



Walcir Fontanini

PROGRAMAÇÃO DE SERVIÇOS WEB POR OTIMIZAÇÃO MULTI-OBJETIVO E TEORIA DOS JOGOS

Campinas
2013



Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação

Walcir Fontanini

PROGRAMAÇÃO DE SERVIÇOS WEB POR OTIMIZAÇÃO MULTI-OBJETIVO E TEORIA DOS
JOGOS

Tese de doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos exigidos para a obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: Automação.

Orientador: Paulo Augusto Valente Ferreira

Este exemplar corresponde à versão final da tese defendida pelo aluno, e orientada pelo Prof. Dr. Paulo Augusto Valente Ferreira

Campinas
2013

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Luciana Pietrosanto Milla - CRB 8/8129

F735p Fontanini, Walcir, 1962-
Programação de serviços web por otimização multi-objetivo e teoria dos jogos
/ Walcir Fontanini. – Campinas, SP : [s.n.], 2013.

Orientador: Paulo Augusto Valente Ferreira.
Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de
Engenharia Elétrica e de Computação.

1. Teoria de jogos. 2. Otimização multiobjetivo. 3. Serviço web. I. Ferreira,
Paulo Augusto Valente, 1958-. II. Universidade Estadual de Campinas. Faculdade
de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Programming web services for multiobjective optimization and game theory

Palavras-chave em inglês:

Game theory

Multiobjective optimization

Web service

Área de concentração: Automação

Titulação: Doutor em Engenharia Elétrica

Banca examinadora:

Paulo Augusto Valente Ferreira [Orientador]

Ricardo Hiroshi Caldeira Takahashi

Marcus Fabius Henriques de Carvalho

Akebo Yamakami

Anésio dos Santos Junior

Data de defesa: 18-12-2013

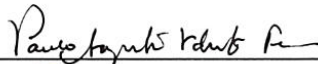
Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE DOUTORADO

Candidato: Walcir Fontanini

Data da Defesa: 18 de dezembro de 2013

Título da Tese: "Programação de Serviços Web por Otimização Multi-Objetivo e Teoria dos Jogos"

Prof. Dr. Paulo Augusto Valente Ferreira (Presidente): 
Prof. Dr. Ricardo Hiroshi Caldeira Takahashi: 
Prof. Dr. Marcius Fabius Henriques de Carvalho: 
Prof. Dr. Akebo Yamakami: 
Prof. Dr. Anésio dos Santos Junior: 

Resumo

O problema de programação de serviços web é considerado. O processo de tomada de decisão em ambientes de negócios web, descritos por tarefas sequenciais e/ou paralelas, envolve selecionar fornecedores de forma ótima dentre um conjunto potencial de provedores de serviços. Características dos serviços como custo, duração de execução, confiabilidade, disponibilidade e reputação são tratadas como múltiplos objetivos a atingir. Inicialmente, a escolha de provedores é feita por Otimização Multi-Objetivo Inteira-Mista, mais especificamente por meio de um modelo de Programação Alvo. Em seguida, o problema de programação de serviços passa a ser tratado no contexto da Teoria dos Jogos, como um jogo envolvendo provedores que buscam maximizar suas utilidades. Diferentes hipóteses sobre a interação entre provedores dão origem a diferentes noções de equilíbrio: Equilíbrio de Nash, Equilíbrio Correlacionado e Equilíbrio de Bayes-Nash. Finalmente, o problema de programação de serviços é modelado como um problema de Leilão de Segundo Preço, o Mecanismo de Leilão Vickrey-Clark-Grooves. A tese inclui exemplos numéricos ilustrativos para todos os modelos propostos.

Palavras-chave: Comércio Eletrônico. Programação de Tarefas. Programação Inteira. Programação Multi-Objetivo. Jogo Bayesiano. Projeto de Mecanismos. Leilão Vickrey-Groves-Clarke.

Abstract

The web services scheduling problem is considered. The decision making process in web-based business environments, described by sequential and/or parallel tasks, involves the optimal selection of suppliers over a set of potential service providers. Characteristics as cost, execution duration, reliability, availability and reputation are treated as multiple objectives to be reached. Initially, the selection of suppliers is performed by Mixed-Integer Multi-Objective Optimization, more specifically, by means of a Goal Programming model. Subsequently, the web services scheduling problem is handled in the Game Theory framework, as a game played by suppliers who aim at maximizing their own utilities. Different hypothesis about the interaction between the suppliers give rise to different equilibrium solutions: Nash Equilibrium, Correlated Equilibrium and Bayes-Nash Equilibrium. Finally, the web services scheduling problem is modeled as a Second Price Auction, the Vickrey-Clark-Grooves Auction Mechanism. The thesis includes illustrative numerical examples for all the models proposed.

Key-words: E-Commerce. Scheduling. Integer Programming. Multiobjective Programming. Bayesian Game, Mechanism Design, Vickrey-Groves-Clarke auction.

Sumário

1	Introdução	1
2	Programação de Serviços Web	5
2.1	Cenário de Aplicação	5
2.1.1	Evolução das Aplicações Web	5
2.1.2	Definição de Serviços Web	7
2.1.3	Construção de uma Aplicação Web	8
2.1.4	O Processo de Negócios	9
2.1.5	Exemplo de Aplicação: Um Planejador de Viagens (<i>Travel Planner</i>) . . .	10
3	Otimização Multiobjetivo	13
3.1	Modelo de Qualidade para Serviços Web	14
3.1.1	CrITÉRIOS de Qualidade de Serviços Web	14
3.1.2	Agregação de Qualidade de Serviços Web	16
3.2	Programação de Serviços Web	17
3.2.1	Uma Solução de Otimização Mista-Inteira Linear	17
3.2.2	A Formulação Multiobjetivo do Problema	19
3.3	Resultados Experimentais	20
3.3.1	Exemplo Ilustrativo	20
3.3.2	Análise Numérica	22
4	Equilíbrio de Nash	23
4.1	Conceitos da Teoria de Jogos Estratégicos	23
4.1.1	Determinação de Todas as Estratégias Mistas	26
4.2	Programação de Serviços Web como Jogo Não Cooperativo	27
4.3	Escolhas Estratégicas dos Jogadores	28
4.4	Estratégias Puras	29
4.5	Utilidade dos Jogadores $U_j(ED_1, \dots, ED_n)$	29
4.6	Cálculo da Utilidade $U_j(ED_1, \dots, ED_n)$	30
4.7	Implementação do Modelo de Otimização	31
4.8	Resultados Numéricos	32
4.8.1	Exemplo 1	32
4.8.2	Exemplo 2	34
4.8.3	Exemplo 3	35

4.9	Resultados pelo Método de Avis-Rosenberg-Savani-Stengel (ARSS)	37
4.9.1	Uma nota sobre a complexidade do cálculo do Equilíbrio de Nash	37
4.9.2	Exemplo 1	39
4.9.3	Exemplo 2	40
4.9.4	Exemplo 3	41
4.9.5	Exemplo 3A	42
4.9.6	Exemplo 3B	42
4.9.7	Exemplo 3C	46
4.9.8	Exemplo 4	48
5	Equilíbrio Correlacionado	51
5.1	Equilíbrio Correlacionado	51
5.2	Passos para Obtenção dos Equilíbrios Correlacionados	53
5.3	Definição dos Sinais	53
5.4	Determinação do Equilíbrio Correlacionado	53
5.5	Resultados Numéricos de Otimização	54
5.5.1	Exemplo 1	54
5.5.2	Exemplo 2	56
5.5.3	Exemplo 3	56
5.5.4	Exemplo 3A	56
5.5.5	Exemplo 3B	56
5.5.6	Exemplo 3C	56
5.5.7	Exemplo 4	56
6	Equilíbrio Bayes-Nash	59
6.1	Utilidade Esperada <i>Ex-Post</i>	62
6.2	Utilidade Esperada <i>Ex-Interim</i>	64
6.3	Utilidade Esperada <i>Ex-Ante</i>	64
6.4	Melhor Resposta Estratégica	64
6.4.1	Equilíbrio de Bayes-Nash	65
6.5	Jogo Induzido pelo Jogo Bayesiano	65
6.6	Programação como Jogo Bayesiano	66
6.7	Resultados Numéricos	67
6.7.1	Exemplo 3	67
6.7.2	Exemplo 4	73
6.7.3	Exemplo 4A	77
7	Programação de Serviços Web por Projeto de Mecanismos	81
7.1	Mecanismos de Revelação	81
7.2	Implementação da Função de Escolha Social por Mecanismos Indiretos	84
7.3	Compatibilidade de Incentivo e o Teorema da Revelação	85
7.4	Mecanismo de Vickrey-Groves-Clarke (VGC)	87
7.5	Resultados Numéricos	89
8	Conclusões	93

DEDICO AOS MEUS FILHOS NICHOLAS E ESTELA, E EM ESPECIAL A MINHA ESPOSA PATRÍCIA.

Agradecimentos

Agradeço,

ao Prof. Paulo Valente pelas longas horas de orientação e nunca desistindo,

a minha esposa Patrícia por toda a sua dedicação e suporte integral nos momentos mais difíceis e

aos meus superiores e colegas do CTI/Centro de Tecnologia da Informação por me disponibilizar tempo para cumprir esta tarefa.

O cientista não é o homem que fornece as verdadeiras respostas; é quem faz as verdadeiras perguntas.

Claude Levi-Strauss

Lista de Figuras

2.1.1 Estrutura de Aplicações Web.	5
2.1.2 Software em Camadas.	6
2.1.3 Aplicações Web em Camadas	7
2.1.4 Aplicação <i>Comprar Livros</i> implementada com serviços web.	8
2.1.5 BPMN: Objetos de Fluxo.	9
2.1.6 <i>Travel Planner</i> mapeado em Serviços Web	11
2.1.7 Planejador de Viagens com Serviço Web.	11
3.1.1 Conversão de processos de negócios em <i>task graphs</i>	15
3.1.2 Grafo de Execução para Múltiplos Paths	17
3.3.1 Exemplo de Diagrama	21
4.3.1 Programação de serviços web como jogo não cooperativo.	28
4.3.2 Funções típicas de custo por tempo de execução.	29
4.8.1 Processo de negócios com duas tarefas sequenciais Exemplo 1.	32
4.8.2 Processo de negócios com cinco tarefas do Exemplo 3.	36
6.0.1 Exemplo de Jogo Bayesiano.	60
6.6.1 Estrutura de custo das tarefas em função dos θ_1 do provedor 1.	67

Lista de Tabelas

3.1	Parâmetros de execução fornecidos ao <i>Solver</i> para o exemplo da Figura 3.3.1. . . .	21
3.2	Exemplo de relatório de otimização para 6 tarefas e 5 provedores de serviços web.	22
3.3	Critérios de qualidade em função do número de provedores de serviços web e tarefas.	22
4.1	Bimatriz com utilidades dos jogadores.	24
4.2	Jogo Dilema dos Prisioneiros.	24
4.3	Melhor Resposta Estratégica do Jogador 1.	24
4.4	Melhor Resposta Estratégica do Jogador 2.	24
4.5	Equilíbrio de Nash para o Dilema dos Prisioneiros.	25
4.6	Melhor resposta estratégica.	25
4.7	Exemplo de Estratégias Mistas.	25
4.8	Exemplo genérico de estratégias mistas.	27
4.9	Cálculo do equilíbrio em estratégias mistas.	27
4.10	Célula típica da bimatriz de utilidades.	30
4.11	Perfil estratégico no formato vetorial ou sequencial.	32
4.12	Tempos e preços de execução do Exemplo 1.	33
4.13	Estratégias dos provedores do Exemplo 1.	33
4.14	Equilíbrios de Nash do Exemplo 1.	33
4.15	Bimatriz associada ao Exemplo 1.	34
4.16	Tempos e custos de execução do Exemplo 2.	34
4.17	Estratégias provedores 1 e 2 do Exemplo 2.	35
4.18	Equilíbrios de Nash do Exemplo 2.	35
4.19	Bimatriz associada ao Exemplo 2.	35
4.20	Tempos e preços de execução do Exemplo 3.	36
4.21	Estratégias dos provedores 1 e 2 do Exemplo 3.	37
4.22	Equilíbrios de Nash do Exemplo 3.	38
4.23	Exemplo 1 resolvido pelo método ARSS.	40
4.24	Exemplo 2 resolvido pelo método ARSS.	41
4.25	Estratégias reduzidas do Exemplo 3.	41
4.27	Exemplo 3 resolvido pelo método ARSS.	42
4.29	Exemplo 3A resolvido pelo método ARSS.	42
4.26	Bimatriz de utilidades para estratégias reduzidas do Exemplo 3.	43
4.28	Bimatriz de utilidades para estratégias reduzidas do Exemplo 3A.	44
4.30	Bimatriz de Utilidades para Estratégias Reduzidas do Exemplo 3B.	45
4.31	Exemplo 3B resolvido pelo método ARSS.	46

4.32	Estratégias reduzidas do Exemplo 3C.	46
4.33	Bimatriz de utilidades para estratégias reduzidas do Exemplo 3C.	47
4.34	Exemplo 3C resolvido pelo ARSS.	48
4.35	Tempo (seg.) e preços em R\$ de execução do Exemplo 4.	48
4.36	Estratégias reduzidas do Exemplo 4.	49
4.38	Exemplo 4 reduzido resolvido pelo método de ARSS.	49
4.37	Bimatriz de utilidades para estratégias reduzidas do Exemplo 4.	50
5.1	Jogo de provedores para serviços web.	51
5.2	Probabilidades.	52
5.3	Sinais como Probabilidades.	53
5.4	Bimatriz com utilidades do Exemplo 1.	55
5.5	Bimatriz com utilidades do Exemplo 2.	57
6.1	Bimatriz do jogo estratégico A.	59
6.2	Bimatriz do jogo estratégico B.	59
6.3	Bimatriz do jogo estratégico C.	59
6.4	Bimatriz do jogo estratégico D.	60
6.5	Utilidades u_1 e u_2 do jogo Bayesiano.	62
6.6	Bimatriz do jogo induzido pelo jogo bayesiano.	66
6.7	Tipos e probabilidades.	67
6.8	Tempos de execução do Exemplo 3.	67
6.9	Preços das tarefas em função de θ Exemplo 3.	68
6.10	Ações reduzidas Exemplo 3.	68
6.11	Bimatriz de utilidades para $\theta_{1,1}$ e $\theta_{2,1}$ Exemplo 3	68
6.12	Bimatriz de utilidades para $\theta_{1,1}$ e $\theta_{2,2}$ Exemplo 3	69
6.13	Bimatriz de utilidades para $\theta_{1,2}$ e $\theta_{2,1}$ Exemplo 3	69
6.14	Bimatriz de utilidades para $\theta_{1,2}$ e $\theta_{2,2}$ Exemplo 3	69
6.15	Estratégias estendidas do Exemplo 3.	70
6.17	Equilíbrio de Nash do jogo com $\theta_1 = \theta_{1,1}, \theta_2 = \theta_{2,1}$ pelo método ARSS Exemplo 3.	71
6.18	Equilíbrios de Nash do jogo com $\theta_1 = \theta_{1,1}, \theta_2 = \theta_{2,2}$ pelo método ARSS Exemplo 3.	71
6.19	Equilíbrios de Nash do jogo com $\theta_1 = \theta_{1,2}, \theta_2 = \theta_{2,1}$ pelo método ARSS Exemplo 3.	71
6.16	Equilíbrio Bayes-Nash do jogo estendido pelo método ARSS Exemplo 3.	72
6.20	Equilíbrio de Nash do jogo com $\theta_1 = \theta_{1,2}, \theta_2 = \theta_{2,2}$ pelo método ARSS Exemplo 3.	73
6.23	Estratégias Reduzidas Exemplo 4.	73
6.24	Bimatriz de utilidades para $\theta_{1,1}$ e $\theta_{2,1}$ Exemplo 4	73
6.25	Bimatriz de utilidades para $\theta_{1,1}$ e $\theta_{2,2}$ Exemplo 4	73
6.21	Tempo de execução em segundos do Exemplo 4.	74
6.22	Preços de execução em R\$ do Exemplo 4.	75
6.26	Bimatriz de utilidades para $\theta_{1,2}$ e $\theta_{2,1}$ Exemplo 4	76
6.27	Bimatriz de utilidades para $\theta_{1,2}$ e $\theta_{2,2}$ Exemplo 4	76
6.28	Estratégias estendidas Exemplo 4.	76
6.30	Estratégias reduzidas Exemplo 4A.	77
6.31	Bimatriz de utilidades para $\theta_{1,1}$ e $\theta_{2,1}$ Exemplo 4A	77
6.29	Exemplo 4 reduzido resolvido pelo método ARSS.	77
6.32	Bimatriz de utilidades para $\theta_{1,1}$ e $\theta_{2,2}$ Exemplo 4A	78
6.33	Bimatriz de utilidades para $\theta_{1,2}$ e $\theta_{2,1}$ Exemplo 4A	78

6.34	Bimatriz de utilidades para $\theta_{1,2}$ e $\theta_{2,2}$ Exemplo 4A	78
6.35	Estratégias estendidas Exemplo 4A.	79
6.36	Exemplo 4A reduzido resolvido pelo método de ARSS.	80
7.1	Exemplo.	87
7.2	Preços e tempos de execução do Exemplo 14.	89
7.3	Plano de atribuição de tarefas do Exemplo 14.	90
7.4	Preços modificados do Exemplo 14.	90
7.5	Pagamentos pelo segundo maior preço do Exemplo 14.	90
7.6	Preços e tempos de execução do Exemplo 3.	90
7.7	Plano de atribuição de tarefas do Exemplo 3.	91
7.8	Preços modificados do Exemplo 3.	91
7.9	Pagamentos pelo segundo menor preço do Exemplo 3.	91

Introdução

Um empreendimento baseado em Internet pode ser visto como composto por vários serviços web a serem fornecidos por provedores com base na tecnologia de Internet. Como o número de serviços web e provedores pode tornar-se muito grande, a utilização de métodos quantitativos para a melhor designação provedor-serviço é geralmente necessária. O problema de programação de serviços web tem despertado grande interesse últimos anos ((Zeng, Benatallah, Ngu, Dumas, Kalagnanam & Chang 2004, Ardagna & Pernici 2007, Huang, Lan & Yang 2009)(Fan, Fang & Jiang 2011)).

Muitas das abordagens adotadas atualmente para o problema são heurísticas, do tipo tentativa-e-erro. Provedores são designados a tarefas sem uma análise mais ampla. Estas heurísticas não integram efetivamente as restrições do processo de negócios – o tempo de execução dos serviço composto pode estar limitado a um dado valor, ou o custo total do negócio não deve exceder o orçamento estipulado – e as preferências do desenvolvedor do negócio.

Nesta tese, o problema de programação de serviços web é formulado com base em cinco objetivos: tempo de execução, custo, reputação, confiabilidade e disponibilidade. Restrições e preferências são associadas a tarefas individuais que constituem o serviço composto. A designação de provedores a tarefas é formulada como um problema de otimização matemática, como em (Zeng et al. 2004, Ardagna & Pernici 2007).

A partir de uma descrição adequada do processo de negócio – a representação BPMN, *Business Process Model Notation* (OMG 2008) – o problema de programação de serviços web é formulado como um problema de Otimização Multiobjetivo (Steuer 1989). Do ponto de vista do projetista, os objetivos são a minimização de custo e do tempo de execução, e a maximização da confiabilidade, reputação e disponibilidade do negócio. O problema de otimização resultante é do tipo linear inteiro, que é então resolvido por um método de Programação Alvo (Gembicki & Haines 1975). Ao usar este método, o projetista pode expressar as suas preferências em relação aos objetivos em termos de metas, e designar provedores a tarefas de forma a minimizar desvios em relação as metas.

A solução ótima para o problema designa provedores para todas as tarefas que constituem o serviço web composto, e assim determina as receitas – utilidades – de todos os provedores.

Certos problemas de programação de serviços web já vem sendo tratados no contexto da Teoria de Jogos ((Shoham & Leyton-Brown 2009)). Em (Khosravifar, Bentahar, Mizouni, Otrok, Alishahi & Thiran 2013), por exemplo, é proposta uma abordagem que analisa estratégias adotadas na composição de serviços web e permite aos provedores equilibrar comportamentos cooperativos e não cooperativos.

Nesta tese, duas linhas principais de modelagem do problema pela Teoria de Jogos são propostas e investigadas. Na primeira, a interação dos provedores é vista como um jogo não cooperativo de informação incompleta, também chamado de jogo Bayesiano. A estrutura do jogo é de conhecimento comum dos provedores, mas cada provedor conhece (provavelmente) apenas a sua própria função utilidade (receita). O Equilíbrio Bayes-Nash do jogo estabelece as estratégias (em termos de utilidades esperadas) que os provedores devem adotar ao competir pelas tarefas do serviço web composto.

Na segunda linha de modelagem, o problema é abordado como um problema de Projeto de Mecanismo, uma importante área da Teoria dos Jogos. Estabelece-se um mecanismo de leilão das tarefas do serviço composto, com uma estrutura tal que, torna ser do interesse dos provedores revelar informações privadas (seus preços para execução de tarefas) de forma verdadeira.

A tese, que inclui resultados de simulação para todos os modelos propostos, está organizada da seguinte forma:

O Capítulo 2 discute o conceito e as tecnologias, tanto de aplicações web comuns, quanto de serviços web. Apresenta a notação de processos de negócios BPMN que será usada em todos os capítulos subsequentes.

O Capítulo 3 apresenta uma solução do problema de programação de serviços web por otimização multiobjetivo. Inicialmente um modelo de qualidade para serviços web é apresentado. Os critérios de qualidade selecionados são: custo, duração da execução, confiabilidade, disponibilidade e reputação. Em seguida, um modelo de agregação de serviços web é formulado (Ardagna & Pernici 2005, Zeng et al. 2004). O problema resultante é do tipo linear-inteiro multiobjetivo, para o qual propomos uma técnica eficiente (Fontanini & Ferreira 2013). A partir do Capítulo 4, a alocação de tarefas é determinada pela minimização do custo total, com restrições sobre o tempo de execução, reputação, disponibilidade e confiabilidade totais.

No Capítulo 4, o problema de programação de serviços web é modelado como um jogo não cooperativo. Conceitos da Teoria de Jogos são discutidos com algum detalhe. Uma modelagem do problema de programação (*scheduling*) como um jogo não cooperativo é apresentada, fazendo provedores se comportarem como jogadores não cooperativos que visam maximizar suas receitas ao adquirirem o direito de executar tarefas mediante estratégias de preços. Os provedores estabelecem as suas estratégias, as quais são então combinadas pelo planejador. O objetivo do planejador passa a ser a minimização do custo total da alocação de tarefas sujeito a restrições sobre tempo de execução, reputação, disponibilidade e confiabilidade. A utilidade de cada provedor é definida como a receita total das tarefas que lhe foram alocadas. Os chamados Equilíbrios de Nash do jogo são caracterizados por meio de exemplos numéricos.

O Capítulo 5 apresenta uma modelagem alternativa para o problema de programação de serviços, que faz uso de Equilíbrio Correlacionado. Após uma breve exposição deste conceito, obtém-se uma solução por Teoria de Jogos computacionalmente mais simples do que a apresentação no Capítulo 4, à custa de hipóteses adicionais sobre o comportamento dos jogadores (provedores).

No Capítulo 6, acrescentamos à modelagem do problema pela Teoria de Jogos, um elemento de incerteza na forma de informações privadas que os provedores podem deter. Normalmente, os tipos de incerteza podem ser quanto as estratégias, quanto ao número de provedores e quanto a utilidade de cada provedor. Neste capítulo, apenas a incerteza sobre as utilidades são consideradas. O modelo resultante leva à caracterização do Equilíbrio Bayes-Nash em termos de utilidades esperadas, ilustrado em uma série de experimentos numéricos (Fontanini & Ferreira 2014).

O Capítulo 7 aborda o problema da programação de serviços web como um problema de

Projeto de Mecanismos, uma subárea tradicional da Teoria de Jogos. O problema é reduzido ao chamado Mecanismo de Vickrey-Groves-Clarke, essencialmente um Leilão de Segundo Preço com uma regra de pagamento característica. O vencedor do leilão é o provedor que ofertar o menor preço, mas é remunerado pela tarefa com o segundo menor preço. Resultados preliminares com o emprego de Mecanismos VGC são apresentados. A abordagem deste capítulo é diferente das empregadas em capítulos anteriores. A ideia é criar um mecanismo tal que seja vantajoso aos provedores divulgarem seus custos e preços.

O Capítulo 8 resume as principais conclusões da tese.

Programação de Serviços Web

2.1 Cenário de Aplicação

2.1.1 Evolução das Aplicações Web

Um serviço web é bastante diferente de uma página web comum porque o serviço web pode fornecer acesso a aplicações através da Internet e entre redes corporativas. Uma página web típica é criada para a interação de um humano, enquanto que um serviço web é desenvolvido para automatizar aplicações; o serviço web é uma tecnologia de software, enquanto que um serviço é uma entidade em um modelo de negócios. Uma tecnologia de serviços web é uma implementação típica para o serviço, mas não a única possibilidade. Por esta razão, um serviço web é identificado como um serviço nesta tese. Para uma exposição detalhada sobre serviços web, ver (Bell 2008, Brown 2008, Papazoglou 2008).

Quando um cliente acessa o web site de uma companhia, ele/ela interage com uma funcionalidade especializada de computador o qual fornece as primeiras páginas HTML de uma aplicação de comércio eletrônico, chamado de servidor web. Para receber um catálogo de produtos de uma companhia, por exemplo, o usuário deve selecionar algum botão na página HTML, e então o servidor web envia um pedido para outro computador especializado que contém as aplicações de negócios, chamado de servidor de aplicação. Devido a existência de várias aplicações corporativas necessárias a usos internos e externos, é prática comum armazenar os dados da aplicação em um computador com uma função especializada, chamado de servidor de bancos de dados. Esta estrutura é ilustrada na Figura 2.1.1.

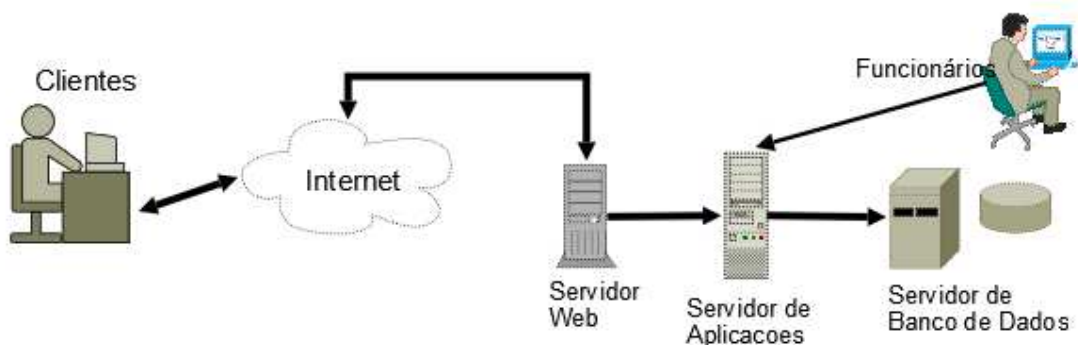


Figura 2.1.1: Estrutura de Aplicações Web.

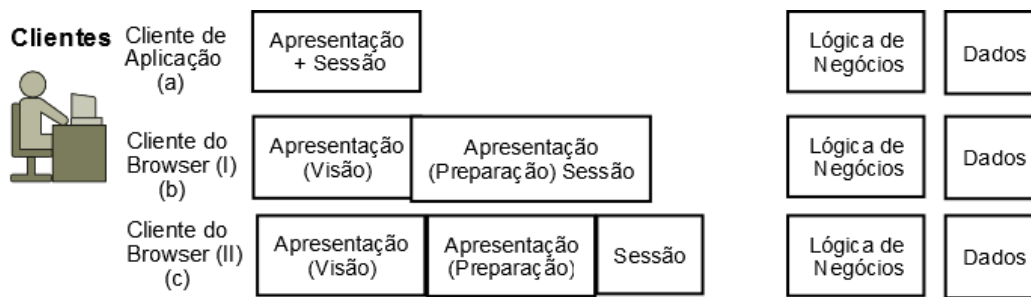


Figura 2.1.2: Software em Camadas.

Uma estratégia para desenvolver aplicações é dividir a aplicação em múltiplas camadas, as quais levam ao conceito de *multitier*. Em sistemas distribuídos, cada camada pode ser alocada a diferentes computadores, que podem ser acessados remotamente. Uma camada vertical acessada remotamente é chamada *tier*, por exemplo, a *Business Logic tier* pode ser usada de várias formas numa aplicação. Existem muitas estruturas em camadas possíveis; uma estrutura muito comum é representada na Figura 2.1.2.

No cenário apresentado na Figura 2.1.2-a, uma aplicação cliente é necessária para instalar um programa que armazena dados relacionados com o estado da sessão, incluindo, por exemplo, o número IP, o *login* do usuário, etc. Quando a lógica de negócios é necessária, a camada lógica de negócios é acessada. A camada lógica de negócios usa a camada de dados para armazenar informações persistentes.

