897–1906, New York, NY, USA, 2014. ACM. 33
- [ZCM<sup>+</sup>10] Zeyuan Allen Zhu, Weizhu Chen, Tom Minka, Chenguang Zhu e Zheng Chen. A novel click model and its applications to online advertising. Em *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, páginas 321–330, New York, NY, USA, 2010. ACM. 36
- [ZYW14] Weinan Zhang, Shuai Yuan e Jun Wang. Optimal real-time bidding for display advertising. Em *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, páginas 1077–1086, New York, NY, USA, 2014. ACM. 17, 33, 34