No cenário mostrado na Figura 2.1.2-b, o cliente *browser I* (i.e. um browser web) mostra o conteúdo e coleta entradas do teclado do usuário. O conteúdo deve ser preparado no formato de páginas HTML antes de sua apresentação; o servidor web é responsável por esta apresentação. Em adição, o servidor web não registra o estado da aplicação, e por isso deve ser gerenciado externamente, por meio de um gerenciador de sessão no servidor web. Quando esta operação não é possível devido a existência de vários servidores web, o gerenciamento de sessão centralizado se justifica. O gerenciamento de sessão deve ser manipulado por uma camada separada, como em cliente *browser II* (Figura 2.1.2-c).

Em uma situação real, a configuração necessária depende de vários fatores, tais como o desempenho esperado, fatores de carga do servidor, confiabilidade, etc. Uma implementação típica é ilustrada na Figura 2.1.3. Nesta implementação, o sistema pode ser particionado de

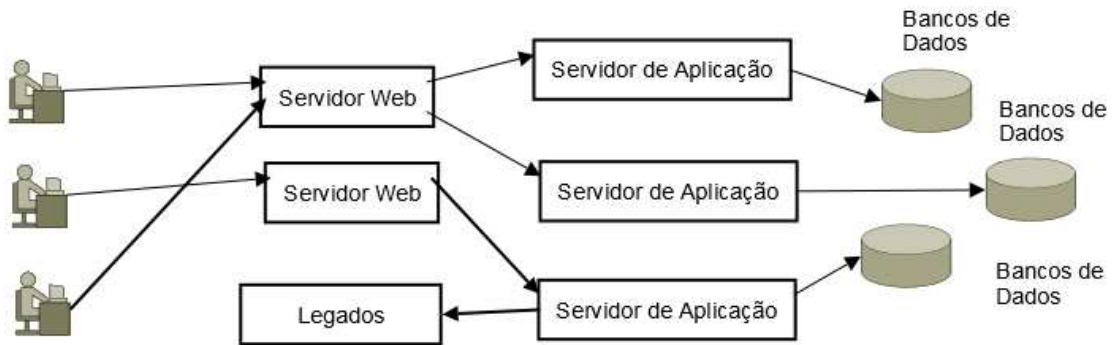


Figura 2.1.3: Aplicações Web em Camadas

muitas formas, incluindo como: (1) conteúdo estático HTML fornecido pelos servidores web; (2) conteúdo dinâmico redirecionado para os servidores de aplicação que diretamente acessam um servidor de banco de dados por informação, tal como um catálogo de produtos e preferências dos usuários, além de gerenciamento de sessão; e (3) se uma lógica de negócios é necessária, então os servidores de aplicação usam os servidores de objetos de negócios na próxima camada, o qual acessa servidores de bancos de dados ou os chamados sistemas legados em *mainframes*.

2.1.2 Definição de Serviços Web

O conceito de serviço web originou-se do conceito de arquitetura orientada à serviço ou *service oriented architecture* (SOA), o qual tem sido definido pelo consórcio internacional *Object Management Group* (OMG) em resposta aos requisitos de departamentos de negócios de grandes corporações e governos. Algumas partes de um processo de negócios típico são automatizadas por um sistema de informação, enquanto outras partes são realizadas por pessoas. Um processo de negócios contém um ou mais *workflows*. Os agentes (ou operadores) executam os objetivos principais de processo de negócio.

Um serviço web pode ser definido como uma aplicação (ou um software de computador) que integra suas interfaces e capacidades de comunicação com outros serviços web usando exclusivamente tecnologia de Internet. Os serviços web podem ser implantados (*deployed*) em servidores web e então agregados a outros serviços web para produzir serviços web compostos. O modelo de processo por trás de um serviço composto descreve funcionalidades necessárias em um serviço web composto, tais como fluxos de dados, fluxo de controle e dependências de transação. Para um melhor entendimento destas funcionalidades, note que o fluxo de dados pode incluir pedaços de informações como nome e idade; o fluxo de controle pode incluir decisões como *se idade menor que 18 anos então obter aprovação dos pais*. As dependências de transação podem descrever condições para uma operação ser completamente bem sucedida ou completamente rejeitada, tais como *fazer reservas somente se ticket foi aprovado*. Um serviço é uma unidade discreta de funcionalidade de negócios que se torna disponível através de um contrato de serviço. O contrato de serviço especifica todas as possíveis interações entre o provedor de serviço e o consumidor de serviço, o qual inclui interfaces do serviço, interfaces de documentos, políticas de serviços, qualidade do serviço e desempenho. A diferença entre um serviço e outras construções de software (tais como componentes e objetos) é que os serviços são explicitamente gerenciados. A qualidade de serviço ou *Quality of Service* (QoS) e o desempenho de serviços são gerenciados por meio de um nível de acordo de serviço ou *service level agreement* (SLA).

O ciclo de vida de um serviço também é gerenciado através das fases de projeto, implantação, melhoria e manutenção. Os padrões para serviços web têm o objetivo de cobrir requisitos de serviços mais genéricos.

2.1.3 Construção de uma Aplicação Web

Uma aplicação web típica, tal como uma loja de livros durante a operação de compra, pode envolver agentes de crédito, de pagamento e de logística. Cada aplicação web específica e respectivos parceiros podem ser encapsulados em serviços web, enquanto que em uma aplicação web convencional, estes parceiros são descritos por programas comuns. Vários parceiros logísticos podem ser agrupados por região, por exemplo, todos os parceiros Sul-Americanos podem ser associados a um serviço web composto.

Considere uma aplicação de software que seja executada em um micro-PC por um consumidor. O programa cria uma lista de livros para o usuário com preços e datas de entrega. O usuário pode usar a lista para comprar somente os livros que estão no seu orçamento. O programa pode ser representado por um fluxograma com somente dois elementos, como na Figura 2.1.4. Se o programa usa o site da livraria para obter preços e para comprar os livros escolhidos pelo usuário, como tal aplicação operaria? De fato, a livraria deveria criar um serviço web específico para vender livros aos seus clientes.¹ A livraria poderia criar este serviço web incluindo duas operações² fundamentais em um programa geral: (i) a habilidade de fazer buscas e (ii) a habilidade de comprar livros através da execução automática do programa. A Figura 2.1.4 ilustra um fluxograma com este serviço web.

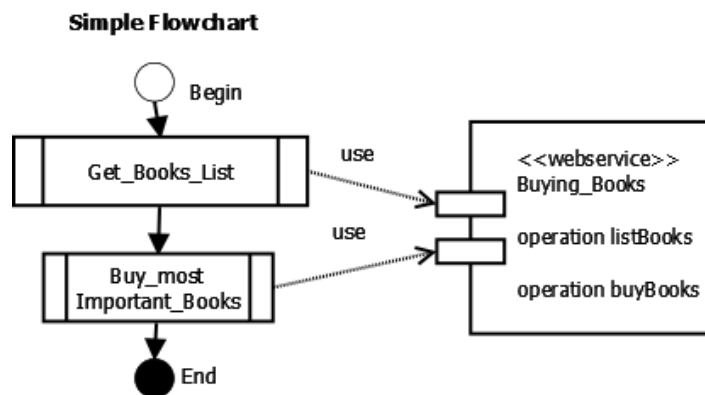


Figura 2.1.4: Aplicação *Comprar Livros* implementada com serviços web.

Um serviço web é um elemento de alto nível de negócios, ao invés de um elemento de baixo nível, tal como uma biblioteca ou um protocolo. Um serviço web é um elemento com características contratuais (por exemplo, preço) e atributos de gerenciamento (por exemplo, disponibilidade). Na Figura 2.1.4, o elemento “*Get_Books_List*” do fluxograma usa a operação “*listBooks*” do serviço web “*Buying_Books*”, e o elemento “*Buy_Most_Important_Books*” usa a operação “*buyBooks*” do serviço web “*Buying_Books*”.

¹Um consumidor poderia ir diretamente ao site da livraria usando um navegador web para fazer suas buscas e comprar os livros. Isto aconteceria através de páginas HTML e não por meio de serviços web.

²Uma operação é um elemento sintático de um serviço web.



Figura 2.1.5: BPMN: Objetos de Fluxo.

2.1.4 O Processo de Negócios

A integração de recursos ou funcionalidades de parceiros para produzir ou compor uma nova funcionalidade representa uma crescente oportunidade estratégica de implementação. A lógica de integração pode ser representada como um modelo de processo de mais alto nível, separado do código-fonte da aplicação real. A descrição do processo de negócios incluída nesta seção é principalmente baseada nos padrões *Business Process Management Notation* (BPMN), conforme apresentado em (OMG 2008)

Um *diagrama de processos de negócios* é um conjunto de elementos gráficos que facilitam o desenvolvimento de diagramas simples do ponto de vista do analista de negócios. Atividades são representadas por retângulos e decisões são indicadas por losângulos. Uma característica essencial do padrão BPMN é um mecanismo para criar modelos de processos de negócios e, ao mesmo tempo, lidar com as suas complexidades. Elementos gráficos têm sido criados para representar categorias de requisitos e permitir um fácil reconhecimento destas categorias, tais como *flow objects*, *connection objects*, *swimlanes* e *artifacts*.

Os *flow objects* contêm três elementos básicos, ou seja, *event*, *activity* e *gateway*.

- Um *event*, indicado por um círculo, representa qualquer coisa que aconteça durante o curso de um processo de negócios. Existem três tipos de eventos com diferentes efeitos no fluxo, incluindo eventos *Begin*, *Intermediate* e *Final*
- Uma *activity* é representada por um retângulo de cantos arredondados. É um termo genérico para qualquer tarefa que uma companhia poderia executar. Uma activity pode ser *atômica* ou *composta*. Os tipos de tarefa são *tarefas simples* e *subprocessos*. A notação de subprocesso é indicada pelo sinal positivo (+) no centro do retângulo.
- Um *gateway*, representado por um losângulo, é usado para controlar a divergência e a convergência de *sequence flow*. As decisões implicadas por este elemento incluem *forking*³, *merging*⁴ e *path combination*. Marcas internas indicam o tipo de comportamento de controle.

Os objetos de conexão são *sequence flow*, *message flow* e *association*. Um *sequence flow* é representado por uma linha sólida com uma flecha e é usado para mostrar a ordem (ou sequência) no qual as atividades de um processo são executadas. Um *message flow* é representado por linha tracejada com flecha e é usado para indicar fluxo de mensagem entre dois processos. Uma

³Este termo é similar ao conceito de processo Unix pelo qual o sistema operacional duplica código em memória a pedido de outros processos Unix clientes.

⁴Isto indica o reverso do *forking*, ou seja, quando dois processos Unix são reunidos ao processo original.

associação é representada por uma linha pontilhada com flecha e é usada para associar dados, texto ou outro artefato⁵ com objetos de fluxo.

Com estes elementos básicos, é possível criar modelos de negócios de mais baixo nível para finalidades de documentação e comunicação. Estes principais elementos e conectores juntos fornecem as características necessárias para suportar a criação de diagramas inteligíveis.

O BPMN suporta *swimlanes* para dois tipos essenciais, ou seja, os elementos *pool* e *lane*. O elemento *pool* contém sequências de atividades de um processo de negócios típico. A *swimlane* representa uma partição de um conjunto de atividades em vários outros *pools*. O elemento *lane* é uma subpartição dentro do *pool* e estende o *pool* verticalmente ou horizontalmente. Na notação BPMN, dois *pools* de diagramas representam dois agentes.

2.1.5 Exemplo de Aplicação: Um Planejador de Viagens (*Travel Planner*)

Considere um serviço web composto chamado de *Travel Planner*, o qual agrega vários serviços web compostos, tais como reserva de passagens aéreas, seguro viagem, reserva de quarto de hotel, aluguel de carro ou bicicleta, e planejamento de rotas turísticas. Todos estes serviços web podem ser executados sequencialmente ou concorrentemente. O modelo de negócios deste programa mais os outros serviços web são apresentados na Figura 2.1.7

Um diagrama BPMN simplificado especificando o serviço web composto Travel Planer é apresentado na Figura 2.1.7. O usuário deve definir o destino e datas de partida e retorno no programa, o qual então usa estas informações para determinar quais vôos estão disponíveis. O próximo passo envolve escolher um hotel com quartos vagos. Várias listas de atrações podem ser fornecidas para ajudar a escolher uma viagem típica. Estas seriam as funcionalidades mais importantes da aplicação Travel Planer.

Neste serviço composto, cada elemento do processo de negócios pode ser realizado por um serviço web parceiro. Reserva de vôo pode ser realizada pelo *PackageFlights*, acomodações em hotel poderia ser feita pelo parceiro *PackagePartner*, e busca de atrações locais poderiam ser implementadas pelo parceiro *PackageINFO*, conforme mostrado na Figura 2.1.6.

Depois destas operações de buscas e reservas terem sido completadas, a distância do hotel para o local de atrações poderia ser calculada (*CalculateDrivingTime*) e um serviço de aluguel de bicicleta (*BikeRental*) ou de aluguel de carro (*CarRental*) poderia ser implementado para completar o planejamento. A Figura-2.1.6 mapeia cada atividade do *Travel Planner* com respeito as operações de cada serviço web específico. Por exemplo, a atividade “*Bycycle Rental*” usa o serviço web *PackageBicycle*, através das operações *getRentValue* e *rentBicycle*.

⁵Em engenharia de software, um artefato pode ser, por exemplo, documento de requisitos, código de programa, um fluxograma e um diagrama *Unified Modeling Language* (UML), diagramas de teste, etc. De fato, artefato é um nome genérico para um produto que é encontrado nas fases intermediárias e finais do processo de desenvolvimento de software.

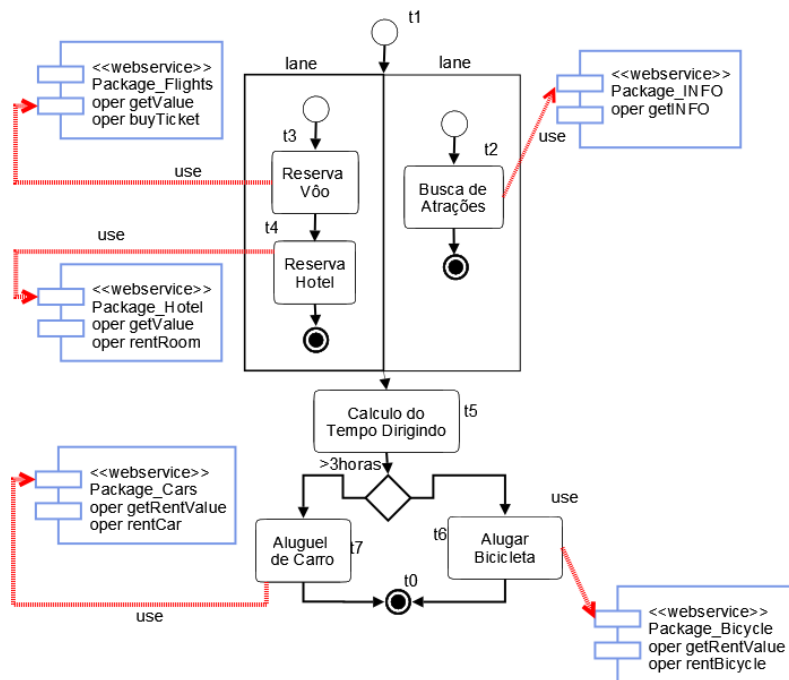


Figura 2.1.6: *Travel Planner* mapeado em Serviços Web

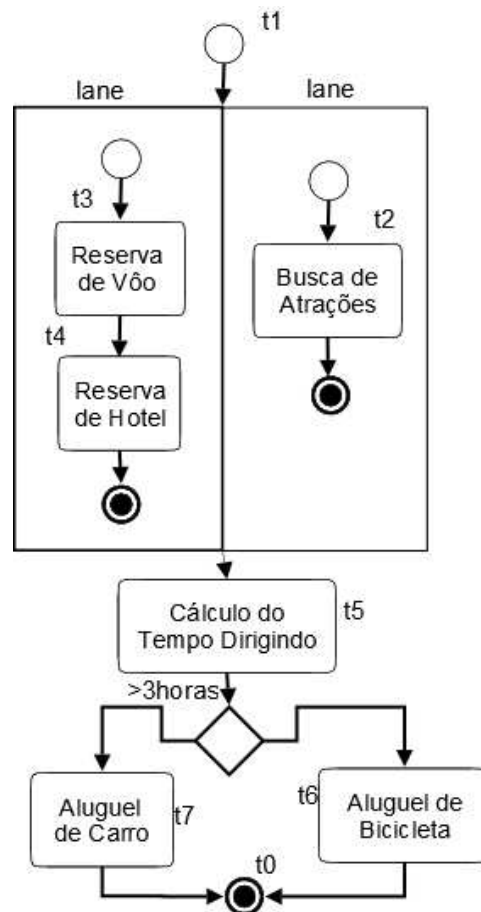


Figura 2.1.7: Planejador de Viagens com Serviço Web.

Otimização Multiobjetivo

Em abordagens modernas para seleção de serviços web, uma estratégia de seleção local é frequentemente adotada, o qual significa que componentes de serviço são atribuídas a tarefas individuais uma-a-uma. Esta estratégia não suporta restrições ou preferências globais dos usuários. Por exemplo, o tempo para a execução de um serviço composto pode ser limitado ou o custo total não pode exceder um orçamento pré-definido.

Serviços componentes que possuem as funcionalidades requeridas são associados a tarefas individuais do serviço composto e invocados durante cada execução do serviço composto. Os serviços disponíveis que atendem a funcionalidade desejada podem ser numerosos e sujeito a mudanças. As abordagens para serviços compostos baseadas em descoberta em tempo de *design* podem ser inapropriadas. A seleção em tempo de execução dos serviços componentes é a abordagem adotada neste texto.

Embora os padrões BPMN tenham sido usados para representar processos de negócios, somente um pequeno subconjunto dos elementos BPMN são usados, tais como *activity*, *gateway*, *begin event*, *end event*, *pool* e *lane*. Nesta seção, os elementos *pool* e *lane* são representados por barras paralelas horizontais. Em adição, um novo elemento é introduzido para identificar explicitamente os ramos THEN e ELSE do *gateway*.

A representação de um processo de negócios pode ser vista como um grafo dirigido, mais conveniente para descrever problemas de otimização. As regras de conversão são como se segue. Primeiro, se o diagrama original contém decisões gateway, então o programa¹ de pré-processamento gera dois grafos, um para o ramo THEN e outro para o ramo ELSE. Segundo, os elementos *pool* e *lane* (ou barras paralelas horizontais para sincronização de tarefas) não são representadas graficamente, pois uma busca em profundidade pode encontrar todos os caminhos. Terceiro, os elementos *activity*, *begin event*, e *end event* são mapeados como tarefas dos grafos. Os *activities* do diagrama BPMN são mapeados em tarefas no modelo de qualidade do serviço web. Em (Ardagna & Pernici 2005, Zeng et al. 2004), diferentes mapeamentos foram usados.

Um conceito que não é incluído em BPMN, mas é necessário para completar o modelo matemático é o de frequência de execução. Quando um gateway é encontrado, é necessário gerar um número real entre 0 e 1 associado com a frequência relativa da execução das alternativas THEN e ELSE. Durante a implementação real, esta frequência seria obtida de dados históricos.

Um processo de negócios pode ser representado como um *task graph*, uma terminologia adotada em (Sinnen 2007), mais conveniente para descrever problemas de otimização. Dado que uma representação de negócios pode conter construções IF-THEN-ELSE, uma possibilidade seria

¹Este é um programa Java que automatiza a conversão de grafos dirigidos BPMN.

gerar dois *task graphs* com frequências de execução específicas. Um *execution path* pode incluir atividades paralelas, mas não atividades em ambos os ramos THEN e ELSE simultaneamente. A Figura 3.1.1 ilustra a conversão de um diagrama BPMN em dois *tasks graphs*.

As regras de conversão propostas nesta pesquisa seriam como se segue. Em primeiro lugar, se um diagrama BPMN original contem decisões *gateway*, então um programa de pré-processamento – um programa Java que converte o diagrama BPMN em um *task graph* – gera dois grafos, um para o ramo THEN e outro para o ramo ELSE. Em segundo lugar, os elementos *pool* e *lane* (ou as barras horizontais paralelas usadas para sincronização de tarefas) não são representadas graficamente porque uma busca em profundidade pode encontrar todos os caminhos de execução. Em terceiro lugar, os elementos *activity*, *begin* e *end* são representados como nós no *task graph*. Os elementos do grafo são, por sua vez, convertidos em modelos matemáticos de otimização, os quais consideram todos os critérios de qualidade (desempenho) selecionados. Modelos diferentes de otimização são usados em (Ardagna & Pernici 2005) e (Zeng, Benatallah, Dumas, Kalagnanam & Sheng 2003).

Um conceito não incluído nos padrões BPMN, mas necessário para completar o modelo matemático é o de **frequência de execução**. Quando um gateway origina elementos THEN e ELSE, é necessário gerar um número real entre 0 e 1 associados com a frequência relativa da execução daquele elemento. Em implementações práticas, as frequências de execução são obtidas de dados gravados.

3.1 Modelo de Qualidade para Serviços Web

No presente estudo, um modelo de qualidade do serviço web baseado em cinco objetivos – tempo de execução, preço, reputação do serviço, confiabilidade e disponibilidade – é considerado. Este modelo de qualidade é usado para criar um modelo global para selecionar serviços. Restrições de qualidade e preferências são associadas a serviços compostos ao invés de as tarefas individuais dentro do serviço composto. O problema da seleção de serviço é então formulado como um problema de otimização matemática.

3.1.1 Critérios de Qualidade de Serviços Web

Um modelo de qualidade para serviços web é proposto em (Zeng et al. 2003, Zeng et al. 2004), cinco critérios de qualidade foram escolhidos para os serviços web elementares e então aplicados aos serviços web compostos, incluindo (1) custo de execução, (2) tempo de execução, (3) reputação, (4) confiabilidade e (5) disponibilidade.

1. **Custo de Execução.** Para um serviço web específico i do provedor j , o custo de execução EC_{ij} é uma quantia que o solicitante do serviço deve pagar pela execução da operação implícita do serviço web. Ou os provedores de serviço explicitamente anunciam os custos de execução ou os custos podem ser acessados *online*.
2. **Duração da Execução.** Para um serviço web específico i do provedor j , a duração da execução ED_{ij} mede o atraso médio entre o pedido da execução e o retorno da saída. A duração da execução é calculada pela expressão $ED_{ij} = T_{process} + T_{transp}$, pois a duração da execução é o tempo de processamento $T_{process}$ somado ao tempo de transmissão T_{trans} . Os provedores de serviços web publicam seus tempos de processamento ou fornecem meios

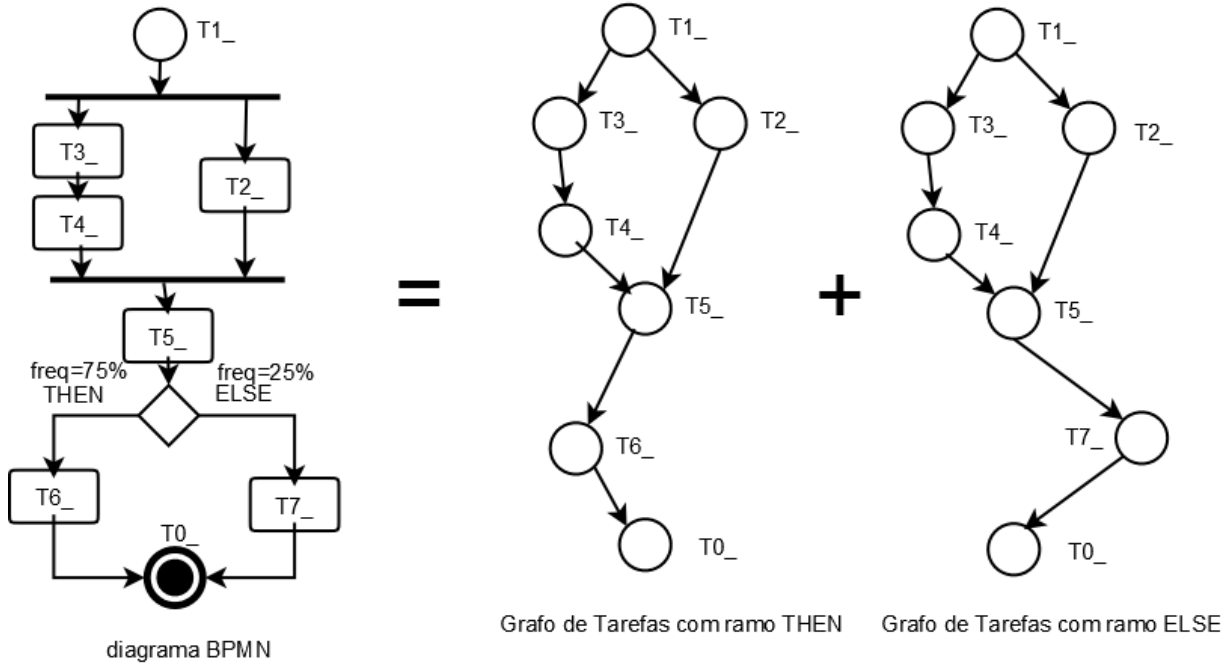


Figura 3.1.1: Conversão de processos de negócios em *task graphs*.

para obter esta informação *online*. O tempo de transmissão é estimado de execuções passadas na forma.

$$T_{trans} = \frac{\sum_{k=1}^n T_k}{n}, \quad (3.1.1)$$

onde T_k é o tempo de transmissão da k -ésima observação do tempo de transmissão, e n é o número de execuções observadas.

3. **Confiabilidade.** A confiabilidade R_{ij} de um serviço web específico i do provedor j é a probabilidade de que um pedido seja corretamente respondido dentro de um intervalo de tempo esperado. Esta informação deveria também ser publicada online na descrição do serviço. Confiabilidade é uma medida relacionada a infra-estruturas de software e hardware; depende do provedor e da qualidade da conexão entre o solicitante do serviço e o provedor. O valor numérico da confiabilidade é calculado de dados históricos em acessos passados, com a expressão $R_{ij} = N_c/K$, onde N_c é o número de respostas corretas do serviço web entregues no intervalo de tempo esperado, e K é o número total de acessos.
4. **Disponibilidade.** A disponibilidade AV_{ij} de um serviço web i do provedor j é a probabilidade de que o serviço esteja disponível. A disponibilidade é calculada pela expressão $AV_{ij} = T_a/\theta$, onde T_a é o tempo total (em segundos) de que o serviço i esteja disponível durante os últimos θ segundos, e θ é uma constante fixada pelo administrador do serviço. O valor de θ depende da aplicação envolvida e varia significativamente. Em aplicações nas quais o serviço web é frequentemente acessado (e.g., mercado de ações), um pequeno valor de θ seria uma melhor aproximação para disponibilidade do serviço. Se o serviço

é pouco usado (e.g., livraria *online*), um grande valor para θ seria mais apropriado. É possível receber mensagens dos provedores de serviço sobre a disponibilidade de seu estado interno.

5. **Reputação.** A reputação RP_{ij} de um serviço web i do provedor j é uma medida subjetiva da confiança do cliente. Depende de experiências anteriores usando o serviço web. Usuários diferentes podem ter opiniões distintas sobre o mesmo serviço. Então, reputação é definida como um ranking médio do serviço web de acordo com seus usuários:

$$RP_{ij} = \frac{\sum_{k=1}^n Rank_k}{n}, \quad (3.1.2)$$

onde $Rank_k$ é o ranking dado pelo usuário sobre o serviço, e n é o número de vezes que o serviços foi avaliado. É prática comum que usuários avaliem webservices na faixa de 0 até 5, por exemplo.

3.1.2 Agregação de Qualidade de Serviços Web

Como mostrado na Figura 3.1.2, um diagrama de processos de negócios pode ter muitos caminhos de execução, dependendo da presença de *gateways* intermediários e processamentos paralelos. Para agregar os parâmetros de qualidade de um processo de negócios, a seguinte notação é introduzida: ep_l é um caminho de execução (*execution path*) que contém sequências de tarefas paralelas (incluindo duas tarefas especiais, *begin* e *end*); L é o conjunto de todos os caminhos de execução; $freq_l$ é a frequência de execução de cada caminho de execução ep_l ; um subcaminho (*subpath*) sp_m^l não contém sequências paralelas e é um subconjunto de um *execution path*; um plano de execução (*execution plan*) epl^l para um ep_l é um conjunto de pares ordenados (i, j) , onde o primeiro elemento deste par é uma tarefa i pertencente ao ep_l e o segundo elemento é um provedor de serviço web j .

[1] **Duração da Execução, ED.** A duração da execução de um plano epl^l é a maior soma de todos os tempos de execução das tarefas em cada *subpath* sp_m^l que pertencem ao *execution path* ep_l :

$$ED_l = \max_{sp_m^l \in ep_l} \sum_{i \in sp_m^l; (k,j) \in epl^l; k=i} ED_{ij}.$$

[2] **Custo de Execução, EC.** O custo de execução de um plano epl^l para um *execution path* ep_l é a soma dos preços de execução EC_{ij} do plano:

$$EC_l = \sum_{(i,j) \in epl^l} EC_{ij}.$$

[3] **Reputação, RP.** A reputação de um plano epl^l para um *execution path* ep_l é uma soma média ponderada das reputações individuais RP_{ij} de acordo com o plano epl^l ; N é o número de elementos na amostra:

$$RP_l = \frac{1}{N} \sum_{(i,j) \in epl^l} RP_{ij}.$$

[4] **Confiabilidade, R.** A confiabilidade de um plano epl^l para um *execution path* ep_l é:

$$R_l = \prod_{(i,j) \in epl^l} R_{ij}$$

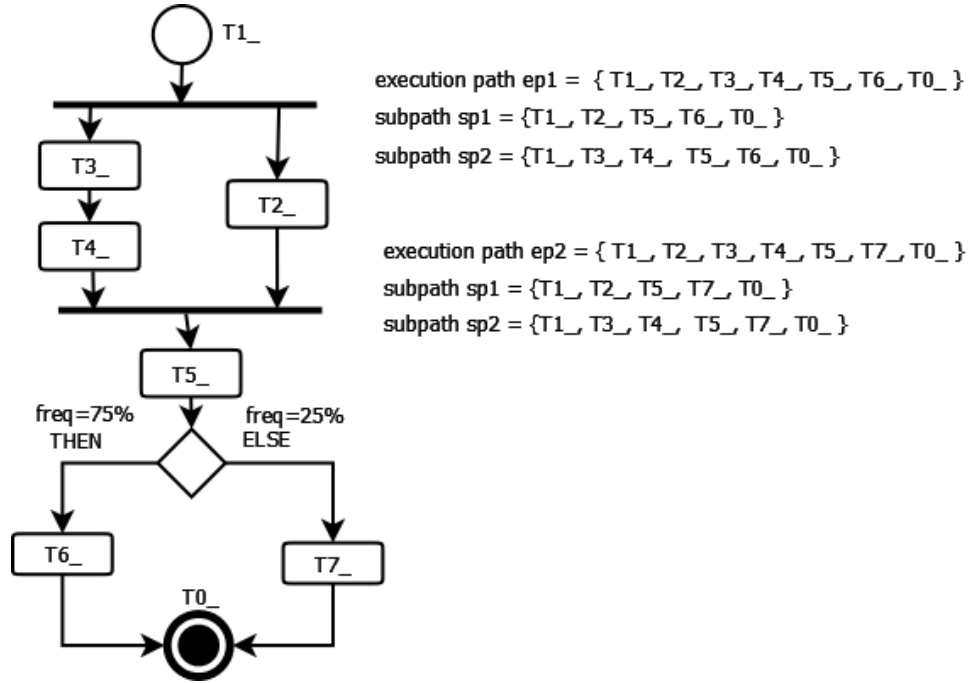


Figura 3.1.2: Grafo de Execução para Múltiplos Paths

[5] **Disponibilidade, AV.** A disponibilidade de um plano epl^l para um *execution path* ep_l é:

$$AV_l = \prod_{(i,j) \in epl^l} AV_{ij}.$$

3.2 Programação de Serviços Web

A programação de serviços web proposto neste capítulo é baseada em políticas de seleção que incluem parâmetros dos pedidos, características dos agentes, dados de execuções passadas e *status* de execuções em andamento. Uma restrição de orçamento garantindo que o custo total não exceda um valor pré-fixado também é incluída.

3.2.1 Uma Solução de Otimização Mista-Inteira Linear

O modelo de otimização abaixo envolve elementos relacionados a programação inteira (Eiselt, Sandblom, Spielberg, Richards, Smith, Laporte & Boffey 2000, Wolsey 1998). As variáveis escolhidas y_{ij} indicam se o serviço web i do provedor j participa em um dado plano de execução ($y_{ij} = 1$) ou não ($y_{ij} = 0$).

1. Restrições de duração e custo de execução.

Seja A o conjunto de todas as tarefas do grafo de processo. Assume-se que existe um conjunto dos provedores de serviços web WS que podem ser atribuídos à tarefa $i \in A$. Entretanto, para cada tarefa i , somente um serviço web deve ser atribuído. A participação de um provedor em um plano selecionado é governada pela seguinte restrição:

$$\sum_{j \in WS} y_{ij} = 1, \forall i \in A. \quad (3.2.1)$$

A variável ST_{il} representa a data de início mais cedo da tarefa i em um *execution path* ep_l ; a variável TD_{il} representa a duração da execução da tarefa i do *execution path* ep_l . A notação $j \rightarrow k$ indica que a tarefa j precede diretamente a tarefa k . O parâmetro MXT é o máximo tempo para completar a execução de um plano. As seguintes restrições se aplicam:

$$\sum_{j \in WS} ED_{ij} y_{ij} = TD_{il}, \forall i \in A, \forall ep_l \in L, \quad (3.2.2)$$

$$(TD_{jl} + ST_{jl}) \leq ST_{kl}, \forall j \rightarrow k, \quad j, k \in A, \quad \forall ep_l \in L, \quad (3.2.3)$$

$$ED_l \geq \sum_{i \in sp_m^l} TD_{il}, \forall ep_l \in L, \quad (3.2.4)$$

$$ED_l \leq MXT, \forall ep_l \in L. \quad (3.2.5)$$

A restrição (3.2.2) impõe que a duração da execução para cada tarefa i em um *execution path* ep_l seja igual à soma das durações de execução dos provedores $j \in W$. Restrição (3.2.3) captura o fato de que, se a tarefa k em um *execution path* ep_l segue imediatamente a tarefa j , então a tarefa k não inicia antes de se completar a tarefa j . A restrição (3.2.4) indica que o tempo de execução de cada *execution path* é maior ou igual do que a soma entre os tempos de execução de todos os *subpaths* sp_m^l . A restrição (3.2.5) garante que a duração total de execução de um *execution path* nunca excederá o limite MXT . (Seria incorreto considerar o parâmetro MXT como a soma de todos os *execution path*, pois o ambiente de *run-time* realiza somente um *execution path*.).

O parâmetro MXC é o orçamento disponível para implementar o plano de execução. A variável EC_l representa o custo de um *execution path* ep_l ; EC_{ij} representa o custo da tarefa i quando realizada pelo provedor j . As seguintes restrições se explicam:

$$\sum_{i \in ep_l, j \in WS} EC_{ij} y_{ij} = EC_l, \quad \forall ep_l \in L, \quad (3.2.6)$$

$$EC_l \leq MXC, \quad \forall ep_l \in L. \quad (3.2.7)$$

De acordo com (3.2.6), o custo total de um *execution path* ep_l é a soma de custos de cada serviço web selecionado para executar a tarefa i no *execution path* ep_l . A restrição (3.2.7) indica que o custo total de qualquer *execution path* nunca excede MXC .

2. A restrição de reputação.

O parâmetro $MNRP$ é a mínima reputação esperada, enquanto que a variável RP_l é a reputação do *execution path* ep_l . O parâmetro RP_{ij} representa a reputação esperada da tarefa i quando implementada pelo provedor j . Segue-se então que:

$$RP_l = \sum_{i \in ep_l, j \in WS} RP_{ij} y_{ij} / |ep_l|, \quad \forall ep_l \in L, \quad (3.2.8)$$

$$RP_l \geq MNRP, \quad \forall ep_l \in L, \quad (3.2.9)$$

onde $|ep_l|$ é a número de tarefas em ep_l .

3. Restrições de disponibilidade e confiabilidade.

O parâmetro MNA é a mínima disponibilidade esperada e a variável AV_l é a disponibilidade do *execution path* ep_l . O parâmetro AV_{ij} representa a disponibilidade esperada da tarefa i quando implementada pelo provedor j . As funções de agregação para a disponibilidade são não lineares. Sua linearização é fornecida pelo logaritmo neperiano ($\ln$):

$$AV_l \geq \ln(MNA), \quad \forall ep_l \in L, \quad (3.2.10)$$

$$AV_l = \sum_{i \in ep_l, j \in WS} \ln(AV_{ij})y_{ij}, \quad \forall ep_l \in L. \quad (3.2.11)$$

Do mesmo modo, a confiabilidade é representada por:

$$R_l \geq \ln(MNRL), \quad \forall ep_l \in L, \quad (3.2.12)$$

$$R_l = \sum_{i \in ep_l, j \in WS} \ln(R_{ij})y_{ij}, \quad \forall ep_l \in L. \quad (3.2.13)$$

Os seguintes objetivos expressam os critérios de desempenho desejados:

$$\min f_1 = \sum_{ep_l \in L} freq_l EC_l \quad (3.2.14)$$

$$\min f_2 = \sum_{ep_l \in L} freq_l ED_l, \quad (3.2.15)$$

$$\max f_3 = \sum_{ep_l \in L} freq_l RP_l, \quad (3.2.16)$$

$$\max f_4 = \sum_{ep_l \in L} freq_l AV_l, \quad (3.2.17)$$

$$\max f_5 = \sum_{ep_l \in L} freq_l R_l. \quad (3.2.18)$$

3.2.2 A Formulação Multiobjetivo do Problema

Nesta tese, o problema de otimização derivado da análise prévia dos ambientes de *run-time* dos serviços web é formulado como um problema de otimização multiobjetivo e resolvido pelo método *goal attainment* (Gembicki & Haimes 1975). Dado um conjunto de objetivos possíveis $f_1, f_2, \dots, f_m$ definidos sobre uma região factível X , o seguinte problema de otimização é considerado:

$$\left| \begin{array}{l} \min \sigma \\ s.a \quad f_i(x) - \sigma w_i \leq \underline{f}_i, \quad i = 1, \dots, m \\ \quad \quad \quad x \in X, \end{array} \right. \quad (3.2.19)$$

onde $\underline{f}_i$, $i = 1, \dots, m$ são metas correspondentes aos objetivos e w_i , $i = 1, \dots, m$ são pesos associados às metas; σ é uma variável escalar. É comum assumir que $\underline{f}_i = f_{i,min}$, $i = 1, \dots, m$, ou seja, cada meta é igual ao mínimo de cada função objetivo individual sobre a região factível. Então, assumindo que $w_i = \underline{f}_{i,min}$, $i = 1, \dots, m$ e resolvendo (3.2.19), minimizamos a maior

variação percentual dos objetivos em relação aos seus mínimos. Como em (3.2.14)-(3.2.18) os objetivos $i = 3, 4, 5$ devem ser maximizados, o problema (3.2.19) é reformulado como se segue:

$$\left| \begin{array}{ll} \min \sigma \\ s.a. & f_i(x) - \sigma w_i \leq \underline{f}_i, \quad i = 1, 2 \\ & f_i(x) + \sigma w_i \geq \bar{f}_i, \quad i = 3, 4, 5 \\ & x \in X, \end{array} \right. \quad (3.2.20)$$

onde $\bar{f}_i = f_{i,max}$, $i = 3, 4, 5$, ou seja, cada meta é o máximo de cada função objetivo correspondente sobre a região factível X . Veja (Steuer 1989), para uma revisão adicional sobre otimização multiobjetivo.

Valores ideais $(\underline{f}_i, \bar{f}_i)$ fornecem os limites mínimos e máximos de performance do serviço web composto. Eles são convenientes para apresentar o método, mas não estritamente necessários, como qualquer conjunto significativo de metas se aplicariam. O vetor de pesos determina a direção da melhoria: quanto menor o peso w_i , mais importante para o projetista é o objetivo f_i . Em adição, o valor do objetivo ótimo do problema (3.2.20) tem uma interpretação clara: indica o máximo desvio das metas. Um ótimo $\sigma^* > 0$ significa que ao menos um valor do objetivo não satisfaz as desigualdades; $\sigma^* \leq 0$, caso contrário. A função objetivo (atributo QoS composto) responsável pelo σ^* é também determinada.

O problema de composição de serviços web em sua formulação clássica misto linear inteira é equivalente ao Problema da Mochila com Múltiplas Escolhas e Multidimensional (PMMM) (Akbar, Rahman, Kaykobad, Manning & Shoja 2006). Como toda instância do PMMM pode ser formulada como um problema clássico de composição web (Ardagna & Pernici 2005), tal problema é NP-Hard. A formulação em programação por metas (3.2.20) é PMMM com restrições adicionais, as quais, em princípio, nos força vê-lo como um problema de otimização mista-inteira geral. Neste caso, (3.2.20) pertence aos problemas NP-Completo (Nemhauser & Wolsey 1999). Na prática, isto significa que grandes instâncias de problema (3.2.20) podem requerer o uso eficiente de metaheurísticas.

3.3 Resultados Experimentais

3.3.1 Exemplo Ilustrativo

Para testar numericamente nossa abordagem, um conjunto de parâmetros aleatórios – como, por exemplo, custos – foram criados por meio de um programa Java específico que converte grafos de execução em estruturas de dados necessárias ao *Solver* GnuLPK (Gnu 2008). Um conjunto de grafos de execução para o processo de negócios foi criado. Estes grafos não são aleatórios, pois devem ser consistentes. Cada *execution subpath* que seja dependente dos elementos IF-THEN-ELSE do grafo foi executado com igual frequência. Por exemplo, considere o processo de negócios ilustrado na Figura 3.3.1 composto de três *execution paths* $\{ep_0, ep_1, ep_2\}$. A Tabela 3.1 descreve os parâmetros usados para resolver o problema de otimização associado. O número de provedores foi fixado em cinco.

O tempo que o usuário espera em uma fila até seu serviço começar foi adicionado ao tempo de execução do serviço web.

O parâmetro tempo de execução variou entre 0.5 até 8 segundos segundo uma distribuição uniforme. Todos os testes foram realizados em um micro-PC Dual-core 3GHz com Windows XP.

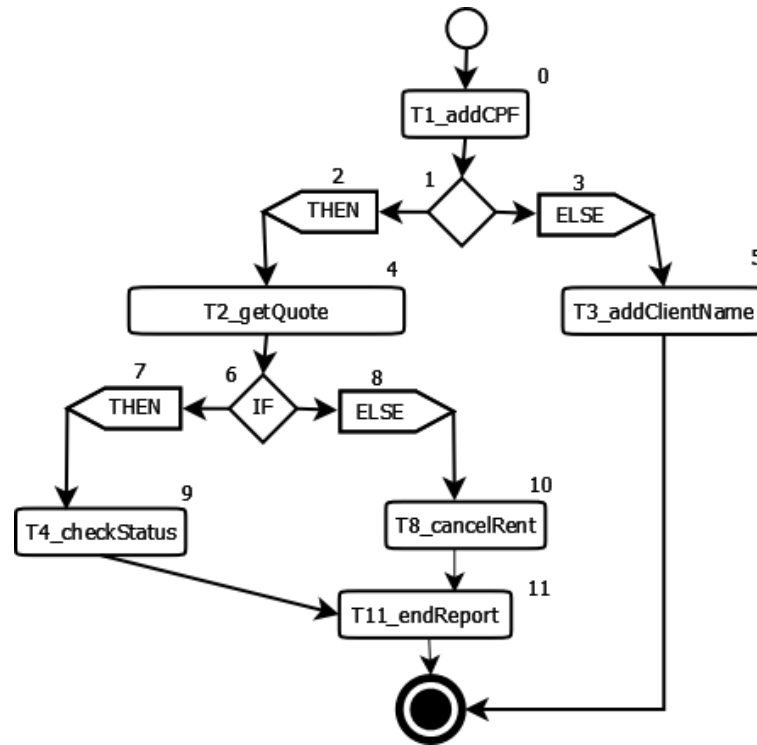


Figura 3.3.1: Exemplo de Diagrama

	Min (ou Max) Esperado	ep_0	ep_1	ep_2
Custo	≤ 266	127	349	322
Tempo	≤ 23.74	11	29.6	30.58
Reputação	≥ 0.92	0.95	0.89	0.92
Disponibilidade	≥ 0.82	0.90	0.75	0.80
Confiabilidade	≥ 0.36	0.54	0.27	0.28

Tabela 3.1: Parâmetros de execução fornecidos ao *Solver* para o exemplo da Figura 3.3.1.

	Solução	Meta Ideal e Peso	Variação Percental (%)
Custo	356	266	33,83
Tempo	30.36	23.74	27,89
Reputação	0.8952	0.9200	2,70
Disponibilidade	0.7941	0.8169	2,79
Confiabilidade	0.3638	0.3754	3,09

Tabela 3.2: Exemplo de relatório de otimização para 6 tarefas e 5 provedores de serviços web.

Objetivo	$10_{ws} \times 25_{task}$	$25_{ws} \times 25_{task}$	$50_{ws} \times 25_{task}$	$10_{ws} \times 5_{task}$	$25_{ws} \times 5_{task}$
<i>Custo</i>	0.9420	1.2410	1.0697	1.3958	0.9479
<i>Tempo</i>	0.5459	1.1259	0.9172	1.5692	1.3860
<i>Reputação</i>	0.0528	0.0755	0.1038	0.0931	0.0638
<i>Disponibilidade</i>	0.0460	0.1220	0.1451	0.0642	0.0831
<i>Confiabilidade</i>	0.0124	0.0286	0.3239	0.1479	0.0525
σ^*	0.9420	1.2410	1.0697	1.5692	1.3860

Tabela 3.3: Critérios de qualidade em função do número de provedores de serviços web e tarefas.

Na Tabela 3.2, um relatório produzido pelo solver é apresentado. O *Solver* foi inicialmente executado para cada objetivo individual em (3.2.14)-(3.2.18), gerando valores ideais para *custo* = 266, *tempo* = 23.74, *reputação* = 0.9200, *disponibilidade* = 0.8169 e *confiabilidade* = 0.3744. O *Solver* foi executado uma sexta vez para resolver o problema de programação por metas (3.2.20) com as metas ideais obtidas anteriormente. O método *goal attainment* retornou *custo* = 356, *tempo* = 30.36, *reputação* = 0.8952, *disponibilidade* = 0.7941 and *confiabilidade* = 0.3638. Ao contrário das abordagens propostas em (Zeng et al. 2003) e (Ardagna & Pernici 2005) baseadas na minimização de uma soma ponderada dos critérios de qualidade, os quais impõem algumas dificuldades na seleção de pesos adequados, a abordagem proposta nesta tese minimiza a variação percentual máxima com respeito aos valores ideais de desempenho.

A soma dos tempos de execução de todos os seis problemas de otimização foi menor que um minuto. O exemplo teve três subcaminhos $\{ep_0, ep_1, ep_2\}$ e o relatório de otimização mostrou que nenhum subcaminho violou os limites de qualidade esperados.

3.3.2 Análise Numérica

Grafos com 5, 10 e 25 nós foram gerados usando o programa Java. O número de provedores de serviços web foram respectivamente 10, 25 e 50. Os parâmetros de qualidade dos serviços web foram randomicamente variados. O parâmetro disponibilidade foi gerado com uma distribuição uniforme entre 0.95 e 0.99999. O parâmetro reputação foi gerado de forma similar com valores entre 0.8 e 0.99. O parâmetro confiabilidade foi gerado com uma distribuição uniforme entre 0.85 e 0.95.

A Tabela 3.3 apresenta variações percentuais de cada objetivo relativas aos seus valores ideais como uma função de n_{ws} , número de serviços web, e m_{task} , o número de tarefas.

Da Tabela 3.3, nota-se que para um processo de negócios com cinco tarefas, o gargalo representado por σ^* foi o tempo de execução, mesmo quando o número de provedores de serviços aumenta de 10 para 25. Para um processo de negócios com 25 tarefas, o gargalo foi o custo, mesmo quando o número de provedores varia significativamente: 10, 25 e 50.

Equilíbrio de Nash

Este capítulo apresenta soluções para o problema de programação de serviços web baseadas em jogos estratégicos não cooperativos, que têm como principal ferramenta de análise o chamado Equilíbrio de Nash.

4.1 Conceitos da Teoria de Jogos Estratégicos

Em certos tipos de **jogos de salão**, baseados em eventos probabilísticos como o lançamento de um par de dados ou o giro de uma roleta, pode-se associar um **prêmio** a cada subconjunto específico de eventos numéricos. A probabilidade de ocorrência desses eventos é totalmente conhecida, mas normalmente desfavorável aos jogadores, de modo que a organização ou entidade promotora geralmente fica com parte da quantia apostada.

Em um **jogo estratégico**, ao menos dois jogadores estão envolvidos, não existindo nenhuma organização que interaja com estes jogadores. Um jogo simples desta categoria seria o Par-ou-Ímpar, no qual a ação de um jogador consiste em escolher de um a vários dedos. Os jogadores apresentam suas estratégias simultaneamente, e então o resultado jogo é estabelecido.

Para que um jogo possa ser analisado com as ferramentas básicas da teoria de jogos estratégicos, é necessário introduzir o conceito de **utilidade**. Se o jogo envolve valores monetários, então a utilidade poderia estar associada a estes valores monetários. Porém, em jogos como o Par-ou-Ímpar, não é obvio qual seria a definição de utilidade. Uma possibilidade seria atribuir o valor 1 quando um jogador vence o jogo e o valor 0 quando perde.

Os jogos mais simples têm apenas dois jogadores e podem ser representados em uma estrutura chamada de **bimatriz**. Cada entrada desta bimatriz contém a utilidade do primeiro jogador seguida da utilidade do segundo jogador. As ações que os jogadores podem tomar são chamadas de **estratégias**, estando associadas com as linhas e as colunas da bimatriz. No exemplo da Tabela 4.1, as estratégias do jogador 1 seriam a_1, a_2, a_3 e a_4 . As estratégias do segundo jogador seriam b_1, b_2 e b_3 .

Nesta representação, a utilidade de cada jogador depende das estratégias de ambos os jogadores. Chamamos **perfil** estratégico as estratégias específicas dos jogadores justapostas. Como exemplo, um perfil estratégico da Tabela 4.1 seria (a_2, b_2) . As utilidades U_1 e U_2 seriam funções com domínio igual ao conjunto de perfis estratégicos e imagem nos reais, por exemplo.

	b_1	b_2	b_3
a_1			
a_2		$U_1(a_2, b_2), U_2(a_2, b_2)$	
a_3			
a_4			

Tabela 4.1: Bimatriz com utilidades dos jogadores.

Um jogo clássico estudado pela Teoria de Jogos é o conhecido Dilema dos Prisioneiros. Neste jogo, dois suspeitos de um crime são presos, mas não existem provas de sua culpa. A polícia interroga separadamente cada suspeito. Se ambos confessam (C) o crime, cada um recebe dois anos de prisão. Se nenhum confessar ($N - C$), ambos recebem um ano de prisão. Mas, se um confessar e o outro não, o que confessar é liberto, e o outro acusado recebe três anos de prisão. A bimatriz para este jogo é mostrada na Tabela 4.2.

	C	N-C
C	2, 2	0, 3
N-C	3, 0	1, 1

Tabela 4.2: Jogo Dilema dos Prisioneiros.

Um conceito associado à bimatriz acima, ou a qualquer outra bimatriz genérica, é a **Melhor Resposta Estratégica**. Se na Tabela 4.2, o jogador 2 escolhe a estratégia C (confessar), a melhor escolha do jogador 1 seria a estratégia C (confessar) com pena de dois anos. Usa-se marcar a melhor estratégia com símbolo “_”, como ilustrado em Tabela 4.3. Por outro lado, se o jogador 2 escolhe a estratégia $N - C$ (não confessar), então a melhor escolha do jogador 1 também seria C (confessar).

	C	N-C
C	<u>2</u> , 2	<u>0</u> , 3
N-C	3, 0	1, 1

Tabela 4.3: Melhor Resposta Estratégica do Jogador 1.

Se o jogador 1 escolhe a estratégia C (confessar), então a escolha do jogador 2 seria a estratégia C (confessar). Por outro lado, se o jogador 1 escolhe a estratégia $N - C$ (não confessar), então a melhor resposta do jogador 2 é a estratégia C (confessar), conforme apresentado na Tabela 4.4.

	C	N-C
C	2, <u>2</u>	0, 3
N-C	3, <u>0</u>	1, 1

Tabela 4.4: Melhor Resposta Estratégica do Jogador 2.

Os perfis estratégicos que estiverem simultaneamente marcados correspondem aos **Equilíbrios de Nash do jogo** (em estratégias puras), conforme Tabela 4.5. Neste exemplo, existe apenas um equilíbrio correspondendo ao perfil estratégico (C, C) , em que ambos os suspeitos

recebem dois anos de prisão. Se o perfil estratégico $(N - C, N - C)$ fosse escolhido, ambos receberiam apenas um ano de prisão. Porém, se o jogador 1 se desviar do equilíbrio de Nash escolhendo $N - C$, o jogador 2 será libertado e o jogador 1 receberá três anos de prisão. Se o jogador 2 se desviar escolhendo $N - C$, sua pena aumenta para três anos e o jogador 1 é libertado. Este seria o principal argumento a favor do equilíbrio: nenhum jogador ganha por se desviar unilateralmente do equilíbrio.

	C	N-C
C	<u>2</u> , <u>2</u>	<u>0</u> , 3
N-C	3, <u>0</u>	1, 1

Tabela 4.5: Equilíbrio de Nash para o Dilema dos Prisioneiros.

Considere agora o jogo estratégico descrito na Tabela 4.6. Neste jogo, não existe equilíbrio de Nash após a marcação com “ ” da melhor resposta estratégica de cada jogador. Neste caso dizemos que não existe equilíbrio de Nash em estratégias puras.

	b_1	b_2
a_1	<u>1</u> , -1	-1, <u>1</u>
a_2	-1, <u>1</u>	<u>1</u> , -1

Tabela 4.6: Melhor resposta estratégica.

Para jogos como o da Tabela 4.6 é possível obter um **equilíbrio de Nash em estratégias mistas**. Pode-se atribuir, por exemplo, probabilidade $\frac{1}{3}$ à estratégia a_1 e a probabilidade $\frac{2}{3}$ à estratégia a_2 . Do mesmo modo, pode-se atribuir probabilidades $\frac{1}{2}$ para as estratégias b_1 e b_2 , conforme indicado na Tabela 4.7. Os equilíbrios de Nash em estratégias puras são obtidos quando é possível atribuir probabilidade 1 a uma estratégia, e probabilidade 0 às demais estratégias.

Quando se implementa um jogo em estratégias mistas, uma interpretação possível é que cada jogador usa um dispositivo randômico para escolher uma única estratégia. Encontrar um equilíbrio de Nash em estratégias mistas significa determinar as probabilidades de cada estratégia para cada jogador.

Se o jogador 2 escolhe a estratégia b_1 com probabilidade $\frac{1}{2}$, a utilidade esperada do jogador 1 seria $\frac{1}{2} [1/3 \cdot 1 + 2/3 \cdot (-1)] = -\frac{1}{6}$, pois escolhe a_1 com probabilidade $\frac{1}{3}$ e escolhe a_2 com probabilidade $\frac{2}{3}$. Se o jogador 2 escolhe a estratégia b_2 com probabilidade $\frac{1}{2}$, a utilidade esperada do jogador 1 seria $\frac{1}{2} [1/3 \cdot (-1) + 2/3 \cdot 1] = \frac{1}{6}$. A utilidade esperada total do jogador 1 seria a soma $\frac{1}{2} [1/3 \cdot 1 + 2/3 \cdot (-1)] + \frac{1}{2} [1/3 \cdot (-1) + 2/3 \cdot 1] = 0$.

		$\frac{1}{2}$	$\frac{1}{2}$
		b_1	b_2
$\frac{1}{3}$	a_1	<u>1</u> , -1	-1, <u>1</u>
$\frac{2}{3}$	a_2	-1, <u>1</u>	<u>1</u> , -1

Tabela 4.7: Exemplo de Estratégias Mistas.

4.1.1 Determinação de Todas as Estratégias Mistas

Considere um jogo estratégico na forma de bimatriz (A, B) , onde A é a matriz de utilidades do jogador 1 e B a matriz de utilidades do jogador 2. As estratégias do jogador 1 pertencem ao conjunto $M = \{1, \dots, m\}$ e as estratégias do jogador 2 pertencem ao conjunto $N = \{1, \dots, n\}$.

Teorema 1. *Condição da Melhor Resposta.*

Sejam x e y estratégias mistas (ou seja, $x_i, y_j \in [0, 1]$ e $x_1 + \dots + x_m = 1$ e $y_1 + \dots + y_n = 1$) dos jogadores 1 e 2, respectivamente. Então x é uma melhor resposta a y se e somente se para todos $i \in M$,

$$x_i > 0 \Rightarrow (Ay)_i = u = \max_k \{(Ay)_k | k \in M\} \quad (4.1.1)$$

e y é uma melhor resposta a x se e somente se para todo $j \in N$,

$$y_j > 0 \Rightarrow (B^T x)_j = v = \max_k \{(B^T x)_k | k \in N\}. \quad (4.1.2)$$

Demonstração. Demonstração ver (Nash 1951). $\square$

A seguir, é apresentado um algoritmo para a determinação de todos os equilíbrios de Nash de um jogo com dois jogadores.

Exemplo. Algoritmo Equilíbrios de Nash por Enumeração do Suporte.

Entrada: Bimatriz de jogo: (A, B) , onde a matriz $A_{m \times n}$ contem as utilidades do jogador 1 e $B_{m \times n}$ contem as utilidades do jogador 2.

Saída: Todos os equilíbrios de Nash do jogo.

Método:(Osborne 2003) Seja I um subconjunto de M e J um subconjunto de N . Para cada escolha (I, J) , resolva as equações:

$$\sum_{i \in I} x_i b_{ij} = v, \quad \forall j \in J, \quad (4.1.3)$$

$$\sum_{i \in I} x_i = 1, \quad (4.1.4)$$

$$\sum_{j \in J} a_{ij} y_j = u, \quad \forall i \in I, \quad (4.1.5)$$

$$\sum_{j \in J} y_j = 1, \quad (4.1.6)$$

também satisfazendo as equações (4.1.1)-(4.1.2).

Note que a equação (4.1.3) pode ser rescrita como:

$$\sum_{i \in I} x_i b_{i1} = \dots = \sum_{i \in I} x_i b_{i,n}. \quad (4.1.7)$$

Da mesma forma, a equação (4.1.5) pode ser rescrita como:

$$\sum_{j \in J} a_{1j} y_j = \dots = \sum_{j \in J} a_{m,j} y_j. \quad (4.1.8)$$

Como exemplo, os equilíbrios de Nash mistos do jogo descrito na Tabela 4.8 seriam obtidos pelas equações abaixo. Note que foi usado o conceito expresso nas equações (4.1.7)-(4.1.8). Para o jogador 1,

$$y_1 u_{11} + y_2 u_{21} = y_1 u_{12} + y_2 u_{22}, \quad (4.1.9)$$

$$y_1 + y_2 = 1. \quad (4.1.10)$$

As equações para o jogador 2 seriam:

$$x_1 v_{11} + x_2 v_{12} = x_1 v_{21} + x_2 v_{22}, \quad (4.1.11)$$

$$x_1 + x_2 = 1. \quad (4.1.12)$$

		y_1	y_2
		b_1	b_2
x_1	a_1	u_{11}, v_{11}	u_{21}, v_{21}
x_2	a_2	u_{12}, v_{12}	u_{22}, v_{22}

Tabela 4.8: Exemplo genérico de estratégias mistas.

Para o exemplo específico da Tabela 4.9, valem as equações:

$$1y_1 + (-1y_2) = -1y_1 + 1y_2, \quad (4.1.13)$$

$$y_1 + y_2 = 1. \quad (4.1.14)$$

Das equações acima, obtemos $2y_1 = 2y_2$, e portanto $y_1 = y_2 = \frac{1}{2}$. Para o jogador 2,

$$-1x_1 + 1x_2 = 1x_1 + (-1x_2), \quad (4.1.15)$$

$$x_1 + x_2 = 1. \quad (4.1.16)$$

Das equações acima, obtemos $2x_2 = 2x_1$, e portanto $x_1 = x_2 = \frac{1}{2}$.

		y_1	y_2
		b_1	b_2
x_1	a_1	1, -1	-1, 1
x_2	a_2	-1, 1	1 - 1

Tabela 4.9: Cálculo do equilíbrio em estratégias mistas.

4.2 Programação de Serviços Web como Jogo Não Cooperativo

Nesta seção descreve-se um possível modelo matemático para o problema de programação de serviços web como um jogo estratégico não cooperativo. O modelo é implementado usando algoritmos especializados em Java em conjunto com um *Solver* de otimização linear-inteira.

No modelo adotado, os jogadores são os provedores de serviços web. A estratégia de cada jogador são os tempos de execução para as tarefas. Passos de discretização das estratégias (tempos de execução) são definidos a priori. Perfis estratégicos combinando as estratégias individuais são formados. A matriz de jogo com as utilidades (receitas dos jogadores) para cada perfil estratégico possível é computado e os equilíbrios de Nash para o problema de programação são obtidos.

4.3 Escolhas Estratégicas dos Jogadores

Nesta situação de conflito entre os jogadores, cada provedor de serviço atua de forma estratégica em relação aos outros provedores para conseguir ganhar o direito de executar as tarefas do fluxograma do processo de negócios, como ilustrado na Figura 4.3.1.

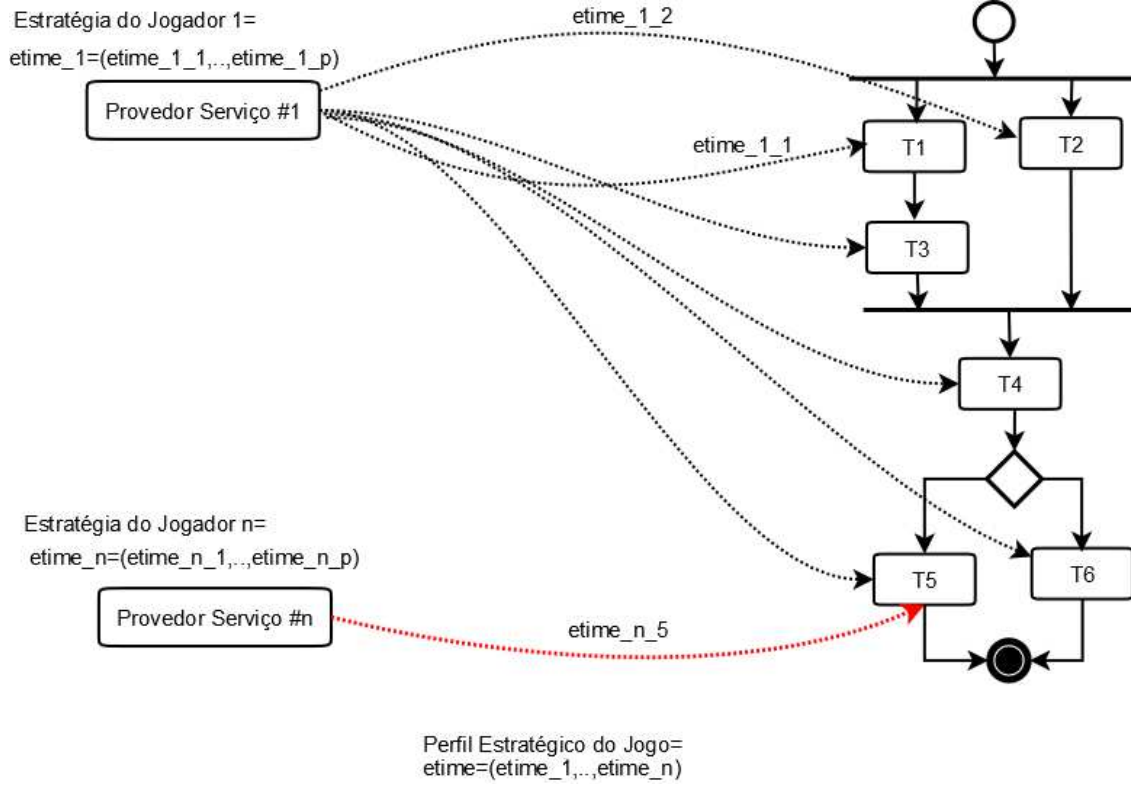


Figura 4.3.1: Programação de serviços web como jogo não cooperativo.

O tempo de execução de uma tarefa j pelo provedor k é representado pela variável ED_{jk} , que tem o seu valor determinado entre dois parâmetros $TMIN_{jk}$ e $TMAX_{jk}$:

$$TMIN_{jk} \leq ED_{jk} \leq TMAX_{jk} \quad (4.3.1)$$

Cada provedor visa maximizar a sua receita total, enquanto que o proprietário do processo de negócio visa minimizar seu custo total. O provedor k realiza a tarefa j a um preço indicado pela variável EC_{jk} , que teria uma faixa de variação entre $PMIN_{jk}$ e $PMAX_{jk}$:

$$PMIN_{jk} \leq EC_{jk} \leq PMAX_{jk} \quad (4.3.2)$$

Consideremos uma situação específica, no qual EC_{jk} é uma função linear decrescente do tempo de execução. A equação 4.3.3 representa esta relação de linearidade entre custo e tempo de execução. Note que $PMAX_{jk} > PMIN_{jk} > 0$ e $TMAX_{jk} > TMIN_{jk} > 0$ para que os custos não se tornem negativos. Uma alternativa seria expressar o custo por uma função côncava decrescente, também ilustrada na Figura 4.3.2.

$$EC_{jk} = -\frac{PMAX_{jk} - PMIN_{jk}}{TMAX_{jk} - TMIN_{jk}}ED_{jk} + PMIN_{jk} \quad (4.3.3)$$

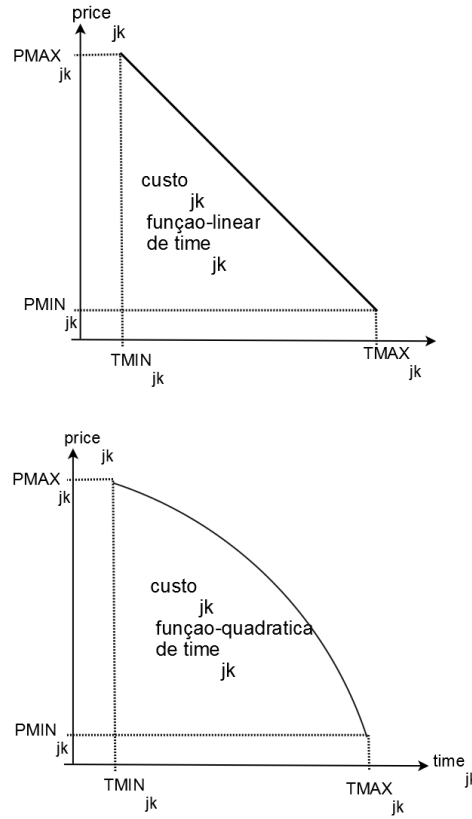


Figura 4.3.2: Funções típicas de custo por tempo de execução.

4.4 Estratégias Puras

Uma estratégia pura do jogador j é um vetor de tempos de execução para cada uma das tarefas do fluxograma, ou seja, $ED_j = (ED_{j1}, \dots, ED_{jk}, \dots, ED_{jn})$. As componentes da ED_{jk} podem assumir valores no intervalo $[TMIN_{jk}, TMAX_{jk}]$.

Para fins de simulação, cria-se uma entrada na matriz de jogo para cada possível combinação de $ED_{j1}, \dots, ED_{jp}$. Com d intervalos de tempo para cada variável ED_{jk} , então a matriz do jogo teria $(d+1)^p$ entradas apenas referentes a tarefa k do jogador j . O número $ndim$ de perfis estratégicos de um jogo com p tarefas e n jogadores é igual à $(d+1)^{pn}$. Se $d = 1$, apenas os extremos $TMIN_{jk}$ e $TMAX_{jk}$ são considerados; $d = 2$ significa que serão usados três valores: $TMIN_{jk}$, $TMAX_{jk}$ e o ponto médio entre estes valores. O perfil estratégico $(ED_1, \dots, ED_n)$ seria a justaposição de cada estratégia individual ED_j .

4.5 Utilidade dos Jogadores $U_j(ED_1, \dots, ED_n)$

A utilidade do jogador (provedor) j é dependente das suas estratégias ED_j e das estratégias dos outros jogadores. A utilidade do jogador j , $U_j(ED_1, \dots, ED_j, \dots, ED_n)$, representa a receita recebida pelo provedor para realizar as tarefas do fluxograma.

No caso de dois provedores, duas tarefas e um intervalo de tempo (apenas os valores mínimo e máximo de tempo de execução), é necessário definir os valores das utilidades $U_1(ED_1, ED_2)$ e $U_2(ED_1, ED_2)$.

Na Tabela 4.5, uma célula típica associada à bimatriz de utilidades é apresentada. Uma estra-

tégia típica do provedor 1 aparece na primeira coluna da tabela, e seria igual a $(tmax_{11}, tmin_{12})$, indicando a escolha de $tmax_{11}$ como estratégia de tempo para a primeira tarefa e $tmin_{12}$ para a estratégia de tempo para a segunda tarefa. A estratégia do provedor 2 corresponde à linha superior da tabela; a sua estratégia é $(tmax_{21}, tmin_{22})$.

	$(tmax_{21}, tmin_{22})$
$(tmax_{11}, tmin_{12})$	$U_1((tmax_{11}, tmin_{12}), (tmax_{21}, tmin_{22})),$ $U_2((tmax_{11}, tmin_{12}), (tmax_{21}, tmin_{22})),$

Tabela 4.10: Célula típica da bimatriz de utilidades.

4.6 Cálculo da Utilidade $U_j(ED_1, \dots, ED_n)$

Os valores $U_j(ED_1, \dots, ED_n)$ não são calculados individualmente, mas de forma simultânea por otimização, de acordo com a estratégia detalhada a seguir.

A receita do provedor j ao executar a tarefa k é EC_{jk} . Note que pela expressão (4.3.3), EC_{jk} é função do parâmetro ED_{jk} .

Definidas as estratégias puras, ED_j , o contratante dos provedores alocará as tarefas aos provedores de acordo com o critério de menor custo de execução de todas as tarefas, respeitadas todas as restrições do problema e que o tempo máximo de execução seja menor que MXT . Por conveniência, as principais restrições do problema serão novamente representadas.

Associando uma variável binária y_{ij} a cada tarefa i ao provedor j , temos:

$$\sum_{j \in WS} y_{ij} = 1, \quad \forall i \in A, \quad (4.6.1)$$

$$\sum_{j \in WS} ED_{ij} y_{ij} = TD_{il}, \quad \forall i \in A, \quad \forall ep_l \in L, \quad (4.6.2)$$

As restrições adicionais do modelo 0-1 são:

$$(TD_{jl} + ST_{jl}) \leq ST_{kl}, \quad \forall j \rightarrow k, \quad j, k \in A, \quad \forall ep_l \in L, \quad (4.6.3)$$

$$ED_l \geq \sum_{i \in sp_m^l} TD_{il}, \quad \forall ep_l \in L, \quad (4.6.4)$$

$$ED_l \leq MXT, \quad \forall ep_l \in L. \quad (4.6.5)$$

As restrições de custo por caminho de execução l são:

$$\sum_{i \in ep_l, j \in WS} EC_{ij} y_{ij} = EC_l, \quad \forall ep_l \in L, \quad (4.6.6)$$

$$EC_l \leq MXC, \quad \forall ep_l \in L. \quad (4.6.7)$$

Pode-se acrescentar reputação, confiabilidade e disponibilidade na forma de restrições adicionais do problema de otimização, ou mesmo considerar o problema multiobjetivo (3.2.20) para o cálculo das utilidades. A abordagem nos neste capítulos e demais foi optar por não usar a solução multiobjetivo.

O parâmetro $MNRP$ é a mínima reputação esperada, enquanto que a variável RP_l é a reputação do caminho de execução l . O parâmetro RP_{ij} representa a reputação esperada da tarefa i quando implementada pelo provedor j . Segue-se então que:

$$RP_l = \sum_{i \in ep_l, j \in WS} RP_{ij} y_{ij} / |ep_l|, \quad \forall ep_l \in L, \quad (4.6.8)$$

$$RP_l \geq MNRP, \quad \forall l, \quad (4.6.9)$$

onde $|l|$ é a número de tarefas em l .

O parâmetro MNA é a mínima disponibilidade esperada, e a variável AV_l é a disponibilidade do caminho de execução ep_l . O parâmetro AV_{ij} representa a disponibilidade esperada da tarefa i quando implementada pelo provedor j . As funções de agregação para a disponibilidade são não-lineares. Sua linearização é fornecida pelo logaritmo neperiano ($\ln$):

$$AV_l \geq \ln(MNA), \quad \forall l, \quad (4.6.10)$$

$$AV_l = \sum_{i \in ep_l, j \in WS} \ln(AV_{ij}) y_{ij}, \quad \forall ep_l \in L. \quad (4.6.11)$$

Similarmente, a restrição de confiabilidade é representada na forma:

$$R_l \geq \ln(MNRL), \quad \forall l, \quad (4.6.12)$$

$$R_l = \sum_{i \in ep_l, j \in WS} \ln(R_{ij}) y_{ij}, \quad \forall ep_l \in L. \quad (4.6.13)$$

Para encontrar a alocação ótima, função das variáveis y_{ij} , é preciso resolver o problema a seguir, no qual $freq_l$ é a frequência de execução do caminho l :

$$\begin{array}{ll} \min_{y_{ij}} & \sum_l EC_l freq_l \\ s.a & (4.6.1), \dots, (4.6.13) \end{array}$$

Com os valores de y_{ij} conhecidos, pode-se calcular as utilidades $U_1(\cdot), \dots, U_n(\cdot)$. O provedor j recebe a receita das tarefas do fluxograma que foram alocadas. A sua receita total é dada por:

$$U_j(ED_1, \dots, ED_j, \dots, ED_n) = \sum_{k=1}^p EC_{jk} y_{jk}, \quad (4.6.14)$$

onde EC_{jk} é calculado por meio de (4.3.3).

4.7 Implementação do Modelo de Otimização

A bimatriz da Tabela 4.5 poderia ser reescrita como na Tabela 4.11. Este último formato é usado pelos algoritmos na implementação. A cada linha corresponde um perfil estratégico usado na obtenção de U_1 e U_2 . A diferença entre duas linhas consecutivas é de apenas um parâmetro.

O Algoritmo .2, apresentado em Apêndice, gera todas as estratégias da Tabela 4.11. O *Solver* Gnulpk foi usado de forma que as especificações do modelo de otimização e dos dados e parâmetros estejam em arquivos distintos. Cada combinação de parâmetros produz um relatório distinto de otimização. O Algoritmo .3, também descrito em Apêndice, é responsável por este processamento.

O Algoritmo .1 (Apêndice) extrai de todos os relatórios de otimização uma matriz para a determinação dos Equilíbrios de Nash puros.

	Perfis Estratégicos
1	$tmin_{11}, tmin_{12}, tmin_{21}, tmin_{22}$
2	$tmax_{11}, tmin_{12}, tmin_{21}, tmin_{22}$
3	$tmin_{11}, tmax_{12}, tmin_{21}, tmin_{22}$
4	$tmax_{11}, tmax_{12}, tmin_{21}, tmin_{22}$
5	$tmin_{11}, tmin_{12}, tmax_{21}, tmin_{22}$
6	$tmax_{11}, tmin_{12}, tmax_{21}, tmin_{22}$
7	$tmin_{11}, tmax_{12}, tmax_{21}, tmin_{22}$
8	$tmax_{11}, tmax_{12}, tmax_{21}, tmin_{22}$
9	$tmin_{11}, tmin_{12}, tmin_{21}, tmax_{22}$
10	$tmax_{11}, tmin_{12}, tmin_{21}, tmax_{22}$
11	$tmin_{11}, tmax_{12}, tmin_{21}, tmax_{22}$
12	$tmax_{11}, tmax_{12}, tmin_{21}, tmax_{22}$
13	$tmin_{11}, tmin_{12}, tmax_{21}, tmax_{22}$
14	$tmax_{11}, tmin_{12}, tmin_{21}, tmax_{22}$
15	$tmin_{11}, tmax_{12}, tmax_{21}, tmax_{22}$
16	$tmax_{11}, tmax_{12}, tmax_{21}, tmax_{22}$

Tabela 4.11: Perfil estratégico no formato vetorial ou sequencial.

4.8 Resultados Numéricos

4.8.1 Exemplo 1

O processo de negócios representado neste exemplo contém apenas duas tarefas sequenciais e um caminho de execução $\{T1, T2\}$, sem quaisquer elementos paralelos ou que exijam decisão. A Figura 4.8.1 apresenta este exemplo muito simples.

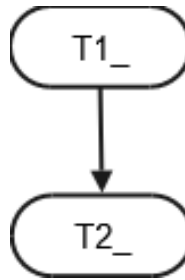


Figura 4.8.1: Processo de negócios com duas tarefas sequenciais Exemplo 1.

O tempo máximo de execução é de 85 segundos, e o parâmetro custo máximo pelo caminho de execução é de R\$ 80. Os tempos máximos e mínimos das tarefas dos dois provedores, bem como os preços mínimo e máximo para as mesmas tarefas são apresentados na Tabela 4.12. A unidade de preço é R\$, e a unidade de tempo é segundos. Por exemplo, o preço mínimo da tarefa 1 pelo provedor 1 seria R\$ 1,50, e o tempo mínimo desta mesma tarefa seria de 15 segundos. Considerou-se apenas os valores mínimos, máximos e médios de tempo e preços ($d = 2$).

	Tempo tarefa 1		Tempo tarefa 2	
	mínimo	máximo	mínimo	máximo
provedor 1	15	20	20	25
provedor 2	35	40	40	55
	Preço tarefa 1		Preço tarefa 2	
	mínimo	máximo	mínimo	máximo
provedor 1	1,50	2,00	2,00	2,50
provedor 2	3,50	4,00	1,00	5,50

Tabela 4.12: Tempos e preços de execução do Exemplo 1.

As estratégias individuais dos provedores são apresentadas na Tabela 4.13. O número de estratégias individuais é pequeno e pode ser apresentado na íntegra.

$estrategia_1$	provedor 1	$estrategia_2$	provedor 2
1	(15,0; 20,0)	1	(35,0; 40,0)
2	(17,5; 20,0)	2	(37,5; 40,0)
3	(20,0; 20,0)	3	(40,0; 40,0)
4	(15,0; 22,5)	4	(35,0; 47,5)
5	(17,5; 22,5)	5	(37,5; 47,5)
6	(20,0; 22,5)	6	(40,0; 47,5)
7	(15,0; 25,0)	7	(35,0; 55,0)
8	(17,5; 25,0)	8	(37,5; 55,0)
9	(20,0; 25,0)	9	(40,0; 55,0)

Tabela 4.13: Estratégias dos provedores do Exemplo 1.

Um relatório parcial de saída típico gerado pelo Algoritmo .1 é mostrado a seguir. Os perfis estratégicos que correspondem aos equilíbrios de Nash em estratégias puras são enumerados na Tabela 4.14. Por exemplo, o perfil estratégico ((20; 25), (37, 5; 55)) corresponde à escolha da estratégia 9 pelo provedor 1, e a estratégia 8 pelo provedor 2. Pelo conceito de teoria de jogos estratégicos, os jogadores poderiam escolher somente a partir destes perfis estratégicos. Na Tabela 4.14, as colunas “Tarefa 1” e “Tarefa 2” correspondem aos provedores escolhidos para executar as tarefas 1 e 2, respectivamente.

índices das estratégias	perfil estratégico	U_1	U_2	Tarefa 1	Tarefa 2
(1, 7)	((15,0; 20,0), (35,0; 55,0))	2,00	1,00	P1	P2
(2, 7)	((15,0; 22,5), (35,0; 55,0))	2,00	1,00	P1	P2
(3, 7)	((15,0; 25,0), (35,0; 55,0))	2,00	1,00	P1	P2
(4, 8)	((15,0; 20,0), (37,5; 55,0))	2,00	1,00	P1	P2
(4, 8)	((15,0; 22,5), (37,5; 55,0))	2,00	1,00	P1	P2
(4, 8)	((15,0; 25,0), (37,5; 55,0))	2,00	1,00	P1	P2
(7, 9)	((15,0; 20,0), (40,0; 55,0))	2,00	1,00	P1	P2
(7, 9)	((15,0; 22,5), (40,0; 55,0))	2,00	1,00	P1	P2
(7, 9)	((15,0; 25,0), (40,0; 55,0))	2,00	1,00	P1	P2

Tabela 4.14: Equilíbrios de Nash do Exemplo 1.

Observa-se que todos os equilíbrios de Nash em estratégias puras fornecem os mesmos retornos para os provedores 1 e 2. A bimatriz do jogo (que inclui os equilíbrios de Nash computados) é descrita na Tabela 4.15.

	1	2	3	4	5	6	7	8	9
1	4,50, 0,00	4,50, 0,00	4,50, 0,00	4,50, 0,00	4,50, 0,00	4,50, 0,00	2,00, 1,00	2,00, 1,00	2,00, 1,00
2	4,25, 0,00	4,25, 0,00	4,25, 0,00	4,25, 0,00	4,25, 0,00	4,25, 0,00	1,75, 1,00	1,75, 1,00	1,75, 1,00
3	4,00, 0,00	4,00, 0,00	4,00, 0,00	4,00, 0,00	4,00, 0,00	4,00, 0,00	1,50, 1,00	1,50, 1,00	1,50, 1,00
4	4,25, 0,00	4,25, 0,00	4,25, 0,00	4,25, 0,00	4,25, 0,00	4,25, 0,00	2,00, 1,00	2,00, 1,00	2,00, 1,00
5	4,00, 0,00	4,00, 0,00	4,00, 0,00	4,00, 0,00	4,00, 0,00	4,00, 0,00	1,75, 1,00	1,75, 1,00	1,75, 1,00
6	3,75, 0,00	3,75, 0,00	3,75, 0,00	3,75, 0,00	3,75, 0,00	3,75, 0,00	1,50, 1,00	1,50, 1,00	1,50, 1,00
7	4,00, 0,00	4,00, 0,00	4,00, 0,00	4,00, 0,00	4,00, 0,00	4,00, 0,00	2,00, 1,00	2,00, 1,00	2,00, 1,00
8	3,75, 0,00	3,75, 0,00	3,75, 0,00	3,75, 0,00	3,75, 0,00	3,75, 0,00	1,75, 1,00	1,75, 1,00	1,75, 1,00
9	3,50, 0,00	3,50, 0,00	3,50, 0,00	3,50, 0,00	3,50, 0,00	3,50, 0,00	1,50, 1,00	1,50, 1,00	1,50, 1,00

Tabela 4.15: Bimatriz associada ao Exemplo 1.

4.8.2 Exemplo 2

Novos tempos e custos de execução são apresentados na Tabela 4.16. Os custos de execução foram alterados para valores mais competitivos, de forma que agora não é obvio qual provedor ganha o direito de executar as tarefas. Os parâmetros tempo máximo de execução no caminho de execução é de 85 segundos; o custo máximo no caminho de execução é de R\$ 80.

	Tempo tarefa 1		Tempo tarefa 2	
	mínimo	máximo	mínimo	máximo
provedor 1	15	20	20	25
provedor 2	13	19	21	24

	Preço tarefa 1		Preço tarefa 2	
	mínimo	máximo	mínimo	máximo
provedor 1	1,50	2,00	1,80	2,25
provedor 2	1,75	1,90	1,70	2,30

Tabela 4.16: Tempos e custos de execução do Exemplo 2.

Todas as nove estratégias dos provedores são mostradas na Tabela 4.17.

$estrategia_1$	provedor 1	$estrategia_2$	provedor 2
1	(15,0; 20,0)	1	(13,0; 21,0)
2	(17,5; 20,0)	2	(16,0; 21,0)
3	(20,0; 20,0)	3	(19,0; 21,0)
4	(15,0; 22,5)	4	(13,0; 22,5)
5	(17,5; 22,5)	5	(16,0; 22,5)
6	(20,0; 22,5)	6	(19,0; 22,5)
7	(15,0; 25,0)	7	(13,0; 24,0)
8	(17,5; 25,0)	8	(16,0; 24,0)
9	(20,0; 25,0)	9	(19,0; 24,0)

Tabela 4.17: Estratégias provedores 1 e 2 do Exemplo 2.

O relatório de otimização produzido pelo *Solver* apresenta apenas três equilíbrios de Nash em estratégias puras, indicados na Tabela 4.18. Por exemplo, o perfil estratégico $((17, 5; 25, 0); (19, 0; 24, 0))$ corresponde à escolha da estratégia 8 pelo provedor 1 e da estratégia 9 pelo provedor 2.

índice das estratégias	perfil estratégico	U_1	U_2	Tarefa 1	Tarefa 2
(8, 7)	$((17,5; 25,0), (13,0; 24,0))$	1,75	1,70	P1	P2
(8, 8)	$((17,5; 25,0), (16,0; 24,0))$	1,75	1,70	P1	P2
(8, 9)	$((17,5; 25,0), (19,0; 24,0))$	1,75	1,70	P1	P2

Tabela 4.18: Equilíbrios de Nash do Exemplo 2.

Novamente, todos os equilíbrios de Nash forneceram as mesmas utilidades para os provedores 1 e 2. O resultado da Tabela 4.18 está representado na bimatriz do jogo descrita na Tabela 4.19.

	1	2	3	4	5	6	7	8	9
1	2,25, 1,90	2,25, 1,83	2,25, 1,75	0,00, 3,90	0,00, 3,83	0,00, 3,75	0,00, 3,60	0,00, 3,53	0,00, 3,45
2	4,00, 0,00	4,00, 0,00	4,00, 0,00	1,75, 2,00	1,75, 2,00	1,75, 2,00	1,75, 1,70	1,75, 1,70	1,75, 1,70
3	3,75, 0,00	3,75, 0,00	3,75, 0,00	1,50, 2,00	1,50, 2,00	1,50, 2,00	1,50, 1,70	1,50, 1,70	1,50, 1,70
4	2,02, 1,90	2,02, 1,83	2,02, 1,75	0,00, 3,90	0,00, 3,83	0,00, 3,75	0,00, 3,60	0,00, 3,53	0,00, 3,45
5	3,77, 0,00	3,77, 0,00	3,77, 0,00	1,75, 2,00	1,75, 2,00	1,75, 2,00	1,75, 1,70	1,75, 1,70	1,75, 1,70
6	3,52, 0,00	3,52, 0,00	3,52, 0,00	1,50, 2,00	1,50, 2,00	1,50, 2,00	1,50, 1,70	1,50, 1,70	1,50, 1,70
7	1,80, 1,90	1,80, 1,83	1,80, 1,75	1,80, 1,90	1,80, 1,83	1,80, 1,75	0,00, 3,60	0,00, 3,53	0,00, 3,45
8	3,55, 0,00	3,55, 0,00	3,55, 0,00	3,55, 0,00	3,55, 0,00	3,55, 0,00	1,75, 1,70	1,75, 1,70	1,75, 1,70
9	3,30, 0,00	3,30, 0,00	3,30, 0,00	3,30, 0,00	3,30, 0,00	3,30, 0,00	1,50, 1,70	1,50, 1,70	1,50, 1,70

Tabela 4.19: Bimatriz associada ao Exemplo 2.

4.8.3 Exemplo 3

Neste exemplo, descrito na Figura 4.8.2, o processo de negócios possui cinco tarefas não sequenciais, ou seja, existe uma decisão, representada pelo losango. Existem dois caminhos de execução: $T1 \rightarrow T2 \rightarrow T3 \rightarrow T5$ e $T1 \rightarrow T2 \rightarrow T4 \rightarrow T5$, porém nenhuma tarefa paralela. Cada caminho de execução tem 50% de frequência de execução. O tempo máximo de execução em qualquer caminho é de 80 segundos; o custo máximo por caminho de execução é de R\$ 85.

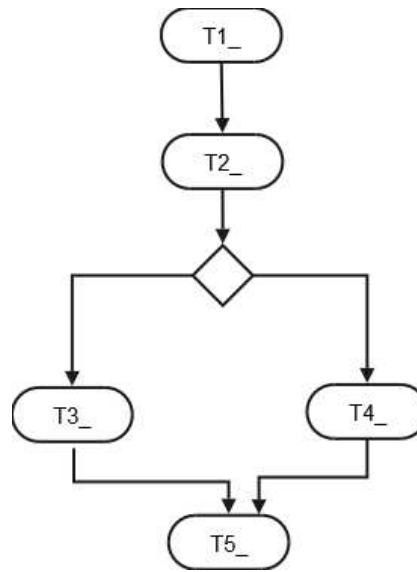


Figura 4.8.2: Processo de negócios com cinco tarefas do Exemplo 3.

A Tabela 4.20 exibe os tempos de execução (unidade de tempo em segundos) mínimos e máximos de cada provedor, para cada uma das cinco tarefas. Também apresenta os preços de execução (unidade monetária em R\$) mínimos e máximos dos dois provedores, para cada uma das cinco tarefas. Por exemplo, o preço mínimo da tarefa 5 pelo provedor 2 seria R\$ 0,05, e o preço máximo da tarefa 2 pelo provedor 1 seria R\$ 2,00. O tempo mínimo de execução da tarefa 5 pelo provedor 2 seria 1 segundo; o tempo máximo de execução da tarefa 1 pelo provedor 1 seria 20 segundos.

Tempos	tarefa 1		tarefa 2		tarefa 3		tarefa 4		tarefa 5	
	mín	máx	mín	máx	mín	máx	mín	máx	mín	máx
provedor 1	15	20	14	19	15	16	16	18	6	8
provedor 2	14	21	13	18	16	18	10	11	1	5
Preços	tarefa 1		tarefa 2		tarefa 3		tarefa 4		tarefa 5	
	mín	máx	mín	máx	mín	máx	mín	máx	mín	máx
provedor 1	1,50	2,50	0,20	2,00	0,85	1,25	0,02	1,00	0,33	0,66
provedor 2	1,30	1,50	1,70	1,80	0,75	1,22	0,25	0,75	0,05	0,25

Tabela 4.20: Tempos e preços de execução do Exemplo 3.

Com a discretização proposta de três pontos de avaliação (mínimo, máximo e médio), a Tabela 4.21 apresenta uma listagem parcial das 243 estratégias de cada provedor. Cada estratégia corresponde a uma tupla com cinco valores de tempos de execução, um para cada tarefa do processo de negócios.

$estrategia_1$	provedor 1	$estrategia_2$	provedor 2
1	(15,0; 14,0; 15,0; 16,0; 6,0)	1	(14,0; 13,0; 16,0; 10,0; 1,0)
2	(17,5; 14,0; 15,0; 16,0; 6,0)	2	(17,5; 13,0; 16,0; 10,0; 1,0)
3	(20,0; 14,0; 15,0; 16,0; 6,0)	3	(21,0; 13,0; 16,0; 10,0; 1,0)
$\vdots$	$\vdots$	$\vdots$	$\vdots$
241	(15,0; 19,0; 16,0; 18,0; 8,0)	241	(14,0; 18,0; 18,0; 11,0; 5,0)
242	(15,0; 19,0; 16,0; 18,0; 8,0)	242	(17,5; 18,0; 18,0; 11,0; 5,0)
243	(15,0; 19,0; 16,0; 18,0; 8,0)	243	(21,0; 18,0; 18,0; 11,0; 5,0)

Tabela 4.21: Estratégias dos provedores 1 e 2 do Exemplo 3.

A Tabela 4.22 apresenta os 17 equilíbrios de Nash em estratégias puras encontrados. O valor da coluna “índices das estratégias”, conforme produzido pelo relatório do *Solver*, é igual a $(estrategia_1; estrategia_2)$, onde $estrategia_1$ e $estrategia_2$ representam os índices das estratégias da Tabela 4.21. As colunas “T1”, “T2”, “T3”, “T4” e “T5” indicam qual provedor executa qual tarefa. Neste exemplo, cada equilíbrio de Nash sempre seria executado da seguinte maneira: o provedor 1 executa as tarefas T2 e T4, enquanto que o provedor 2 executa as tarefas T1, T3 e T5.

Eventualmente, o problema de programação se torna infactível devido ao fato do perfil estratégico ser incompatível com o tempo máximo de execução. Nesta condição, o *Solver* conclui que não houve solução e retorna um valor negativo adequado para todas as utilidades dos provedores. Nenhum equilíbrio de Nash pode ser obtido sem que todas as soluções sejam factíveis.

4.9 Resultados pelo Método de Avis-Rosenberg-Savani-Stengel (ARSS)

Nesta seção, apresentamos os resultados obtidos pelo algoritmo proposto em (Avis, Rosenberg, Savani & von Stengel 2010), para os mesmos exemplos da seção anterior. Este algoritmo usa o método *lrsNash* baseado em um algoritmo de enumeração de vértices *lrs* (*lexographic reverse search* ou busca lexicográfica reversa). O algoritmo enumera os vértices de um politopo de melhores respostas.

O algoritmo calcula os equilíbrios de Nash tanto em estratégias puras quanto em estratégias mistas. Os elementos mostrados abaixo das colunas s_{ij} nas Tabela 4.23, Tabela 4.24, Tabela 4.27, Tabela 4.29, Tabela 4.31, Tabela 4.34 e Tabela 4.38 são probabilidades associadas às estratégias dos jogadores.

As colunas indicadas por s_{ij} se referem à estratégia i do jogador j . Por exemplo, s_{91} é a nona estratégia do jogador 1.

Cada Equilíbrio de Nash é indicado pela coluna EE da tabela. A utilidade de cada jogador j é indicada na coluna EP_j .

4.9.1 Uma nota sobre a complexidade do cálculo do Equilíbrio de Nash

Segundo (Papadimitriou 2007), a estrutura do problema do Equilíbrio de Nash é diferente do da estrutura do problema clássico da satisfabilidade de expressões booleanas, no qual deve-se

	índices das estratégias	perfil estratégico	U_1	U_2	T_1	T_2	T_3	T_4	T_5
1	(77,45)	((20,0; 16,5; 16,0; 18,0; 6,0), (17,5; 13,0; 18,0; 10,5; 1,0))	1,12	2,40	P2	P1	P2	P1	P2
2	(162,45.)	((20,0; 16,5; 16,0; 18,0; 7,0), (17,5; 13,0; 18,0; 10,5; 1,0))	1,12	2,40	P2	P1	P2	P1	P2
3	(240,45)	((20,0; 16,5; 16,0; 18,0; 8,0), (17,5; 13,0; 18,0; 10,5; 1,0))	1,12	2,40	P2	P1	P2	P1	P2
4	(77,48)	((20,0; 16,5; 16,0; 18,0; 6,0), (17,5; 15,5; 18,0; 10,5; 1,0))	1,12	2,40	P2	P1	P2	P1	P2
5	(162,48)	((20,0; 16,5; 16,0; 18,0; 7,0), (17,5; 15,5; 18,0; 10,5; 1,0))	1,12	2,40	P2	P1	P2	P1	P2
6	(240,48)	((20,0; 16,5; 16,0; 18,0; 8,0), (17,5; 15,5; 18,0; 10,5; 1,0))	1,12	2,40	P2	P1	P2	P1	P2
7	(77,51)	((20,0; 16,5; 16,0; 18,0; 6,0), (17,5; 18,0; 18,0; 10,5; 1,0))	1,12	2,40	P2	P1	P2	P1	P2
8	(162,51)	((20,0; 16,5; 16,0; 18,0; 7,0), (17,5; 18,0; 18,0; 10,5; 1,0))	1,12	2,40	P2	P1	P2	P1	P2
9	(240,51)	((20,0; 16,5; 16,0; 18,0; 8,0), (17,5; 18,0; 18,0; 10,5; 1,0))	1,12	2,40	P2	P1	P2	P1	P2
10	(77,72)	((20,0; 16,5; 16,0; 18,0; 6,0), (17,5; 13,0; 18,0; 11,0; 1,0))	1,12	2,40	P2	P1	P2	P1	P2
11	(162,72)	((20,0; 16,5; 16,0; 18,0; 7,0), (17,5; 13,0; 18,0; 11,0; 1,0))	1,12	2,40	P2	P1	P2	P1	P2
12	(240,72)	((20,0; 16,5; 16,0; 18,0; 8,0), (17,5; 13,0; 18,0; 11,0; 1,0))	1,12	2,40	P2	P1	P2	P1	P2
13	(77,75)	((20,0; 16,5; 16,0; 18,0; 6,0), (17,5; 15,5; 18,0; 11,0; 1,0))	1,12	2,40	P2	P1	P2	P1	P2
14	(162,75)	((20,0; 16,5; 16,0; 18,0; 7,0), (17,5; 15,5; 18,0; 11,0; 1,0))	1,12	2,40	P2	P1	P2	P1	P2
14	(240,75)	((20,0; 16,5; 16,0; 18,0; 8,0), (17,5; 15,5; 18,0; 11,0; 1,0))	1,12	2,40	P2	P1	P2	P1	P2
15	(77,78)	((20,0; 16,5; 16,0; 18,0; 6,0), (17,5; 18,0; 18,0; 11,0; 1,0))	1,12	2,40	P2	P1	P2	P1	P2
16	(162,77)	((20,0; 16,5; 16,0; 18,0; 7,0), (17,5; 18,0; 18,0; 11,0; 1,0))	1,12	2,40	P2	P1	P2	P1	P2
17	(240,77)	((20,0; 16,5; 16,0; 18,0; 8,0), (17,5; 18,0; 18,0; 11,0; 1,0))	1,12	2,40	P2	do	P2	P1	P2

Tabela 4.22: Equilíbrios de Nash do Exemplo 3.

determinar se existe uma alocação às variáveis lógicas desta expressão que a tornem verdadeira, ou seja, não se garante que tal solução verdadeira logicamente exista.

Ao contrário, para o Equilíbrio de Nash sempre existirá uma solução em estratégias mistas. Então (Papadimitriou 2007) propõe o conceito de intratabilidade chamado de PPAD-completo para explicar o problema da complexidade de algoritmos de determinação do Equilíbrio de Nash em estratégias mistas. Um problema PPAD estaria em termos de complexidade entre os problemas P e NP clássicos, ou seja, Polinomial e Não-Polinomial, respectivamente. Esta classe de algoritmos se supõe aplicar-se em caminhos dirigidos em grafos.

4.9.2 Exemplo 1

O algoritmo encontrou nove Equilíbrios de Nash em estratégias puras. Conforme pode ser observado da Tabela 4.23, existem apenas valores 1.0 (indicando 100% de chance de executar uma estratégia) ou 0.0 (indicando nunca executar esta tarefa).

Os valores das utilidades em R\$ obtidas na Tabela 4.23, 2,00 e 1,00, correspondem exatamente aos valores das utilidades U_1 e U_2 obtidos na Tabela 4.14.

Para o jogador 1, temos a seguinte situação: a estratégia s_{11} nos equilíbrios 1, 2 e 3 na Tabela 4.23 corresponde à estratégia (15.0, 20.0), ou seja, $indice_1 = 1$ da Tabela 4.13; a estratégia s_{41} nos equilíbrios 3, 4 e 5 na Tabela 4.23 corresponde à estratégia (15.0, 22.5), ou seja, $indice_1 = 4$ da Tabela 4.13; e finalmente, a estratégia s_{71} nos equilíbrios 7, 8 e 9 na Tabela 4.23 corresponde à estratégia (15.0, 25.0), ou seja, $indice_1 = 7$ da Tabela 4.13.

Para o jogador 2, temos a outra situação: a estratégia s_{72} nos equilíbrios 1, 4 e 7 da Tabela 4.23 corresponde à estratégia (35.0, 55.0), ou seja, $indice_2 = 7$ da Tabela 4.23; a estratégia s_{82} nos equilíbrios 2.5 e 8 da Tabela 4.23 corresponde à estratégia (37.5, 55.0), ou seja, $indice_2 = 8$ da Tabela 4.23; e finalmente, a estratégia s_{92} nos equilíbrios 3, 6 e 9 da Tabela 4.23 corresponde à estratégia (40.0, 55.0), ou seja, $indice_2 = 9$ da Tabela 4.23.

EE	s_{11}	s_{21}	s_{31}	s_{41}	s_{51}	s_{61}	s_{71}	s_{81}	s_{91}	EP_1
1	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2,00
2	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2,00
3	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2,00
4	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	2,00
5	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	2,00
6	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	2,00
7	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	2,00
8	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	2,00
9	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	2,00
EE	s_{12}	s_{22}	s_{32}	s_{42}	s_{52}	s_{62}	s_{72}	s_{82}	s_{92}	EP_2
1	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1,00
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1,00
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1,00
4	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1,00
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1,00
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1,00
7	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1,00
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1,00
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1,00

Tabela 4.23: Exemplo 1 resolvido pelo método ARSS.

4.9.3 Exemplo 2

No primeiro equilíbrio de Nash da Tabela 4.24, o provedor 1 escolhe entre as estratégias 2 e 8 com probabilidades 0.85 e 0.15, respectivamente, ao passo que o provedor 2 escolhe a estratégia 7 com probabilidade 1.0. No nono equilíbrio de Nash, o provedor 1 escolhe a estratégia pura 8, e o provedor 2 escolhe a estratégia pura 9. As utilidades em R\$ dos provedores 1 e 2, em qualquer equilíbrio, são sempre, respectivamente, 1,75 e 1,70. As estratégias puras obtidas pelo método de ARSS são as mesmas obtidas anteriormente.

EE	s_{11}	s_{21}	s_{31}	s_{41}	s_{51}	s_{61}	s_{71}	s_{81}	s_{91}	EP_1
1	0.0	0.85	0.0	0.0	0.0	0.0	0.0	0.15	0.0	1,75
2	0.0	0.85	0.0	0.0	0.0	0.0	0.0	0.15	0.0	1,75
3	0.0	0.85	0.0	0.0	0.0	0.0	0.0	0.15	0.0	1,75
4	0.0	0.0	0.0	0.0	0.85	0.0	0.0	0.15	0.0	1,75
5	0.0	0.0	0.0	0.0	0.85	0.0	0.0	0.15	0.0	1,75
6	0.0	0.0	0.0	0.0	0.85	0.0	0.0	0.15	0.0	1,75
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1,75
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1,75
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1,75
EE	s_{12}	s_{22}	s_{32}	s_{42}	s_{52}	s_{62}	s_{72}	s_{82}	s_{92}	EP_2
1	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1,70
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1,70
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1,70
4	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1,70
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1,70
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1,70
7	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1,70
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1,70
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1,70

Tabela 4.24: Exemplo 2 resolvido pelo método ARSS.

4.9.4 Exemplo 3

A utilização do método de ARSS no Exemplo 3 exigiria a especificação e apresentação de bimatrizes de dimensões 243×243 . Para reduzir a dimensionalidade do problema, extraiu-se aleatoriamente um subconjunto de 10 estratégias. Isto diminui o espaço de pesquisa para apenas 100 perfis estratégicos. A Tabela 4.25 apresenta uma possível realização: uma estratégia do provedor 1 é usar a estratégia 224 original; uma estratégia do provedor 2 é usar a estratégia 172 original, e assim em diante.

índice estratégia original prov. 1	provedor 1	índice estratégia original prov. 2	provedor 2
224	(17,5; 19,0; 15,0; 18,0; 8,0)	172	(14,0; 13,0; 17,0; 10,0; 5,0)
233	(17,5; 19,0; 15,5; 18,0; 8,0)	66	(21,0; 13,0; 17,0; 11,0; 1,0)
100	(15,0; 14,0; 16,0; 16,0; 7,0)	109	(14,0; 13,0; 16,0; 10,5; 3,0)
186	(20,0; 16,5; 16,0; 16,0; 8,0)	210	(21,0; 13,0; 18,0; 10,5; 5,0)
102	(20,0; 14,0; 16,0; 16,0; 7,0)	65	(17,5; 13,0; 17,0; 11,0; 1,0)
237	(20,0; 14,0; 16,0; 18,0; 8,0)	137	(17,5; 13,0; 16,0; 11,0; 3,0)
116	(17,5; 19,0; 15,0; 17,0; 7,0)	131	(17,5; 15,5; 18,0; 10,5; 3,0)
88	(15,0; 19,0; 15,0; 16,0; 7,0)	201	(21,0; 13,0; 17,0; 10,5; 5,0)
157	(15,0; 16,5; 16,0; 18,0; 7,0)	19	(14,0; 13,0; 18,0; 10,0; 1,0)
14	(17,5; 16,5; 15,5; 16,0; 6,0)	119	(17,5; 13,0; 17,0; 10,5; 3,0)

Tabela 4.25: Estratégias reduzidas do Exemplo 3.

Para este conjunto reduzido de estratégias, o *Solver* foi aplicado a cada perfil estratégico para obter as utilidades de cada provedor, conforme Tabela 4.26. Os números na primeira coluna (linha) da Tabela 4.26 representam índices das estratégias dos provedores.

Aplicando-se o método ARSS, obtemos a solução descrita na Tabela 4.27. Não existe solução em estratégias puras, mas apenas dois equilíbrios de Nash em estratégias mistas. Considerando o único equilíbrio de Nash, o provedor 1 deve usar 15% das vezes a estratégia $s_{41} = 186$ e 85% das vezes $s_{91} = 157$; o provedor 2 deve usar 99% das vezes as estratégias $s_{72} = 131$ e 1% das vezes $s_{92} = 19$. As utilidades para os provedores 1 e 2 foram R\$ 1,11 e R\$ 1,95, respectivamente.

EE	s_{11}	s_{21}	s_{31}	s_{41}	s_{51}	s_{61}	s_{71}	s_{81}	s_{91}	s_{101}	EP_1
1	0.0	0.0	0.0	0.15	0.0	0.0	0.0	0.0	0.85	0.0	1,11
EE	s_{12}	s_{22}	s_{32}	s_{42}	s_{52}	s_{62}	s_{72}	s_{82}	s_{92}	s_{102}	EP_2
1	0.0	0.0	0.0	0.0	0.0	0.0	0.99	0.0	0.01	0.0	1,95

Tabela 4.27: Exemplo 3 resolvido pelo método ARSS.

Os exemplos a seguir têm por finalidade analisar a influência da frequência dos caminhos de execução sobre os equilíbrios de Nash do jogo.

4.9.5 Exemplo 3A

Consideremos frequência de 20% para o caminho de execução $T1 \rightarrow T2 \rightarrow T3 \rightarrow T5$ e frequência de 80% para o caminho de execução $T1 \rightarrow T2 \rightarrow T4 \rightarrow T5$. A bimatriz contendo as utilidades referentes ao Exemplo 3A é apresentada na Tabela 4.28. Comparando-se as Tabelas 4.27 e 4.29, observa-se que o equilíbrio de Nash se manteve inalterado quando as frequências variaram de 50%-50% para 20%-80%. Observa-se também que embora o equilíbrio não tenha sido alterado, as utilidades variaram de R\$ 1,11 para R\$ 1,12, no caso do provedor 1, e de R\$ 1,95 para R\$ 1,76, no caso do provedor 2.

EE	s_{11}	s_{21}	s_{31}	s_{41}	s_{51}	s_{61}	s_{71}	s_{81}	s_{91}	s_{101}	EP_1
1	0.0	0.0	0.0	0.15	0.0	0.0	0.0	0.0	0.85	0.0	1,12
EE	s_{12}	s_{22}	s_{32}	s_{42}	s_{52}	s_{62}	s_{72}	s_{82}	s_{92}	s_{101}	EP_2
1	0.0	0.0	0.0	0.0	0.0	0.0	0.99	0.0	0.01	0.0	1,76

Tabela 4.29: Exemplo 3A resolvido pelo método ARSS.

4.9.6 Exemplo 3B

Alternativamente, consideremos frequência de 80% para o caminho de execução $T1 \rightarrow T2 \rightarrow T3 \rightarrow T5$ e frequência de 20% para o caminho de execução $T1 \rightarrow T2 \rightarrow T4 \rightarrow T5$. A bimatriz de utilidades referentes ao Exemplo 3B é apresentada na Tabela 4.30. Comparando-se as Tabelas 4.27 e 4.31, observa-se o surgimento de mais três equilíbrios de Nash em estratégias mistas. Nos novos equilíbrios, a utilidade do provedor 1 decresce ligeiramente, mas as utilidades do provedor 2 aumentam significativamente.

	172	66	109	210	65	137	131	201	19	119
224	0,21, 2,04	0,21, 2,04	0,21, 2,26	0,21, 1,73	0,21, 2,14	0,21, 2,16	0,21, 1,92	0,21, 1,84	0,21, 2,13	0,21, 2,04
233	0,21, 2,04	0,21, 2,04	0,73, 1,65	0,21, 1,73	0,21, 2,14	0,73, 1,55	0,21, 1,92	0,21, 1,84	0,21, 2,13	0,21, 2,04
100	0,42, 3,73	0,42, 3,48	0,42, 3,70	0,00, 3,78	0,42, 3,58	0,42, 3,48	0,00, 3,93	0,42, 3,40	0,00, 4,30	0,42, 3,60
186	3,03, 0,42	1,53, 1,68	3,03, 0,40	1,10, 1,98	1,53, 1,77	1,53, 1,67	1,10, 2,17	1,53, 1,60	2,60, 1,00	1,53, 1,00
102	1,92, 2,23	0,42, 3,48	1,92, 2,20	0,00, 3,78	0,42, 3,58	0,42, 3,48	0,00, 3,93	0,42, 3,40	1,50, 2,80	0,42, 3,60
237	1,93, 1,85	0,43, 3,35	1,93, 1,95	0,01, 3,53	0,43, 3,45	0,43, 3,35	0,01, 3,68	0,43, 3,15	1,51, 2,43	0,43, 3,35
116	0,46, 2,04	0,20, 2,17	0,20, 2,51	0,20, 1,98	0,20, 2,27	0,20, 2,28	0,20, 2,17	0,20, 2,09	0,46, 2,13	0,20, 2,29
88	0,20, 2,42	0,20, 2,17	0,20, 2,51	0,20, 1,98	0,20, 2,27	0,20, 2,28	0,20, 2,17	0,20, 2,09	0,20, 2,50	0,20, 2,29
157	1,54, 1,55	1,54, 1,55	1,54, 1,65	1,11, 1,73	1,54, 1,65	1,54, 1,55	1,11, 1,92	1,54, 1,35	1,11, 2,13	1,54, 1,55
14	1,10, 2,42	1,10, 2,17	1,63, 1,90	1,10, 1,98	1,10, 2,27	1,63, 1,67	1,10, 2,17	1,10, 2,09	1,10, 2,50	1,10, 2,29

Tabela 4.26: Bimatriz de utilidades para estratégias reduzidas do Exemplo 3.

	172	66	109	210	65	137	131	201	19	119
224	0,22, 1,75	0,22, 1,75	0,22, 1,89	0,22, 1,50	0,22, 1,85	0,22, 1,79	0,22, 1,70	0,22, 1,55	0,22, 1,90	0,22, 1,75
233	0,22, 1,75	0,22, 1,75	0,43, 1,65	0,22, 1,50	0,22, 1,85	0,43, 1,55	0,22, 1,70	0,22, 1,55	0,22, 1,90	0,22, 1,75
100	0,17, 3,95	0,17, 3,55	0,17, 3,85	0,00, 3,70	0,17, 3,65	0,17, 3,55	0,00, 3,85	0,17, 3,55	0,00, 4,30	0,17, 3,75
186	2,77, 0,65	1,27, 1,75	2,77, 0,55	1,10, 1,90	1,27, 1,85	1,10, 1,75	1,27, 2,10	1,27, 1,75	2,60, 1,00	1,27, 1,95
102	1,67, 2,45	0,17, 3,55	1,67, 2,35	0,00, 3,70	0,17, 3,65	0,17, 3,55	0,00, 3,85	0,17, 3,55	1,50, 2,80	0,17, 3,75
237	1,69, 1,85	0,19, 3,85	1,69, 1,95	0,02, 3,30	0,19, 3,45	0,19, 3,35	0,02, 3,45	0,19, 3,15	1,52, 2,20	0,19, 3,35
116	0,61, 1,75	0,20, 1,95	0,20, 2,29	0,20, 1,90	0,20, 2,05	0,20, 1,99	0,20, 2,10	0,20, 1,95	0,61, 1,90	0,20, 2,15
88	0,20, 2,35	0,20, 1,95	0,20, 2,29	0,20, 1,90	0,20, 2,05	0,20, 1,99	0,20, 2,10	0,20, 1,95	0,20, 2,50	0,20, 2,15
157	1,29, 1,55	1,29, 1,55	1,29, 1,65	1,12, 1,50	1,29, 1,65	1,29, 1,55	1,12, 1,70	1,29, 1,35	1,12, 1,90	1,29, 1,55
14	1,10, 2,35	1,10, 1,95	1,31, 2,05	1,10, 1,90	1,10, 2,05	1,31, 1,75	1,10, 2,10	1,10, 1,95	1,10, 2,50	1,10, 2,15

Tabela 4.28: Bimatriz de utilidades para estratégias reduzidas do Exemplo 3A.

	172	66	109	210	65	137	131	201	19	119
224	0,20, 2,34	0,20, 2,34	0,20, 2,63	0,20, 1,95	0,20, 2,44	0,20, 2,53	0,20, 2,15	0,20, 2,14	0,20, 2,35	0,20, 2,34
233	0,20, 2,34	0,20, 2,34	1,04, 1,65	0,20, 1,95	0,20, 2,44	1,04, 1,55	0,20, 2,15	0,20, 2,14	0,20, 2,35	0,20, 2,34
100	0,68, 3,50	0,68, 3,40	0,68, 3,55	0,00, 3,85	0,68, 3,50	0,68, 3,40	0,00, 4,00	0,68, 3,25	0,00, 4,30	0,68, 3,45
186	3,28, 0,20	1,78, 1,60	3,28, 0,25	1,10, 2,05	1,78, 1,70	1,78, 1,60	1,10, 2,25	1,78, 1,45	2,60, 1,00	1,78, 1,65
102	2,18, 2,00	0,68, 3,40	2,18, 2,05	0,00, 3,85	0,68, 3,50	0,68, 3,40	0,00, 4,00	0,68, 3,25	1,50, 2,80	0,68, 3,45
237	2,18, 1,85	0,68, 3,35	2,18, 1,95	0,00, 3,75	0,68, 3,45	0,68, 3,35	0,00, 3,90	0,68, 3,15	1,50, 2,65	0,68, 3,35
116	0,30, 2,34	0,20, 2,39	0,20, 2,73	0,20, 1,90	0,20, 2,49	0,20, 2,58	0,20, 2,25	0,20, 2,24	0,30, 2,35	0,20, 2,44
88	0,20, 2,49	0,20, 2,39	0,20, 2,73	0,20, 2,05	0,20, 2,49	0,20, 2,58	0,20, 2,25	0,20, 2,24	0,20, 2,50	0,20, 2,44
157	1,78, 1,55	1,78, 1,55	1,78, 1,65	1,10, 1,95	1,78, 1,65	1,78, 1,55	1,10, 2,15	1,78, 1,35	1,10, 2,35	1,78, 1,55
14	1,10, 2,49	1,10, 2,39	1,94, 1,75	1,10, 2,05	1,10, 2,49	1,94, 1,60	1,10, 2,25	1,10, 2,24	1,10, 2,50	1,10, 2,44

Tabela 4.30: Bimatriz de Utilidades para Estratégias Reduzidas do Exemplo 3B.

EE	s_{11}	s_{21}	s_{31}	s_{41}	s_{51}	s_{61}	s_{71}	s_{81}	s_{91}	s_{101}	EP_1
1	0.0	0.0	0.0	0.14	0.0	0.0	0.0	0.0	0.86	0.0	1,10
2	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1,10
3	0.0	0.0	0.0	0.16	0.0	0.0	0.0	0.0	0.15	0.69	1,10
4	0.0	0.0	0.0	0.30	0.0	0.0	0.0	0.0	0.0	0.70	1,10
EE	s_{12}	s_{22}	s_{32}	s_{42}	s_{52}	s_{62}	s_{72}	s_{82}	s_{92}	s_{101}	EP_2
1	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	2,16
2	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	2,25
3	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	2,24
4	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	2,25

Tabela 4.31: Exemplo 3B resolvido pelo método ARSS.

4.9.7 Exemplo 3C

Para este exemplo escolheu-se 5 estratégias em cada extremo da discretização do tempo de execução de cada uma das cinco tarefas que representam o processo de negócios do Exemplo 3. Em um extremo, os tempos são menores, correspondendo a altos custos; e no outro extremo estão os maiores tempos de execução que correspondem aos menores custos.

índice estratégia original prov. 1	provedor 1	índice estratégia original prov. 2	provedor 2
1	(15,0; 14,0; 15,0; 16,0; 6,0)	1	(14,0; 13,0; 16,0; 10,0; 1,0)
2	(17,5; 14,0; 15,0; 16,0; 6,0)	2	(17,5; 13,0; 16,0; 10,0; 1,0)
3	(20,0; 14,0; 15,0; 16,0; 6,0)	3	(21,0; 13,0; 16,0; 10,0; 1,0)
4	(15,0; 16,5 15,0; 16,0,;6,0)	4	(14,0; 15,5; 16,0; 10,0; 1,0)
5	(17,5; 16,5; 15,0; 16,0; 6,0)	5	(17,5; 15,5; 16,0; 10,0; 1,0)
239	(17,5; 16,5; 16,0; 18,0; 8,0)	239	(17,5; 15,5; 18,0; 11,0; 5,0)
240	(20,0; 16,5; 16,0; 18,0; 8,0)	240	(21,0; 15,5; 18,0; 11,0; 5,0)
241	(15,0; 19,0; 16,0; 18,0; 8,0)	241	(14,0; 18,0; 18,0; 11,0; 5,0)
242	(17,5; 19,0; 16,0; 18,0; 8,0)	242	(17,5; 18,0; 18,0; 11,0; 5,0)
243	(20,0; 19,0; 16,0; 18,0; 8,0)	243	(21,0; 18,0; 18,0; 11,0; 5,0)

Tabela 4.32: Estratégias reduzidas do Exemplo 3C.

O método ARSS encontrou quatro equilíbrios de Nash, sendo que para o provedor 1 duas estratégias mistas e duas puras, e para o provedor 2 apenas estratégias puras, conforme mostrado na Tabela 4.34. As utilidades dos quatro equilíbrios foram R\$1,11 e R\$ 1,83 para os provedores 1 e 2, respectivamente.

O primeiro equilíbrio de Nash consiste no provedor 1 escolher 93.4% das vezes a estratégia $s_{61} = 239$, e 6.6% das vezes a estratégia $s_{71} = 240$; o provedor 2 deve escolher 100% das vezes a estratégia $s_{52} = 5$.

O segundo equilíbrio de Nash consiste no provedor 1 escolher 93.4% das vezes a estratégia $s_{61} = 239$ e 6.6% das vezes a estratégia $s_{71} = 240$; o provedor 2 deve escolher 100% das vezes a estratégia $s_{92} = 242$.

O terceiro equilíbrio de Nash consiste no provedor 1 escolher 100% das vezes a estratégia $s_{71} = 240$; o provedor 2 deve escolher 100% das vezes a estratégia $s_{52} = 5$.

	1	2	3	4	5	239	240	241	242	243
1	0,00, 4,54	0,00, 4,44	0,00, 4,34	0,00, 4,49	0,00, 4,38	0,00, 3,70	0,00, 3,60	0,00, 3,75	0,00, 3,65	0,00, 3,55
2	0,00, 4,54	0,00, 4,44	0,00, 4,34	0,00, 4,49	0,00, 4,38	0,00, 3,70	0,00, 3,60	0,00, 3,75	0,00, 3,65	0,00, 3,55
3	0,00, 3,04	0,00, 4,44	0,00, 4,34	1,50, 2,99	0,00, 4,38	0,00, 3,70	0,00, 3,60	1,50, 2,25	0,00, 3,65	0,00, 3,55
4	1,10, 2,73	1,10, 2,63	1,10, 2,54	1,10, 2,73	1,10, 2,63	1,10, 1,95	1,10, 1,85	1,10, 2,05	1,10, 1,95	1,10, 1,85
5	1,10, 2,73	1,10, 2,63	1,10, 2,54	1,10, 2,73	1,10, 2,63	1,10, 1,95	1,10, 1,85	1,10, 2,05	1,10, 1,95	1,10, 1,85
239	1,54, 1,75	1,54, 1,65	1,54, 1,55	1,54, 1,75	1,54, 1,65	1,11, 1,83	1,11, 1,73	1,11, 1,93	1,11, 1,83	1,11, 1,73
240	3,05, 0,25	1,54, 1,65	1,54, 1,55	3,04, 0,25	1,54, 1,65	1,11, 1,83	1,11, 1,73	2,61, 0,42	1,11, 1,83	1,11, 1,73
241	0,63, 1,75	0,63, 1,65	0,63, 1,55	0,63, 1,75	0,63, 1,65	0,21, 1,83	0,21, 1,73	0,21, 1,93	0,21, 1,83	0,21, 1,73
242	0,63, 1,75	0,63, 1,65	0,63, 1,55	0,63, 1,75	0,63, 1,65	0,21, 1,83	0,21, 1,73	0,21, 1,93	0,21, 1,83	0,21, 1,73
243	2,13, 0,25	0,63, 1,65	0,63, 1,55	2,13, 0,25	0,63, 1,65	0,21, 1,83	0,21, 1,73	1,71, 0,42	0,21, 1,83	0,21, 1,73

Tabela 4.33: Bimatriz de utilidades para estratégias reduzidas do Exemplo 3C.

O último equilíbrio de Nash consiste no provedor 1 escolher 100% das vezes a estratégia $s_{71} = 240$; o provedor 2 deve escolher 100% das vezes a estratégia $s_{92} = 240$.

EE	s_{11}	s_{21}	s_{31}	s_{41}	s_{51}	s_{61}	s_{71}	s_{81}	s_{91}	s_{101}	EP_1
1	0.0	0.0	0.0	0.0	0.0	0.934	0.066	0.0	0.0	0.0	1,11
2	0.0	0.0	0.0	0.0	0.0	0.934	0.066	0.0	0.0	0.0	1,11
3	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1,11
4	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1,11
EE	s_{12}	s_{22}	s_{32}	s_{42}	s_{52}	s_{62}	s_{72}	s_{82}	s_{92}	s_{102}	EP_2
1	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1,83
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1,83
3	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1,83
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1,83

Tabela 4.34: Exemplo 3C resolvido pelo ARSS.

4.9.8 Exemplo 4

Este exemplo é referente ao mesmo processo de negócios com seis tarefas da Figura 3.3.1 do Capítulo 3. Existem três caminhos de execução $T1 \rightarrow T2 \rightarrow T4 \rightarrow T11$, $T1 \rightarrow T2 \rightarrow T8 \rightarrow T11$ e $T1 \rightarrow T3$ com as seguintes frequências 0.35, 0.35 e 0.30, respectivamente.

Tempos	tarefa 1		tarefa 2		tarefa 3		tarefa 4		tarefa 5		tarefa 6	
	mín	máx	mín	máx	mín	máx	mín	máx	mín	máx	mín	máx
prov. 1	15	20	14	19	15	16	16	18	6	8	2	5
prov. 2	14	21	13	18	16	18	10	11	1	5	3	4
Preços												
	mín	máx	mín	máx	mín	máx	mín	máx	mín	máx	mín	máx
prov. 1	1,50	2,50	0,20	2,00	0,85	1,25	0,02	1,00	0,33	0,66	0,50	1,50
prov. 2	1,30	1,50	1,70	1,80	0,75	1,22	0,25	0,75	0,05	0,25	0,51	1,51

Tabela 4.35: Tempo (seg.) e preços em R\$ de execução do Exemplo 4.

Das 723 estratégias puras por provedor, foram selecionadas aleatoriamente 10 estratégias por provedor, apresentadas na Tabela 4.36.

índice estratégia original prov. 1	provedor 1	índice estratégia original prov. 2	provedor 2
455	(17,5; 16,5; 16,0; 17,0; 8,0; 3,5)	132	(21,0; 15,5; 18,0; 10,5; 3,0; 3,0)
124	(15,0; 19,0; 15,5; 17,0; 7,0; 2,0)	491	(17,5; 15,5; 16,0; 10,0; 1,0; 4,0)
673	(15,0; 19,0; 16,0; 16,0; 8,0; 5,0)	640	(14,0; 13,0; 18,0; 11,0; 3,0; 4,0)
593	(17,5; 19,0; 16,0; 16,0; 7,0; 5,0)	487	(14,0; 13,0; 16,0; 10,0; 1,0; 4,0)
339	(20,0; 16,5; 15,5; 16,0; 7,0; 3,5)	41	(17,5; 15,5; 17,0; 10,5; 1,0; 3,0)
584	(17,5; 19,0; 15,5; 16,0; 7,0; 5,0)	431	(17,5; 18,0; 18,0; 10,0; 5,0; 3,5)
34	(15,0; 19,0; 15,0; 17,0; 6,0; 2,0)	641	(17,5; 13,0; 18,0; 11,0; 3,0; 4,0)
137	(17,5; 14,0; 15,0; 18,0; 7,0; 2,0)	514	(14,0; 13,0; 16,0; 10,5; 1,0; 4,0)
39	(20,0; 14,0; 15,5; 17,0; 6,0; 2,0)	483	(21,0; 15,5; 18,0; 11,0; 5,0; 3,5)
470	(17,5; 14,0; 15,5; 18,0; 8,0; 3,5)	466	(14,0; 18,0; 16,0; 11,0; 5,0; 3,5)

Tabela 4.36: Estratégias reduzidas do Exemplo 4.

Após aplicar o *Solver* aos 100 perfis estratégicos resultantes, obtém-se as utilidades descritas na bimatriz na Tabela 4.37.

Da bimatriz da Tabela 4.37, extraiu-se as duas matrizes com as utilidades dos provedores 1 e 2 utilizadas pelo método ARSS. O resultado é apresentado na Tabela 4.38, onde se verifica que dois equilíbrios de Nash em estratégias mistas foram obtidos. O provedor 1 obteve a utilidade de R\$ 0,72 e o provedor 2 obteve a utilidade R\$ 2,13.

O primeiro equilíbrio de Nash, para o provedor 1, consiste escolher 93% das vezes entre as estratégias $s_{11} = 455$ e 7% das vezes $s_{51} = 339$; o provedor 2 deve escolher 100% das vezes a estratégia $s_{72} = 641$.

O segundo equilíbrio de Nash, o provedor 1 deve escolher 100% das vezes a estratégia $s_{51} = 339$; o provedor 2 deve escolher 100% das vezes a estratégia $s_{72} = 641$.

EE	s_{11}	s_{21}	s_{31}	s_{41}	s_{51}	s_{61}	s_{71}	s_{81}	s_{91}	s_{101}	EP_1
1	0.93	0.0	0.0	0.0	0.07	0.0	0.0	0.0	0.0	0.0	0,72
2	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0,72
EE	s_{12}	s_{22}	s_{32}	s_{42}	s_{52}	s_{62}	s_{72}	s_{82}	s_{92}	s_{101}	EP_2
1	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	2,13
2	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	2,13

Tabela 4.38: Exemplo 4 reduzido resolvido pelo método de ARSS.

	132	491	640	487	41	431	641	514	483	466
455	1,37, 1,78	1,19, 1,81	0,72, 2,23	1,19, 1,91	1,66, 1,65	1,54, 1,68	0,72, 2,13	1,01, 2,08	1,37, 1,66	1,66, 1,60
124	1,11, 1,78	0,68, 1,81	0,13, 2,23	0,68, 1,91	1,11, 1,99	0,31, 2,33	0,13, 2,13	0,50, 2,08	0,13, 2,32	0,50, 2,26
673	0,46, 1,78	0,75, 1,74	0,46, 1,89	0,75, 1,84	0,75, 1,65	0,46, 1,94	0,46, 1,79	0,75, 1,75	0,46, 1,66	0,75, 1,60
593	0,46, 1,78	0,75, 1,74	0,46, 1,89	0,75, 1,84	0,75, 1,65	0,46, 1,94	0,46, 1,79	0,75, 1,75	0,46, 1,66	0,75, 1,60
339	1,37, 1,78	1,08, 2,07	2,21, 0,73	2,58, 0,67	1,37, 1,99	1,37, 1,94	0,72, 2,13	2,58, 0,58	1,37, 1,66	3,23, 0,10
584	0,46, 1,78	0,82, 1,74	0,46, 1,89	0,82, 1,84	0,46, 1,99	0,46, 1,94	0,46, 1,79	0,82, 1,75	0,46, 1,66	0,82, 1,60
34	1,11, 1,78	0,31, 2,23	0,13, 2,23	0,31, 2,33	1,11, 1,99	0,31, 2,38	0,13, 2,13	0,13, 2,51	0,13, 2,32	0,13, 2,69
137	0,98, 2,75	0,01, 3,37	0,01, 3,31	0,01, 3,50	0,98, 2,96	0,01, 3,44	0,01, 3,21	0,01, 3,50	0,01, 3,37	0,01, 3,70
39	0,97, 2,92	0,55, 2,94	1,50, 1,90	2,05, 1,58	0,97, 3,13	0,18, 3,44	0,00, 3,30	1,87, 1,75	0,00, 3,46	1,87, 1,86
470	0,66, 2,75	0,37, 2,94	0,01, 3,31	0,37, 3,08	0,66, 2,96	0,66, 2,78	0,01, 3,21	0,37, 3,28	0,66, 2,71	1,02, 2,62

Tabela 4.37: Bimatriz de utilidades para estratégias reduzidas do Exemplo 4.

Equilíbrio Correlacionado

5.1 Equilíbrio Correlacionado

A principal diferença entre um jogo estratégico analisado por meio do Equilíbrio de Nash e o mesmo jogo analisado pelo Equilíbrio Correlacionado é a existência de um **agente de sinalização** que usa algum dispositivo randômico para indicar aos jogadores qual perfil estratégico seria escolhido. Neste capítulo adota-se o pressuposto de que os provedores concordam em coordenar as suas estratégias para benefício mútuo (e prejuízo do contratante de serviços web). Questões que possam envolver disputas legais ou judiciais não são consideradas neste capítulo, porém em uma situação mais realista contratos legais deveriam limitar algumas ações possíveis neste modelo correlacionado.

Considere um exemplo com dois provedores de serviços. Os tempos de execução de uma certa tarefa podem ser de 3 segundos ou 4 segundos para o primeiro provedor. Para o segundo provedor, podem ser 2 segundos ou 3 segundos. O agente de sinalização usa um dispositivo randômico para selecionar uma combinação qualquer dos quatro perfis estratégicos da Tabela 5.1.

Suponha, como exemplo, que a atribuição de probabilidades é como na Tabela 5.2. De acordo com a Tabela 5.2, o perfil estratégico (4, 2) nunca será escolhido pelo dispositivo randômico; o perfil estratégico (3, 2) será escolhido 25% das vezes; o perfil estratégico (3, 3) será escolhido 15% das vezes; e, finalmente, o perfil estratégico (4, 3) será escolhido 60% das vezes. Nestas condições, a utilidade esperada em R\$ do primeiro jogador será $2 \times .25 + 0 \times .15 + 3 \times 0.00 + 1 \times 0.60 = 1.10$, e a utilidade esperada em R\$ do segundo jogador será $2 \times .25 + 3 \times .15 + 0 \times 0.00 + 1 \times 0.60 = 1.55$.

Um Equilíbrio Correlacionado representa uma escolha de probabilidades tal que as utilidades esperadas de ambos os jogadores são maiores ou iguais do que as que seriam obtidas por qualquer equilíbrio de Nash.

O equilíbrio correlacionado pode ser imaginado como um dispositivo de randomização que sinaliza ao primeiro jogador usar a linha i , e ao segundo jogador usar a coluna j com probabilidade

	2 segundos	3 segundos
3 segundos	R\$2, R\$2	R\$0, R\$3
4 segundos	R\$3, R\$0	R\$1 , R\$1

Tabela 5.1: Jogo de provedores para serviços web.

	2 segundos	3 segundos
3 segundos	25%	15%
4 segundos	0%	60%

Tabela 5.2: Probabilidades.

$prob_{ij}$. Qualquer número entre 0 e 1 poderia ser sinalizado.

Um perfil estratégico é um equilíbrio correlacionado se sempre que ao primeiro jogador é dado o sinal para usar i , então o jogador não ganha desobedecendo, dada a probabilidade condicional que este sinal induz nas ações do segundo jogador. Similarmente, o segundo jogador não poderia ganhar desobedecendo o sinal para usar j .

Se ao primeiro jogador 1 é sinalizado o uso da linha i , e ele realmente usa i , então (assumindo que o jogador 2 está obedecendo o seu sinal) a utilidade esperada do primeiro jogador seria

$$\sum_j U_1(i, j) prob_{ij} / \sum_j prob_{ij}. \quad (5.1.1)$$

Observe que $prob_{ij} / \sum_i prob_{ij}$ representa a probabilidade do jogador 1 condicionada ao fato do jogador 1 haver adotado a estratégia i .

Se o jogador 1 usar outra linha k como estratégia, a utilidade esperada seria

$$\sum_j U_1(k, j) prob_{ij} / \sum_j prob_{ij}. \quad (5.1.2)$$

Dado que o primeiro jogador não pode ganhar desobedecendo, deve valer para qualquer par de linhas i e k ,

$$\sum_j (U_1(i, j) - U_1(k, j)) prob_{ij} / \sum_j prob_{ij} \geq 0. \quad (5.1.3)$$

De forma idêntica, dado que o jogador 2 não pode ganhar ao não utilizar a coluna j , deve valer para qualquer par de colunas j e l ,

$$\sum_i (U_2(i, j) - U_2(i, l)) prob_{ij} / \sum_i prob_{ij} \geq 0. \quad (5.1.4)$$

Um equilíbrio correlacionado deve satisfazer as desigualdades (5.1.3)-(5.1.4).

O Equilíbrio Correlacionado apresenta uma série de propriedades importantes ((Papadimitriou & Roughgarden 2008)).

Proposição 2. *Um Equilíbrio de Nash é necessariamente um Equilíbrio Correlacionado.*

Para definir um Equilíbrio Correlacionado a partir de um equilíbrio de Nash, basta definir a probabilidade de cada perfil como um produto das probabilidades das estratégias mistas correspondentes dos jogadores 1 e 2. Uma consequência da Proposição 2 é a de que sempre existe um Equilíbrio Correlacionado.

Proposição 3. *O conjunto dos Equilíbrios Correlacionados é um conjunto convexo.*

Em particular, a Proposição 3 garante que qualquer par de utilidades geradas pela combinação convexa de pares produzidos por Equilíbrios de Nash pode ser caracterizado como um Equilíbrio Correlacionado.

5.2 Passos para Obtenção dos Equilíbrios Correlacionados

1. Definir as estratégias puras do jogo, usando os tempos de execução das tarefas.
2. Computar a matriz de jogo contendo os perfis estratégicos puros e as utilidades dos jogadores de forma análoga à realizada no Capítulo 3.
3. Calcular os equilíbrios correlacionados para obter as probabilidades de cada perfil estratégico.
4. Gerar a sinalização de acordo com a distribuição de probabilidades obtida e comunicá-la aos jogadores.

5.3 Definição dos Sinais

Quais seriam os sinais externos gerados pelo agente de sinalização? No caso de dois jogadores, são gerados dois sinais, d_1 e d_2 , correspondentes ao índice da linha do jogador 1 e ao índice da coluna do jogador 2. A primeira estratégia do jogador deve ser escolhida quando for gerado o número 1; a segunda estratégia deve ser escolhida quando for gerado o número 2, e assim por diante. O máximo número gerado é m , indicando a escolha da m -ésima estratégia.

Por exemplo, o elemento $prob_{12}$ significa que o perfil estratégico $(1, 2)$ é escolhido com probabilidade igual a $prob_{12}$. O perfil estratégico $(1, 2)$ corresponde à estratégia 1 do jogador 1 e à estratégia 2 do jogador 2.

	1	2	...	m
1	$prob_{11}$	$prob_{12}$	...	$prob_{1m}$
2	$prob_{21}$	$prob_{22}$	...	$prob_{2m}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
m	$prob_{m1}$	$prob_{m2}$	...	$prob_{mm}$

Table 5.3: Sinais como Probabilidades.

Considere a variável aleatória $d = (d_1, d_2)$, $d_1, d_2 \in \{1, \dots, m\}$. A probabilidade conjunta de d é $prob(d_1, d_2) = prob_{d_1 d_2}$, a ser calculada.

5.4 Determinação do Equilíbrio Correlacionado

Equilíbrios Correlacionados são encontrados ao se especificar uma distribuição de probabilidade $prob_{d_1 d_2}$, tal que as utilidades de todos os jogadores sejam máximas. São obtidos em termos da utilidade média ou esperada de cada jogador, pois cada perfil estratégico é escolhido com uma probabilidade $prob_{ij}$.

Os equilíbrios correlacionados podem ser obtidos resolvendo-se um problema multiobjetivo envolvendo as utilidades esperadas dos jogadores 1 e 2,

$$f_1 = \sum_{i=1}^m \sum_{j=1}^m U_1(i, j) prob_{ij}, \quad (5.4.1)$$

$$f_2 = \sum_{i=1}^m \sum_{j=1}^m U_2(i, j) prob_{ij}, \quad (5.4.2)$$

e as restrições de equilíbrio correlacionado:

$$\sum_{j=1}^m (U_1(i, j) - U_1(k, j)) prob_{ij} \geq 0, \forall i, k = 1, \dots, m, \quad (5.4.3)$$

$$\sum_{i=1}^m (U_2(i, j) - U_2(i, k)) prob_{ij} \geq 0, \forall j, k = 1, \dots, m \quad (5.4.4)$$

$$\sum_{i=1}^m \sum_{j=1}^m prob_{ij} = 1 \quad (5.4.5)$$

$$prob_{ij} \geq 0, \forall i, j = 1, \dots, m \quad (5.4.6)$$

Equilíbrios correlacionados são geralmente obtidos maximizando-se a soma $f_1 + f_2$ sujeito às restrições de equilíbrio correlacionado.

Uma característica marcante deste problema é a de ser linear nas variáveis $prob_{ij}$. Com isso, a determinação de Equilíbrios Correlacionados é em geral muito mais simples do que a determinação do Equilíbrio de Nash.

Além disso, assegura-se que o Equilíbrio Correlacionado obtido é Pareto-ótimo em relação a todos os Equilíbrios Correlacionados do problema, isto é, não existe qualquer outro Equilíbrio Correlacionado cujo par de retornos esperados domine o produzido pela solução do problema linear ((Papadimitriou & Roughgarden 2008)).

O agente de sinalização gera os sinais de acordo com a distribuição de probabilidades obtida, e os comunica aos provedores.

5.5 Resultados Numéricos de Otimização

5.5.1 Exemplo 1

Este exemplo corresponde ao Exemplo 1 do Capítulo 4, com a bimatriz reproduzida na Tabela 5.4 por conveniência.

Um equilíbrio correlacionado foi obtido pela resolução do problema formulado na seção anterior por meio de um *Solver*. Probabilidade diferente de zero foi obtida apenas para o perfil estratégico (1, 8) correspondendo à probabilidade 100%. As utilidades em R\$ dos provedores 1 e 2, nestas condições, foram 2,00 e 1,00, respectivamente. Estes valores coincidem com os retornos produzidos pelos equilíbrios de Nash puros do jogo descrito no Exemplo 1 do Capítulo 4.

	1	2	3	4	5	6	7	8	9
1	4,50, 0,00	4,50, 0,00	4,50, 0,00	4,50, 0,00	4,50, 0,00	4,50, 0,00	2,00, 1,00	2,00, 1,00	2,00, 1,00
2	4,25, 0,00	4,25, 0,00	4,25, 0,00	4,25, 0,00	4,25, 0,00	4,25, 0,00	1,75, 1,00	1,75, 1,00	1,75, 1,00
3	4,00, 0,00	4,00, 0,00	4,00, 0,00	4,00, 0,00	4,00, 0,00	4,00, 0,00	1,50, 1,00	1,50, 1,00	1,50, 1,00
4	4,25, 0,00	4,25, 0,00	4,25, 0,00	4,25, 0,00	4,25, 0,00	4,25, 0,00	2,00, 1,00	2,00, 1,00	2,00, 1,00
5	4,00, 0,00	4,00, 0,00	4,00, 0,00	4,00, 0,00	4,00, 0,00	4,00, 0,00	1,75, 1,00	1,75, 1,00	1,75, 1,00
6	3,75, 0,00	3,75, 0,00	3,75, 0,00	3,75, 0,00	3,75, 0,00	3,75, 0,00	1,50, 1,00	1,50, 1,00	1,50, 1,00
7	4,00, 0,00	4,00, 0,00	4,00, 0,00	4,00, 0,00	4,00, 0,00	4,00, 0,00	2,00, 1,00	2,00, 1,00	2,00, 1,00
8	3,75, 0,00	3,75, 0,00	3,75, 0,00	3,75, 0,00	3,75, 0,00	3,75, 0,00	1,75, 1,00	1,75, 1,00	1,75, 1,00
9	3,50, 0,00	3,50, 0,00	3,50, 0,00	3,50, 0,00	3,50, 0,00	3,50, 0,00	1,50, 1,00	1,50, 1,00	1,50, 1,00

Tabela 5.4: Bimatriz com utilidades do Exemplo 1.

5.5.2 Exemplo 2

Este exemplo corresponde ao Exemplo 2 do Capítulo 4. A bimatriz associada é reapresentada na Tabela 5.5.

O equilíbrio correlacionado obtido atribui aos perfis estratégicos (2, 7) e (8, 7) probabilidades 85% e 15%, respectivamente. Estes perfis estratégicos correspondem às utilidades R\$ 1,75 e R\$ 1,70 para os provedores 1 e 2, respectivamente.

5.5.3 Exemplo 3

A mesma bimatriz de utilidades do Exemplo 4 (Tabela 4.26) do Capítulo 3 foi utilizada neste exemplo. Através do *Solver*, pode-se obter a máxima utilidade do provedor 1 igual a R\$ 1,11 e a máxima utilidade do provedor 2 igual à R\$ 1,96. O equilíbrio correlacionado obtido compõe-se dos perfis estratégicos (116, 210) com probabilidade de 15,3%, (157, 210) com probabilidade de 0,2% e (116, 210) com probabilidade de 84,3%.

5.5.4 Exemplo 3A

A bimatriz de utilidades está apresentada na Tabela 4.28 do Capítulo 4. O equilíbrio correlacionado consistiu dos perfis estratégicos: (210, 116) com 15.2 % de probabilidade de execução, (19, 116) com 83.5% de probabilidade de execução e (119, 116) com 1.1% de probabilidade de execução. As utilidades dos provedores 1 e 2 foram R\$ 1,12 e R\$ 1,76, respectivamente.

5.5.5 Exemplo 3B

A bimatriz de utilidades está apresentada na Tabela 4.30 do Capítulo 4. Apenas o perfil estratégico (186, 131) com probabilidade de execução de 100% foi selecionado pelo *Solver*. As utilidades dos provedores 1 e 2 foram R\$ 1,10 e R\$ 2,25, respectivamente.

Neste exemplo, obteve-se claramente uma das principais propriedades do Equilíbrio Correlacionado: a de produzir equilíbrios Pareto-ótimos. Dentre os quatro equilíbrios de Nash mistos do Exemplo 3B do Capítulo 3, o equilíbrio de Nash com os máximos retornos para ambos os provedores foi o que correspondeu ao Equilíbrio Correlacionado.

5.5.6 Exemplo 3C

A bimatriz de utilidades está apresentada na Tabela 4.33 do Capítulo 4. O *Solver* obteve as utilidades de R\$ 1,11 e R\$ 1,83 para os provedores 1 e 2, respectivamente. O equilíbrio correlacionado obtido é composto dos perfis estratégicos (239, 239) com probabilidade de 93.4% de execução e o (239, 240) com probabilidade de 6.6% de execução.

5.5.7 Exemplo 4

A bimatriz de utilidades do Exemplo 4 (Tabela 4.37) do Capítulo 4 foi utilizada neste exemplo. Com o *Solver*, foi obtido a máxima utilidade do provedor 1 igual a R\$ 0,72 e a máxima utilidade do provedor 2 igual a R\$ 2,13. O equilíbrio correlacionado obtido compõe-se do perfil estratégico (34, 132) com 93.3% de probabilidade de execução e do perfil estratégico (34, 41) com 6.7% de probabilidade de execução.

	1	2	3	4	5	6	7	8	9
1	2,25, 1,90	2,25, 1,83	2,25, 1,75	0,00, 3,90	0,00, 3,83	0,00, 3,75	0,00, 3,60	0,00, 3,53	0,00, 3,45
2	4,00, 0,00	4,00, 0,00	4,00, 0,00	1,75, 2,00	1,75, 2,00	1,75, 2,00	1,75, 1,70	1,75, 1,70	1,75, 1,70
3	3,75, 0,00	3,75, 0,00	3,75, 0,00	1,50, 2,00	1,50, 2,00	1,50, 2,00	1,50, 1,70	1,50, 1,70	1,50, 1,70
4	2,02, 1,90	2,02, 1,83	2,02, 1,75	0,00, 3,90	0,00, 3,83	0,00, 3,75	0,00, 3,60	0,00, 3,53	0,00, 3,45
5	3,77, 0,00	3,77, 0,00	3,77, 0,00	1,75, 2,00	1,75, 2,00	1,75, 2,00	1,75, 1,70	1,75, 1,70	1,75, 1,70
6	3,52, 0,00	3,52, 0,00	3,52, 0,00	1,50, 2,00	1,50, 2,00	1,50, 2,00	1,50, 1,70	1,50, 1,70	1,50, 1,70
7	1,80, 1,90	1,80, 1,83	1,80, 1,75	1,80, 1,90	1,80, 1,83	1,80, 1,75	0,00, 3,60	0,00, 3,53	0,00, 3,45
8	3,55, 0,00	3,55, 0,00	3,55, 0,00	3,55, 0,00	3,55, 0,00	3,55, 0,00	1,75, 1,70	1,75, 1,70	1,75, 1,70
9	3,30, 0,00	3,30, 0,00	3,30, 0,00	3,30, 0,00	3,30, 0,00	3,30, 0,00	1,50, 1,70	1,50, 1,70	1,50, 1,70

Tabela 5.5: Bimatriz com utilidades do Exemplo 2.

Equilíbrio Bayes-Nash

Neste capítulo, apresenta-se uma solução para o problema de programação de serviços web modelado como um jogo de Informação Incompleta, isto é, como um jogo em que parte das informações (estratégias, retornos, número de jogadores, etc) não é de conhecimento comum a todos os jogadores. Em outras palavras, um ou mais jogadores detem informação privada (privilegiada) sobre o jogo.

Como exemplo introdutório, considere quatro jogos estratégicos independentes, de nomes A, B, C e D, com utilidades distintas umas das outras. Existem apenas dois jogadores em qualquer jogo. As estratégias do jogador 1 serão sempre *Up* ou *Down*, e para o jogador 2 serão sempre *Left* ou *Right*. Pode-se aplicar o conceito de equilíbrio de Nash a qualquer bimatriz das Tabelas 6.1, 6.2, 6.3 e 6.4, desde que os jogadores saibam em qual dos quatro jogos eles se encontram.

A	Left	Right
Up	2,0	0,2
Down	0,2	2,0

Tabela 6.1: Bimatriz do jogo estratégico A.

B	Left	Right
Up	2,2	0,3
Down	0,3	1,1

Tabela 6.2: Bimatriz do jogo estratégico B.

C	Left	Right
Up	2,2	0,0
Down	0,0	1,1

Tabela 6.3: Bimatriz do jogo estratégico C.

D	Left	Right
Up	2,1	0,0
Down	0,0	1,2

Tabela 6.4: Bimatriz do jogo estratégico D.

Dentre outras possibilidades, um jogo bayesiano modela a situação na qual não existe certeza de qual jogo se está participando, porém existe uma probabilidade de se estar em qualquer destes jogos.

A Figura 6.0.1 representa um Jogo Bayesiano com dois jogadores e duas estratégias possíveis para o jogador 1, *Up* e *Down*, pertencentes ao conjunto $\{U, D\}$. As estratégias do jogador 2 são *Left* e *Right*, pertencentes ao conjunto $\{L, R\}$. Existem quatro jogos A, B, C e D, e as probabilidades de se estar em cada jogo seriam 0.3 no jogo A, 0.1 no jogo B, 0.2 no jogo C e 0.4 no jogo D. A partir da Figura 6.0.1 constroi-se a Tabela 6.5 que facilitará as definições posteriores.

No jogo Bayesiano, usa-se o conceito de **tipo** para indicar qual jogo estamos nos referindo. Cada jogador i tem o seu próprio conjunto de tipos indicados por θ_i .

Os valores $\theta_{1,1}$ e $\theta_{1,2}$ são as instâncias do tipo θ_1 , da mesma forma que $\theta_{2,1}$ e $\theta_{2,2}$ são as instâncias do tipo θ_2 . O jogador 1 tem dois tipos possíveis pertencentes ao conjunto $\{\theta_{1,1}, \theta_{1,2}\}$; o jogador 2 tem dois tipos pertencentes ao conjunto $\{\theta_{2,1}, \theta_{2,2}\}$. Se considerarmos o tipo $\theta_{1,1}$, teremos as utilidades dos jogos A ou B; se considerarmos o tipo $\theta_{1,2}$, teremos as utilidades dos jogos C ou D; se considerarmos o tipo $\theta_{2,1}$, teremos as utilidades dos jogos A ou C; e finalmente, se considerarmos o tipo $\theta_{2,2}$, teremos as utilidades dos jogos B ou D.

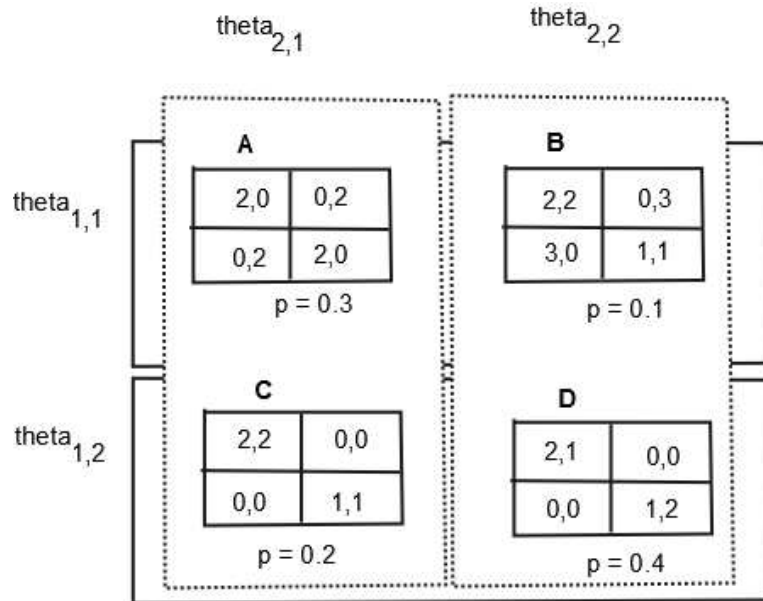


Figura 6.0.1: Exemplo de Jogo Bayesiano.

Um **Jogo Bayesiano** é uma tupla (N, A, Θ, p, u) , onde $N = \{1, \dots, n\}$ é o conjunto de jogadores; $A = A_1 \times \dots \times A_n$, onde cada A_i denota o conjunto das ações disponíveis ao jogador i ; $\Theta = \Theta_1 \times \dots \times \Theta_n$, onde cada Θ_i é o conjunto de tipos ao jogador i ; $p : \Theta_1 \times \dots \times \Theta_n \rightarrow [0, 1]$ é uma distribuição de probabilidades, conhecida à priori; e $u = (u_1, \dots, u_m)$, onde $u_i : A \times \Theta \rightarrow \mathbb{R}$

é a função de utilidade do jogador i , que incorpora estratégias e tipos (diferentemente de um jogo comum que inclui apenas estratégias).

As probabilidades conjuntas, segundo a Figura 6.0.1, são:

- $p(\theta_{1,1}, \theta_{2,1}) = 0.3$ é a probabilidade dos tipos serem $\theta_{1,1}$ e $\theta_{2,1}$;
- $p(\theta_{1,1}, \theta_{2,2}) = 0.1$ é a probabilidade dos tipos serem $\theta_{1,1}$ e $\theta_{2,2}$;
- $p(\theta_{1,2}, \theta_{2,1}) = 0.2$ é a probabilidade dos tipos serem $\theta_{1,2}$ e $\theta_{2,1}$;
- $p(\theta_{1,2}, \theta_{2,2}) = 0.4$ é a probabilidade dos tipos serem $\theta_{1,2}$ e $\theta_{2,2}$.

As probabilidades individuais são definidas como probabilidades marginais. Para o jogador 1,

$$\begin{aligned} p(\theta_{1,1}) &= p(\theta_{1,1}, \theta_{2,1}) + p(\theta_{1,1}, \theta_{2,2}) = 0.3 + 0.1 = 0.4, \\ p(\theta_{1,2}) &= p(\theta_{1,2}, \theta_{2,1}) + p(\theta_{1,2}, \theta_{2,2}) = 0.2 + 0.4 = 0.6, \end{aligned}$$

de forma que $p(\theta_{1,1}) + p(\theta_{1,2}) = 0.4 + 0.6 = 1$.

Para o jogador 2,

$$\begin{aligned} p(\theta_{2,1}) &= p(\theta_{1,1}, \theta_{2,1}) + p(\theta_{1,2}, \theta_{2,1}) = 0.3 + 0.2 = 0.5, \\ p(\theta_{2,2}) &= p(\theta_{1,1}, \theta_{2,2}) + p(\theta_{1,2}, \theta_{2,2}) = 0.1 + 0.4 = 0.5, \end{aligned}$$

de forma que $p(\theta_{2,1}) + p(\theta_{2,2}) = 0.5 + 0.5 = 1$.

As probabilidades condicionais do jogador 2 são obtidas das probabilidades conjuntas:

- $p(\theta_{2,1} \mid \theta_{1,1}) = \frac{p(\theta_{2,1}, \theta_{1,1})}{p(\theta_{2,1}, \theta_{1,1}) + p(\theta_{2,2}, \theta_{1,1})} = \frac{0.3}{0.3+0.1} = \frac{3}{4}$ é a probabilidade condicional do jogador 2 no jogo “A” quando o tipo do jogador 1 é $\theta_{1,1}$;
- $p(\theta_{2,2} \mid \theta_{1,1}) = \frac{p(\theta_{2,2}, \theta_{1,1})}{p(\theta_{2,1}, \theta_{1,1}) + p(\theta_{2,2}, \theta_{1,1})} = \frac{0.1}{0.3+0.1} = \frac{1}{4}$ é a probabilidade condicional do jogador 2 no jogo “B” quando o tipo do jogador 1 é $\theta_{1,1}$;
- $p(\theta_{2,1} \mid \theta_{1,2}) = \frac{p(\theta_{2,1}, \theta_{1,2})}{p(\theta_{2,1}, \theta_{1,2}) + p(\theta_{2,2}, \theta_{1,2})} = \frac{0.2}{0.2+0.4} = \frac{1}{3}$ é a probabilidade condicional do jogador 2 no jogo “C” quando o tipo do jogador 1 é $\theta_{1,2}$;
- $p(\theta_{2,2} \mid \theta_{1,2}) = \frac{p(\theta_{2,2}, \theta_{1,2})}{p(\theta_{2,1}, \theta_{1,2}) + p(\theta_{2,2}, \theta_{1,2})} = \frac{0.4}{0.2+0.4} = \frac{2}{3}$ é a probabilidade condicional do jogador 2 no jogo “D” quando o tipo do jogador 1 é $\theta_{1,2}$.

O mesmo procedimento se aplica ao cálculo das probabilidades condicionais do jogador 1. Estas informações de probabilidade serão usadas no cálculo das utilidades *ex-ante* e *ex-interim*.

a_1	a_2	θ_1	θ_2	$u_1(a_1, a_2, \theta_1, \theta_2)$	$u_2(a_1, a_2, \theta_1, \theta_2)$
U	L	$\theta_{1,1}$	$\theta_{2,1}$	2	0
U	L	$\theta_{1,1}$	$\theta_{2,2}$	2	2
U	L	$\theta_{1,2}$	$\theta_{2,1}$	2	2
U	L	$\theta_{1,2}$	$\theta_{2,2}$	2	1
U	R	$\theta_{1,1}$	$\theta_{2,1}$	0	2
U	R	$\theta_{1,1}$	$\theta_{2,2}$	0	3
U	R	$\theta_{1,2}$	$\theta_{2,1}$	0	0
U	R	$\theta_{1,2}$	$\theta_{2,2}$	0	0
D	L	$\theta_{1,1}$	$\theta_{2,1}$	0	2
D	L	$\theta_{1,1}$	$\theta_{2,2}$	3	0
D	L	$\theta_{1,2}$	$\theta_{2,1}$	0	0
D	L	$\theta_{1,2}$	$\theta_{2,2}$	0	0
D	R	$\theta_{1,1}$	$\theta_{2,1}$	2	0
D	R	$\theta_{1,1}$	$\theta_{2,2}$	1	1
D	R	$\theta_{1,2}$	$\theta_{2,1}$	1	1
D	R	$\theta_{1,2}$	$\theta_{2,2}$	1	2

Tabela 6.5: Utilidades u_1 e u_2 do jogo Bayesiano.

6.1 Utilidade Esperada *Ex-Post*

A utilidade *ex-post* é a utilidade esperada quando todos os jogadores sabem os tipos uns dos outros.

Considere uma situação onde existe um perfil estratégico misto $\sigma = (\sigma_1, \dots, \sigma_n)$ e tipos conhecidos $\theta = (\theta_1, \dots, \theta_n)$. As ações a_i de cada jogador i pertencem ao conjunto A_i , e o vetor de ações $a = (a_1, \dots, a_n)$ pertencem ao conjunto $A = A_1 \times \dots \times A_n$. A definição de **Utilidade Esperada *Ex-Post*** é:

$$EU - ex - post_i(\sigma, \theta) = \sum_{(a_1, \dots, a_n) \in A_1 \times \dots \times A_n} \left(\prod_{j \in I} \sigma_j(a_j | \theta_j) \right) u_i(a, \theta).$$

A fórmula acima pode ser aplicada em situações onde existam tanto estratégias mistas quanto estratégias puras. O exemplo a seguir ilustra estes casos, mas apenas estratégias puras serão consideradas nos conceitos de equilíbrio usados.

Aplicando-se esta definição ao exemplo na Tabela 6.5, temos:

$$EU - ex - post_2((\sigma_1, \sigma_2), (\theta_1, \theta_2)) = \sum_{a \in A} \left(\prod_{j \in I} \sigma_j(a_j | \theta_j) \right) u_2(a, \theta).$$

A produtória $\prod_{j \in I} \sigma_j(a_j | \theta_j)$ é expandida para $\sigma_1(a_1 | \theta_1) \sigma_2(a_2 | \theta_2)$. A utilidade *ex-post* seria expressa como:

$$EU - ex - post_2(\sigma, \theta) = \sum_{(a_1, a_2) \in A_1 \times A_2} (\sigma_1(a_1 | \theta_1) \sigma_2(a_2 | \theta_2)) u_2(a_1, a_2, \theta_1, \theta_2).$$

Como $a_1 \in \{U, D\}$ e $a_2 \in \{L, R\}$, a expansão parcial (pois falta expandir os θ_i) neste exemplo seria:

$$\begin{aligned} EU - ex - post_2(\sigma, \theta) = \\ \sigma_1(U|\theta_1) \cdot \sigma_2(L|\theta_2) \cdot u_2(U, L, \theta_1, \theta_2) + \sigma_1(U|\theta_1) \cdot \sigma_2(R|\theta_2) \cdot u_2(U, R, \theta_1, \theta_2) + \\ \sigma_1(D|\theta_1) \cdot \sigma_2(L|\theta_2) \cdot u_2(D, L, \theta_1, \theta_2) + \sigma_1(D|\theta_1) \cdot \sigma_2(R|\theta_2) \cdot u_2(D, R, \theta_1, \theta_2). \end{aligned}$$

Note que $\sigma_1(U|\theta_1)$ e $\sigma_1(D|\theta_1)$ denotam *estratégias mistas* ao jogador 1, ou seja, probabilidades que o jogador 1 escolhe para randomizar as ações do conjunto $\{U, D\}$. De forma similar, $\sigma_2(L|\theta_2)$ e $\sigma_2(R|\theta_2)$ são *estratégias mistas* do jogador 2, ou seja, probabilidades que o jogador 2 escolhe para randomizar as ações do conjunto $\{L, R\}$.

O tipo θ_1 é igual a $\theta_{1,1}$ ou $\theta_{1,2}$; o tipo θ_2 é igual a $\theta_{2,1}$ ou $\theta_{2,2}$. Portanto, a expressão de $EU - ex - post_2$, no caso específico de θ_1 ser igual a instância $\theta_{1,2}$ e θ_2 ser igual a instância $\theta_{2,1}$ corresponde a seguinte expansão total:

$$EU - ex - post_2(\sigma_1(U|\theta_{1,2}), \sigma_1(D|\theta_{1,2}), \sigma_2(L|\theta_{2,1}), \sigma_2(R|\theta_{2,1}), \theta_{1,2}, \theta_{2,1}) =$$

$$\sigma_1(U|\theta_{1,2})\sigma_2(L|\theta_{2,1})u_2(U, L, \theta_{1,2}, \theta_{2,1}) + \quad (6.1.1)$$

$$\sigma_1(U|\theta_{1,2})\sigma_2(R|\theta_{2,1})u_2(U, R, \theta_{1,2}, \theta_{2,1}) + \quad (6.1.2)$$

$$\sigma_1(D|\theta_{1,2})\sigma_2(L|\theta_{2,1})u_2(D, L, \theta_{1,2}, \theta_{2,1}) + \quad (6.1.3)$$

$$\sigma_1(D|\theta_{1,2})\sigma_2(R|\theta_{2,1})u_2(D, R, \theta_{1,2}, \theta_{2,1}). \quad (6.1.4)$$

Por exemplo, suponha que o jogador 1 escolha U se θ_1 for igual à $\theta_{1,1}$, e escolha D se θ_1 for igual à $\theta_{1,2}$. Resumidamente.

$$U, \quad \theta_1 = \theta_{1,1}, \quad (6.1.5)$$

$$D, \quad \theta_1 = \theta_{1,2}. \quad (6.1.6)$$

Do mesmo modo, suponha que o jogador 2 escolha L se θ_2 for igual à $\theta_{2,1}$, e escolha R se θ_2 for igual à $\theta_{2,2}$:

$$L, \quad \theta_2 = \theta_{2,1}, \quad (6.1.7)$$

$$R, \quad \theta_2 = \theta_{2,2}. \quad (6.1.8)$$

Este exemplo não envolve qualquer estratégia mista porque as randomizações entre as ações serão 0 ou 1:

- De (6.1.5), deduzimos que $\sigma_1(U | \theta_{1,1})$ é igual a 1 e $\sigma_1(D | \theta_{1,1})$ é igual a 0;
- De (6.1.6), deduzimos que $\sigma_1(U | \theta_{1,2})$ é igual a 0 e $\sigma_1(D | \theta_{1,2})$ é igual a 1;
- De (6.1.7), deduzimos que $\sigma_2(L | \theta_{2,1})$ é igual a 1 e $\sigma_2(R | \theta_{2,1})$ é igual a 0;
- De (6.1.8), deduzimos que $\sigma_2(L | \theta_{2,2})$ é igual a 0 e $\sigma_2(R | \theta_{2,2})$ é igual a 1.

Considerando a soma das quatro parcelas (6.1.1)-(6.1.4), apenas a parcela (6.1.3) não é nula, o que fornece:

$$\begin{aligned} EU - ex - post_2(\sigma_1(U|\theta_{1,2}), \sigma_1(D|\theta_{1,2}), \sigma_2(L|\theta_{2,1}), \sigma_2(R|\theta_{2,1}), \theta_{1,2}, \theta_{2,1}) = \\ u_2(D, L, \theta_{1,2}, \theta_{2,1}). \end{aligned}$$

6.2 Utilidade Esperada *Ex-Interim*

A **Utilidade Esperada *Ex-Interim*** é a utilidade esperada quando cada jogador i conhece apenas o seu próprio tipo θ_i . A definição é:

$$EU - ex - interim_i(\sigma, \theta_i) = \sum_{\theta_{-i} \in \Theta_{-i}} p(\theta_{-i} \mid \theta_i) EU - ex - post_i(\sigma, \theta_i, \theta_{-i}).$$

A utilidade *ex-interim* é a soma ponderada pelas probabilidades condicionais $p(\theta_{-i} \mid \theta_i)$ das utilidades *ex-post* de cada estratégia mista e tipo θ_i conhecido; θ_{-i} denota o vetor de tipos de todos os demais jogadores.

6.3 Utilidade Esperada *Ex-Ante*

A **Utilidade Esperada *Ex-Ante*** é a utilidade esperada quando nenhuma informação privada é conhecida. A definição é:

$$EU - ex - ante_i(\sigma) = \sum_{\theta \in \Theta} p(\theta) EU - ex - post_i(\sigma, \theta),$$

onde $p(\theta)$ é a probabilidade conjunta $p(\theta_1, \dots, \theta_n)$.

A utilidade *ex-ante* é a soma ponderada pela probabilidade conjunta $p(\theta)$ das utilidades *ex-post* das estratégias mistas σ e tipos θ dos jogadores. Reecrevendo a definição acima, temos:

$$EU - ex - ante_i(\sigma) = \sum_{\theta_i \in \Theta_i} p(\theta_i) EU - ex - interim_i(\sigma, \theta_i),$$

onde $p(\theta_i)$ é a probabilidade marginal $p(\theta_i) = \sum_{\theta_{-i}} p(\theta_i, \theta_{-i})$.

Assim, de forma alternativa, a utilidade *ex-ante* é a soma ponderada pela probabilidade marginal $p(\theta_i)$ das utilidades *ex-interim* das estratégias mistas e tipo θ_i conhecido.

6.4 Melhor Resposta Estratégica

Qual estratégia mista σ_i o jogador i deveria escolher de forma a maximizar a sua utilidade esperada *ex-ante*, dado o perfil estratégico σ_{-i} dos demais jogadores? Todas as estratégias mistas σ_i que satisfazem esta condição fazem parte do conjunto de Melhores Respostas do jogador i .

O conjunto de **Melhores Respostas Estratégicas** do jogador i ao perfil estratégico misto σ_{-i} dos outros jogadores é:

$$BR_i(\sigma_{-i}) = \arg \max_{\sigma'_i \in \Sigma_i} EU - ex - ante_i(\sigma'_i, \sigma_{-i}). \quad (6.4.1)$$

Se $\sigma_i \in BR_i(\sigma_{-i})$, então

$$EU - ex - ante_i(\sigma_i, \sigma_{-i}) \geq EU - ex - ante_i(\sigma''_i, \sigma_{-i}), \quad \forall \sigma''_i \in \Sigma_i,$$

onde Σ_i é o conjunto de todas as estratégias mistas do jogador i .

De forma equivalente, em termos da utilidade esperada *ex-interim* ,

$$\begin{aligned} \sum_{\theta_i \in \Theta_i} p(\theta_i) EU - ex-interim_i(\sigma_i, \sigma_{-i}, \theta_i) &\geq \\ \sum_{\theta_i \in \Theta_i} p(\theta_i) EU - ex - interim_i(\sigma_i'', \sigma_{-i}, \theta_i), &\quad \forall \sigma_i'' \in \Sigma_i. \end{aligned}$$

Em termos da utilidade esperada *ex-post*,

$$\begin{aligned} \sum_{\theta_i \in \Theta_i} \left(p(\theta_i) \sum_{\theta_{-i} \in \Theta_{-i}} [p(\theta_{-i} | \theta_i) EU - ex - post_i(\sigma_i, \sigma_{-i}, \theta_i, \theta_{-i})] \right) &\geq \\ \sum_{\theta_i \in \Theta_i} \left(p(\theta_i) \sum_{\theta_{-i} \in \Theta_{-i}} [p(\theta_{-i} | \theta_i) EU - ex - post_i(\sigma_i'', \sigma_{-i}, \theta_i, \theta_{-i})] \right), &\quad \forall \sigma_i'' \in \Sigma_i. \end{aligned}$$

6.4.1 Equilíbrio de Bayes-Nash

Um Equilíbrio Bayes-Nash é um perfil estratégico que, para todo jogador i , satisfaz

$$\sigma_i \in BR_i(\sigma_{-i}), \quad (6.4.2)$$

significando que cada jogador sempre escolhe a sua melhor resposta às estratégias mistas dos outros jogadores.

6.5 Jogo Induzido pelo Jogo Bayesiano

Cada jogador usa apenas estratégias puras, conforme explicado na seção sobre utilidade esperada *ex-post*. Mas, o equilíbrio é calculado usando utilidade esperada *ex-ante*, pois se fosse usado *ex-post* ou *ex-interim*, a matriz de utilidades não seria única para todos os agentes e seus potenciais tipos.

Na exposição a seguir, usaremos as seguintes convenções para estratégias: UU quando o jogador 1 escolher sempre U independente de θ_1 ser igual a $\theta_{1,1}$ ou $\theta_{1,2}$; UD quando o jogador 1 escolher U se θ_1 for igual a $\theta_{1,1}$ e escolher D se θ_1 for igual a $\theta_{1,2}$; DU quando o jogador 1 escolher D se θ_1 for igual a $\theta_{1,1}$ e U se θ_1 for igual a $\theta_{1,2}$; DD quando o jogador 1 escolher sempre D independentemente de θ_1 ser igual a $\theta_{1,1}$ ou a $\theta_{1,2}$. A mesma notação é introduzida para o jogador 2.

Como exemplo, quando as escolhas dos jogadores forem UU e LL , a utilidade do jogador 2 será:

$$\begin{aligned} U_2(UU, LL) = & p(\theta_{1,1}, \theta_{2,1})u_2(U, L, \theta_{1,1}, \theta_{2,1}) + p(\theta_{1,1}, \theta_{2,2})u_2(U, L, \theta_{1,1}, \theta_{2,2}) + \\ & p(\theta_{1,2}, \theta_{2,1})u_2(U, L, \theta_{1,2}, \theta_{2,1}) + p(\theta_{1,2}, \theta_{2,2})u_2(U, L, \theta_{1,2}, \theta_{2,2}) = \\ & 0.3(0) + 0.1(2) + 0.2(2) + 0.4(1) = 1. \end{aligned}$$

Quando as escolhas dos jogadores forem UD e LL , a utilidade será:

$$\begin{aligned} U_2(UD, LL) = & p(\theta_{1,1}, \theta_{2,1})u_2(U, L, \theta_{1,1}, \theta_{2,1}) + p(\theta_{1,1}, \theta_{2,2})u_2(U, L, \theta_{1,1}, \theta_{2,2}) + \\ & p(\theta_{1,2}, \theta_{2,1})u_2(D, L, \theta_{1,2}, \theta_{2,1}) + p(\theta_{1,2}, \theta_{2,2})u_2(D, L, \theta_{1,2}, \theta_{2,2}) = \\ & 0.3(0) + 0.1(2) + 0.2(0) + 0.4(0) = 0.2. \end{aligned}$$

No jogo induzido pelo jogo Bayesiano, as estratégias do jogador 1 pertencem ao conjunto $\{UU, UD, DU, DD\}$, e as do jogador 2 pertencem ao conjunto $\{LL, LR, RL, RR\}$. A bimatriz da Tabela 6.6 exibe todas as utilidades associadas.

	LL	LR	RL	RR
UU	2, 1	1, 0.7	1, 1.2	0, 0.9
UD	0.8, 0.2	1, 1.1	0.4, 1	0.6, 1.9
DU	1.5, 1.4	0.5, 1.1	1.7, 0.4	0.7, 0.1
DD	0.3, 0.6	0.5, 1.5	1.1, 0.2	1.3, 1.1

Tabela 6.6: Bimatriz do jogo induzido pelo jogo bayesiano.

Na Tabela 6.6, não existe Equilíbrio de Nash em estratégias puras. Mas, supondo que o perfil estratégico (DU, LL) fosse um Equilíbrio de Nash, cada jogador deveria proceder da seguinte forma: se o tipo θ_1 do jogador 1 for θ_{11} , então escolha a estratégia D ; se o tipo θ_1 do jogador 1 for θ_{12} , então o jogador escolhe a estratégia U . A resposta para o jogador 2 seria sempre escolher a estratégia L , independente do valor do tipo θ_2 ser θ_{21} ou θ_{22} .

6.6 Programação como Jogo Bayesiano

Para usar a estrutura conceitual do jogo estratégico Bayesiano, são necessários os seguintes passos: em primeiro lugar, definir estratégias como sendo os tempos de execução; em segundo lugar, definir os tipos como sendo os custos de execução; em terceiro lugar, definir as probabilidades conjuntas para os tipos; em quarto lugar, determinar o passo de discretização para as estratégias e tipos; em quinto lugar, determinar a utilidade $U_j(ED_1, \dots, ED_n, \theta_1, \dots, \theta_n)$ de cada jogador j em função das ações puras e dos tipos dos jogadores; em sexto lugar, determinar as utilidades do jogo induzido; finalmente, aplicar o conceito de equilíbrio de Nash ao jogo induzido.

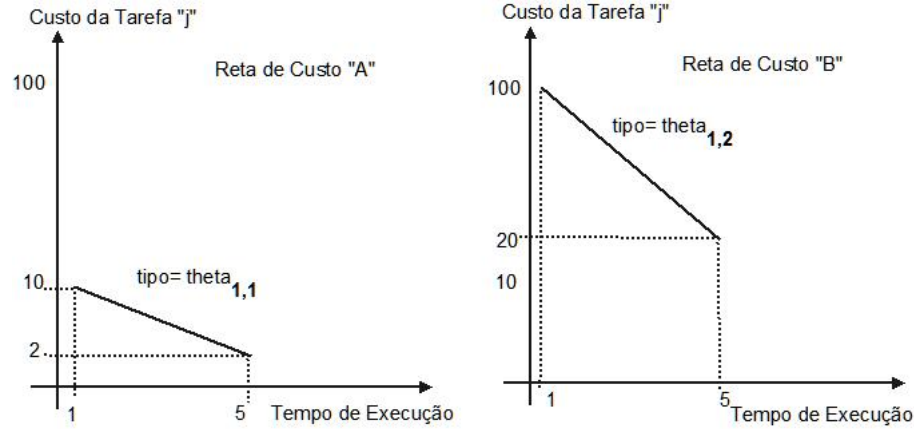
Os tipos para este jogo bayesiano são duas estruturas de custos: conservadora e agressiva. A diferença entre estes tipos está no fato que os custos das tarefas na estrutura agressiva é bem maior que os mesmos custos das tarefas na estrutura conservadora.

Ao jogador i associa-se um tipo θ_i que assume valores no conjunto de instâncias $\{\theta_{i,1}, \theta_{i,2}\}$, onde $\theta_{i,1}$ refere-se a estrutura de custos conservadora e $\theta_{i,2}$ refere-se a estrutura de custos agressiva. A Figura 6.6.1 apresenta graficamente uma situação típica com duas retas de custo de tarefa pelo tempo de execução para o jogador 1. Neste exemplo, na reta “A”, a tarefa é executada entre 1 e 5 segundos por um custo entre R\$ 2,00 e R\$ 10,00; na reta “B”, a tarefa é executada entre 1 e 5 segundos por um custo entre R\$ 20,00 e R\$ 100,00. O custo de execução de uma tarefa é muito maior no segundo tipo.

Para facilitar a exposição, apenas dois jogadores são considerados neste capítulo. Com o tipo representando apenas dois estados diferentes: conservador e agressivo, temos apenas quatro situações envolvendo tipos e jogadores, conforme apresentado na Tabela 6.7. Desta forma, a solução do problema de programação não depende do número de tarefas no processo de negócios. A interpretação do tipo $(\theta_{1,1}, \theta_{2,1})$ é a estrutura conservadora para os dois provedores; $(\theta_{1,1}, \theta_{2,2})$ indica que o provedor 1 é conservador e o provedor 2 é agressivo; $(\theta_{1,2}, \theta_{2,1})$ indica que o provedor 1 é agressivo e o provedor 2 é conservador; $(\theta_{1,2}, \theta_{2,2})$ indica que ambos os provedores são agressivos. Cada combinação de tipos (θ_1, θ_2) possui uma certa probabilidade conjunta $p(\theta_1, \theta_2)$, por hipótese, conhecido de ambos os jogadores.

Tipo	Probabilidade
$(\theta_{1,1}, \theta_{2,1})$	0.25
$(\theta_{1,1}, \theta_{2,2})$	0.25
$(\theta_{1,2}, \theta_{2,1})$	0.25
$(\theta_{1,2}, \theta_{2,2})$	0.25

Tabela 6.7: Tipos e probabilidades.

Figura 6.6.1: Estrutura de custo das tarefas em função dos θ_1 do provedor 1.

6.7 Resultados Numéricos

6.7.1 Exemplo 3

Este exemplo corresponde ao mesmo Exemplo 3, descrito da Figura 4.8.2 do Capítulo 4, o processo de negócios possui cinco tarefas não sequenciais, ou seja, existe uma decisão, representada pelo losango.

Naquele exemplo os tempos mínimos e máximos de execução possuem apenas um conjunto de valores, independentes de tipos, conforme Tabela 6.8.

Tempos	tarefa 1		tarefa 2		tarefa 3		tarefa 4		tarefa 5	
	mín	máx	mín	máx	mín	máx	mín	máx	mín	máx
provedor 1	15	20	14	19	15	16	16	18	6	8
provedor 2	14	21	13	18	16	18	10	11	1	5

Tabela 6.8: Tempos de execução do Exemplo 3.

Assumindo que os preços das tarefas apresentam dois conjuntos distintos de parâmetros, um para cada valor de θ_1 e θ_2 , conforme apresentado na Tabela 6.9.

Preços	tarefa 1		tarefa 2		tarefa 3		tarefa 4		tarefa 5	
provedor 1	mín	máx	mín	máx	mín	máx	mín	máx	mín	máx
$\theta_{1,1}$	1,50	2,50	0,20	2,00	0,85	1,25	0,02	1,00	0,33	0,66
$\theta_{1,2}$	15,00	25,00	2,00	22,00	8,50	12,50	2,00	10,00	3,30	6,60
Preços	tarefa 1		tarefa 2		tarefa 3		tarefa 4		tarefa 5	
provedor 2	mín	máx	mín	máx	mín	máx	mín	máx	mín	máx
$\theta_{2,1}$	1,30	1,50	1,70	1,80	0,75	1,22	0,25	0,75	0,05	0,25
$\theta_{2,2}$	13,00	15,00	17,00	18,00	7,50	12,20	2,50	7,50	5,00	25,00

Tabela 6.9: Preços das tarefas em função de θ Exemplo 3.

índice	estrategia provedor 1	índice	estrategia provedor 2
14	(17,5; 16,5; 15,5; 16,0; 6,0)	45	(21,0; 18,0; 17,0; 10,5; 1,0)
226	(15,0; 14,0; 15,5; 18,0; 8,0)	201	(21,0; 13,0; 17,0; 10,5; 5,0)
30	(20,0; 14,0; 15,0; 17,0; 6,0)	130	(14,0; 15,5; 18,0; 10,5; 3,0)
99	(20,0; 19,0; 15,5; 16,0; 7,0)	187	(14,0; 18,0; 18,0; 10,0; 5,0)
144	(20,0; 19,0; 15,0; 18,0; 7,0)	230	(17,5; 15,5; 17,0; 11,0; 5,0)

Tabela 6.10: Ações reduzidas Exemplo 3.

A probabilidade conjunta $p(\theta_1, \theta_2)$ utilizada foi 0.25 para todos os possíveis valores de (θ_1, θ_2) , conforme descreve a Tabela 6.7.

Um programa de apoio Java foi usado com a especificação das Tabelas 6.8 e 6.9 para gerar um subconjunto aleatório de 10 ações, conforme apresentado na Tabela 6.10.

O programa de apoio também foi usado para encontrar as bimatrizes de utilidades nas Tabelas 6.11, 6.12, 6.13 e 6.14.

	45	201	130	187	230
14	1.1, 2.29	1.1, 2.09	1.1, 2.27	1.1, 2.3	1.1, 2.07
226	0.01, 3.74	0.01, 3.64	0.01, 3.78	0.01, 3.63	0.01, 3.69
30	0.0, 3.99	0.0, 3.89	1.5, 2.53	1.75, 2.13	0.0, 3.82
99	0.2, 2.29	0.2, 2.09	1.7, 0.77	1.7, 0.8	0.2, 2.07
144	0.21, 2.04	0.21, 1.84	1.71, 0.52	1.71, 0.42	0.21, 1.94

Tabela 6.11: Bimatriz de utilidades para $\theta_{1,1}$ e $\theta_{2,1}$ Exemplo 3

	45	201	130	187	230
14	4.79, 0.0	4.79, 0.0	4.79, 0.0	4.79, 0.0	4.79, 0.0
226	5.36, 0.0	5.36, 0.0	5.36, 0.0	5.36, 0.0	5.36, 0.0
30	5.04, 0.0	5.04, 0.0	5.04, 0.0	5.04, 0.0	5.04, 0.0
99	3.22, 0.0	3.22, 0.0	3.22, 0.0	3.22, 0.0	3.22, 0.0
144	2.83, 0.0	2.83, 0.0	2.83, 0.0	2.83, 0.0	2.83, 0.0

Tabela 6.12: Bimatriz de utilidades para $\theta_{1,1}$ e $\theta_{2,2}$ Exemplo 3

	45	201	130	187	230
14	0.0, 3.99	0.0, 3.89	0.0, 4.03	0.0, 4.0	0.0, 3.82
226	0.0, 3.99	0.0, 3.89	0.0, 4.03	0.0, 4.0	0.0, 3.82
30	0.0, 3.99	0.0, 3.89	0.0, 4.03	0.0, 4.0	0.0, 3.82
99	0.0, 3.99	0.0, 3.89	0.0, 4.03	0.0, 4.0	0.0, 3.82
144	0.0, 3.99	0.0, 3.89	0.0, 4.03	0.0, 4.0	0.0, 3.82

Tabela 6.13: Bimatriz de utilidades para $\theta_{1,2}$ e $\theta_{2,1}$ Exemplo 3

	45	201	130	187	230
14	18.6, 20.43	12.0, 25.43	18.6, 21.25	12.0, 27.5	12.0, 25.18
226	4.3, 34.92	4.3, 35.92	4.3, 36.25	4.3, 35.75	4.3, 36.42
30	6.6, 37.42	0.0, 43.42	21.6, 23.75	18.0, 25.75	0.0, 42.67
99	6.95, 20.43	6.95, 20.43	21.95, 6.25	21.95, 7.5	6.95, 20.18
144	7.95, 17.93	7.95, 17.93	22.95, 3.75	22.95, 3.75	7.95, 18.93

Tabela 6.14: Bimatriz de utilidades para $\theta_{1,2}$ e $\theta_{2,2}$ Exemplo 3

A Tabela 6.15 apresenta as 25 estratégias do jogo bayesiano estendido. Por exemplo, para o provedor 1, a estratégia estendida “[226,14]” indica usar a estratégia 226 se $\theta_1 = \theta_{1,1}$, e usar a estratégia 14 se $\theta_1 = \theta_{1,2}$; para o provedor 2 a estratégia estendida “[201,55]” indica usar a estratégia 201 se $\theta_2 = \theta_{2,1}$, ou usar a estratégia 55 se $\theta_2 = \theta_{2,2}$. A combinação “[226,14],[201,55]” corresponde a um perfil estratégico estendido do jogo Bayesiano.

	provedor 1	provedor 2
1	[14,14]	[45,45]
2	[226,14]	[201,45]
3	[30,14]	[130,45]
4	[99,14]	[187,45]
5	[144,14]	[230,45]
6	[14,226]	[45,201]
7	[226,226]	[201,201]
8	[30,226]	[130,201]
9	[99,226]	[187,201]
10	[144,226]	[230,201]
11	[14,30]	[45,130]
12	[226,30]	[201,130]
13	[30,30]	[130,130]
14	[99,30]	[187,130]
15	[144,30]	[230,130]
16	[14,99]	[45,187]
17	[226,99]	[201,187]
18	[30,99]	[130,187]
19	[99,99]	[187,187]
20	[144,99]	[230,187]
21	[14,144]	[45,230]
22	[226,144]	[201,230]
23	[30,144]	[130,230]
24	[99,144]	[187,230]
25	[144,144]	[230,230]

Tabela 6.15: Estrategias estendidas do Exemplo 3.

O jogo original possui 5 ações para cada provedor. No jogo Bayesiano estendido cada jogador tem 25 estratégias estendidas.

Foi obtido apenas um Equilíbrio de Bayes-Nash em estratégias mistas pelo método ARSS, conforme apresentado na Tabela 6.16. Nesta tabela, s_{191} representa a ação 19 do provedor 1 e s_{232} representa a ação 23 do provedor 2.

Com o mapeamento da Tabela 6.15, para o provedor 1 na Tabela 6.16, a ação s_{11} corresponde a [14, 14], indicando escolher a estratégia de tempo 14, ou seja, (17, 5; 16, 5; 15, 5; 16, 0; 6, 0) (Tabela 6.10), independente de $\theta_1 = \theta_{1,1}$ ou $\theta_1 = \theta_{1,2}$; a ação s_{31} corresponde a [30, 14], indicando escolher a estratégia de tempo 30, ou seja, (20, 0; 14, 0; 15, 0; 17, 0; 6, 0), se $\theta_1 = \theta_{1,1}$, e escolher a estratégia 14, ou seja, (17, 5; 16, 5; 15, 5; 16, 0; 6, 0), se $\theta_1 = \theta_{1,2}$; a ação s_{211} corresponde ao [14, 144] indicando escolher a estratégia 14, ou seja, (17, 5; 16, 5; 15, 5; 16, 0; 6, 0), se $\theta_1 = \theta_{1,1}$, ou escolher a estratégia 144, ou seja, (20, 0; 19, 0; 15, 0; 18, 0; 7, 0), se $\theta_1 = \theta_{1,2}$.

Para o provedor 2, a ação s_{62} corresponde a [45, 201] indicando escolher a estratégia 45, ou seja, (21, 0; 18, 0; 17, 0; 10, 5; 1, 0), se $\theta_2 = \theta_{2,1}$, ou escolher 201, ou seja, (21, 0; 13, 0; 17, 0; 10, 5; 5, 0), se $\theta_2 = \theta_{2,2}$; a ação s_{82} corresponde a [130, 201], indicando escolher a estratégia 130, ou seja, (14, 0; 15, 5; 18, 0; 10, 5; 3, 0), se $\theta_2 = \theta_{2,1}$ ou escolher 201, ou seja, (21, 0; 13, 0; 17, 0; 10, 5; 5, 0), se $\theta_2 = \theta_{2,2}$; a ação s_{182} corresponde a [130, 187], indicando escolher a estratégia 130, ou seja,

$(14, 0; 15, 5; 18, 0; 10, 5; 3, 0)$, se $\theta_2 = \theta_{2,1}$, ou escolher 187, ou seja, $(14, 0; 18, 0; 18, 0; 10, 0; 5, 0)$, se $\theta_2 = \theta_{2,2}$.

As Tabelas 6.17, 6.18, 6.19 e 6.20 mostram os Equilíbrios de Nash para cada combinação de $\theta_1 \in \{\theta_{1,1}, \theta_{1,2}\}, \theta_2 \in \{\theta_{2,1}, \theta_{2,2}\}$.

EE	s_{11}	s_{21}	s_{31}	s_{41}	s_{51}	EP_1
1	0.994	0.0	0.0	0.0	0.006	1,10
EE	s_{12}	s_{22}	s_{32}	s_{42}	s_{52}	EP_2
1	0.0	0.41	0.0	0.0	0.59	2,29

Tabela 6.17: Equilíbrio de Nash do jogo com $\theta_1 = \theta_{1,1}, \theta_2 = \theta_{2,1}$ pelo método ARSS Exemplo 3.

EE	s_{11}	s_{21}	s_{31}	s_{41}	s_{51}	EP_1
1	0.0	0.0	1.0	0.0	0.0	5,36
2	0.0	0.0	1.0	0.0	0.0	5,36
3	0.0	0.0	1.0	0.0	0.0	5,36
4	0.0	0.0	1.0	0.0	0.0	5,36
5	0.0	0.0	1.0	0.0	0.0	5,36
EE	s_{12}	s_{22}	s_{32}	s_{42}	s_{52}	EP_2
1	1.0	0.0	0.0	0.0	0.0	0,00
2	0.0	1.0	0.0	0.0	0.0	0,00
3	0.0	0.0	1.0	0.0	0.0	0,00
4	0.0	0.0	0.0	1.0	0.0	0,00
5	0.0	0.0	0.0	0.0	1.0	0,00

Tabela 6.18: Equilíbrios de Nash do jogo com $\theta_1 = \theta_{1,1}, \theta_2 = \theta_{2,2}$ pelo método ARSS Exemplo 3.

EE	s_{11}	s_{21}	s_{31}	s_{41}	s_{51}	EP_1
1	1.0	0.0	0.0	0.0	0.0	0,00
2	0.0	1.0	0.0	0.0	0.0	0,00
3	0.0	0.0	1.0	0.0	0.0	0,00
4	0.0	0.0	0.0	1.0	0.0	0,00
5	0.0	0.0	0.0	0.0	1.0	0,00
EE	s_{12}	s_{22}	s_{32}	s_{42}	s_{52}	EP_2
1	0.0	0.0	1.0	0.0	0.0	4,03
2	0.0	0.0	1.0	0.0	0.0	4,03
3	0.0	0.0	1.0	0.0	0.0	4,03
4	0.0	0.0	1.0	0.0	0.0	4,03
5	0.0	0.0	1.0	0.0	0.0	4,03

Tabela 6.19: Equilíbrios de Nash do jogo com $\theta_1 = \theta_{1,2}, \theta_2 = \theta_{2,1}$ pelo método ARSS Exemplo 3.

EE	s_{11}	s_{21}	s_{31}	s_{41}	s_{51}	s_{61}	s_{71}	s_{81}	$\cdots$	s_{171}	s_{181}	s_{191}	s_{201}	s_{211}	s_{221}	s_{231}	s_{241}	s_{251}	EP_1
1	0.85	0.0	0.023	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.13	0.0	0.0	0.0	0.0	4,47
EE	s_{12}	s_{22}	s_{32}	s_{42}	s_{52}	s_{62}	s_{72}	s_{82}	$\cdots$	s_{172}	s_{182}	s_{192}	s_{202}	s_{212}	s_{222}	s_{232}	s_{242}	s_{252}	EP_2
1	0.0	0.0	0.0	0.0	0.0	0.43	0.0	0.30	0.0	0.0	0.27	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7,69

Tabela 6.16: Equilíbrio Bayes-Nash do jogo estendido pelo método ARSS Exemplo 3.

EE	s_{11}	s_{21}	s_{31}	s_{41}	s_{51}	EP_1
1	0.873	0.0	0.0	0.0	0.127	12,00
EE	s_{12}	s_{22}	s_{32}	s_{42}	s_{52}	EP_2
1	0.0	0.73	0.0	0.27	0.0	24,47

Tabela 6.20: Equilíbrio de Nash do jogo com $\theta_1 = \theta_{1,2}$, $\theta_2 = \theta_{2,2}$ pelo método ARSS Exemplo 3.

6.7.2 Exemplo 4

Este exemplo é referente ao mesmo processo de negócios com seis tarefas da Figura 3.3.1 do Capítulo 3. Existem três caminhos de execução $T1 \rightarrow T2 \rightarrow T4 \rightarrow T11$, $T1 \rightarrow T2 \rightarrow T8 \rightarrow T11$ e $T1 \rightarrow T3$ com frequências 0.35, 0.35 e 0.30, respectivamente.

Os tipos (estruturas de custos), as ações e as utilidades esperadas de tipos são descritos nas Tabelas 6.21 a 6.28.

indice	estrategia provedor 1	indice	estrategia provedor 2
473	(17,5; 16,5; 15,5; 18,0; 8,0; 3,5)	209	(17,5; 13,0; 18,0; 10,5; 5,0; 3,0)
225	(20,0; 19,0; 15,0; 18,0; 8,0; 2,0)	407	(17,5; 13,0; 16,0; 10,0; 5,0; 3,5)
24	(20,0; 16,5; 16,0; 16,0; 6,0; 2,0)	299	(17,5; 13,0; 16,0; 11,0; 1,0; 3,5)
634	(15,0; 16,5; 15,5; 18,0; 7,0; 5,0)	405	(21,0; 18,0; 18,0; 11,0; 3,0; 3,5)

Tabela 6.23: Estratégias Reduzidas Exemplo 4.

	209	407	299	405
473	1.37, 1.68	1.74, 1.42	1.74, 1.47	1.37, 1.61
225	1.11, 1.68	0.14, 2.5	0.14, 2.56	0.14, 2.26
24	1.69, 1.85	1.01, 2.33	1.01, 2.22	0.72, 2.35
634	1.05, 1.68	1.41, 1.41	1.41, 1.47	1.05, 1.61

Tabela 6.24: Bimatriz de utilidades para $\theta_{1,1}$ e $\theta_{2,1}$ Exemplo 4

	209	407	299	405
473	3.84, 0.0	3.84, 0.0	3.84, 0.0	3.84, 0.0
225	3.15, 0.0	3.15, 0.0	3.15, 0.0	3.15, 0.0
24	3.84, 0.15	3.84, 0.15	4.04, 0.0	4.04, 0.0
634	4.06, 0.0	4.06, 0.0	4.06, 0.0	4.06, 0.0

Tabela 6.25: Bimatriz de utilidades para $\theta_{1,1}$ e $\theta_{2,2}$ Exemplo 4

Tempos	tarefa 1		tarefa 2		tarefa 3		tarefa 4		tarefa 5		tarefa 6	
	mín	máx	mín	máx	mín	máx	mín	máx	mín	máx	mín	máx
prov. 1	15	20	14	19	15	16	16	18	6	8	2	5
prov. 2	14	21	13	18	16	18	10	11	1	5	3	4

Tabela 6.21: Tempo de execução em segundos do Exemplo 4.

Preços	tarefa 1		tarefa 2		tarefa 3		tarefa 4		tarefa 5		tarefa 6	
prov. 1	mín	máx	mín	máx	mín	máx	mín	máx	mín	máx	mín	máx
$\theta_{1,1}$	1,50	2,50	0,20	2,00	0,85	1,25	0,02	1,00	0,33	0,66	0,50	1,50
$\theta_{1,1}$	15,00	25,00	2,00	20,00	8,50	12,50	2,00	10,00	3,30	6,60	5,00	15,00
prov. 2	mín	máx	mín	máx	mín	máx	mín	máx	mín	máx	mín	máx
$\theta_{2,1}$	1,30	1,50	1,70	1,80	0,75	1,22	0,25	0,75	0,05	0,25	0,51	1,51
$\theta_{2,2}$	13,00	15,00	17,00	18,00	7,50	12,20	2,50	7,50	0,50	2,50	5,10	15,10

Tabela 6.22: Preços de execução em R\$ do Exemplo 4.

	209	407	299	405
473	0.0, 4.0	0.0, 3.93	0.0, 3.82	0.0, 3.46
225	0.0, 4.0	0.0, 3.93	0.0, 3.82	0.0, 3.46
24	0.0, 4.0	0.0, 3.93	0.0, 3.82	0.0, 3.46
634	0.0, 4.0	0.0, 3.93	0.0, 3.82	0.0, 3.46

Tabela 6.26: Bimatriz de utilidades para $\theta_{1,2}$ e $\theta_{2,1}$ Exemplo 4

	209	407	299	405
473	14.35, 16.77	18.03, 14.15	18.03, 14.75	14.35, 16.07
225	11.75, 16.77	2.0, 24.98	2.0, 25.59	2.0, 22.64
24	16.9, 18.52	10.13, 23.34	10.13, 22.19	7.15, 23.52
634	11.1, 16.77	14.78, 14.15	14.78, 14.75	11.1, 16.07

Tabela 6.27: Bimatriz de utilidades para $\theta_{1,2}$ e $\theta_{2,2}$ Exemplo 4

	provedor 1	provedor 2
1	[473,473]	[209,209]
2	[225,473]	[407,209]
3	[24,473]	[299,209]
4	[634,473]	[405,209]
5	[473,225]	[209,407]
6	[225,225]	[407,407]
7	[24,225]	[299,407]
8	[634,225]	[405,407]
9	[473,24]	[209,299]
10	[225,24]	[407,299]
11	[24,24]	[299,299]
12	[634,24]	[405,299]
13	[473,634]	[209,405]
14	[225,634]	[407,405]
15	[24,634]	[299,405]
16	[634,634]	[405,405]

Tabela 6.28: Estratégias estendidas Exemplo 4.

Obteve-se apenas um Equilíbrio Bayes-Nash em estratégias mistas pelo método ARSS, conforme apresentado na Tabela 6.29.

Para o provedor 1 na Tabela 6.29, a ação s_{11} corresponde a [473,473] indicando escolher a estratégia 473, ou seja, (17, 5; 16, 5; 15, 5; 18, 0; 8, 0; 3, 5), independente, se $\theta_1 = \theta_{1,1}$ ou $\theta_1 = \theta_{1,2}$; a ação s_{31} corresponde a [24,473] indicando escolher a estratégia 24, ou seja, (20, 0; 16, 5; 16, 0; 16, 0; 6, 0; 2, 0), se $\theta_1 = \theta_{1,1}$, ou a estratégia 473, ou seja, (17, 5; 16, 5; 15, 5; 18, 0; 8, 0; 3, 5), se $\theta_1 = \theta_{1,2}$; a ação s_{91} corresponde a [473,24], indicando escolher a estratégia 473, ou seja, (17, 5; 16, 5; 15, 5; 18, 0; 8, 0; 3, 5), se $\theta_1 = \theta_{1,1}$, ou escolhe a estratégia 24, ou seja, (20, 0; 16, 5; 16, 0; 16, 0; 6, 0; 2, 0) se $\theta_1 = \theta_{1,2}$.

índice	estrategia provedor 1	índice	estrategia provedor 2
243	(20,0; 19,0; 16,0; 18,0; 8,0; 2,0)	547	(14,0; 18,0; 16,0; 11,0; 1,0; 4,0)
233	(17,5; 19,0; 15,5; 18,0; 8,0; 2,0)	286	(14,0; 18,0; 17,0; 10,5; 1,0; 3,5)
717	(20,0; 16,5; 15,5; 18,0; 8,0; 5,0)	546	(21,0; 15,5; 16,0; 11,0; 1,0; 4,0)
553	(15,0; 16,5; 15,5; 18,0; 6,0; 5,0)	581	(17,5; 15,5; 17,0; 10,0; 3,0; 4,0)
1	(15,0; 14,0; 15,0; 16,0; 6,0; 2,0)	693	(21,0; 18,0; 17,0; 10,5; 5,0; 4,0)

Tabela 6.30: Estratégias reduzidas Exemplo 4A.

	547	286	546	581	693
243	0.43, 1.91	1.93, 0.73	0.43, 1.71	0.43, 1.78	0.43, 1.65
233	0.5, 1.91	0.14, 2.58	0.5, 1.71	0.14, 2.12	0.14, 1.99
717	2.91, 0.07	2.55, 0.42	1.41, 1.37	1.05, 1.79	1.05, 1.66
553	1.41, 1.57	1.05, 1.92	1.41, 1.37	1.05, 1.79	1.05, 1.66
1	0.0, 3.53	0.0, 3.86	0.0, 3.36	0.0, 3.52	0.0, 3.27

Tabela 6.31: Bimatriz de utilidades para $\theta_{1,1}$ e $\theta_{2,1}$ Exemplo 4A

Para o provedor 2 na Tabela 6.29, a ação s_{12} corresponde a $[209, 209]$ indicando escolher a estratégia 209, ou seja, $(17, 5; 13, 0; 18, 0; 10, 5; 5, 0; 3, 0)$, se $\theta_2 = \theta_{2,1}$ ou $\theta_2 = \theta_{2,2}$; a ação s_{21} corresponde $[407, 209]$ indicando escolher a estratégia 407, ou seja, $(17, 5; 13, 0; 16, 0; 10, 0; 5, 0; 3, 5)$, se $\theta_2 = \theta_{2,1}$ ou a estratégia 209, ou seja, $(17, 5; 13, 0; 18, 0; 10, 5; 5, 0; 3, 0)$, se $\theta_2 = \theta_{2,2}$; a ação s_{62} corresponde a $[407, 407]$ indicando escolher a estratégia 407, ou seja, $(17, 5; 13, 0; 16, 0; 10, 0; 5, 0; 3, 5)$, se $\theta_2 = \theta_{2,1}$ ou $\theta_2 = \theta_{2,2}$.

EE	s_{11}	s_{21}	s_{31}	s_{41}	s_{51}	s_{61}	s_{71}	s_{81}	s_{91}	EP_1
1	0.17	0.0	0.47	0.0	0.0	0.0	0.0	0.0	0.34	5,14
EE	s_{12}	s_{22}	s_{32}	s_{42}	s_{52}	s_{62}	s_{72}	s_{82}	s_{92}	EP_2
1	0.69	0.066	0.0	0.0	0.0	0.24	0.0	0.0	0.0	5,80

Tabela 6.29: Exemplo 4 reduzido resolvido pelo método ARSS.

6.7.3 Exemplo 4A

Dados idênticos ao Exemplo 4, porém com as probabilidades conjuntas dos tipos alterados da seguinte forma: $p(\theta_{1,1}, \theta_{2,1}) = 0.4$, $p(\theta_{1,1}, \theta_{2,2}) = 0.3$, $p(\theta_{1,2}, \theta_{2,1}) = 0.1$ e $p(\theta_{1,2}, \theta_{2,2}) = 0.2$. As utilidades esperadas e as estratégias estendidas deste exemplo são descritas nas Tabelas 6.30 a 6.35. Foram obtidos varios equilíbrio Bayes-Nash, descritos na Tabela 6.36. Todos eles formam as mesmas utilidades esperadas, o que elimina a questão de qual equilíbrio implementar. O provedor 1 pode escolher qualquer uma das suas estratégias mistas. O provedor 2 adota a única estratégia disponível.

	547	286	546	581	693
243	3.01, 0.0	3.01, 0.0	3.01, 0.0	3.01, 0.0	3.01, 0.0
233	3.58, 0.0	3.58, 0.0	3.58, 0.0	3.58, 0.0	3.58, 0.0
717	3.01, 0.0	3.01, 0.0	3.01, 0.0	3.01, 0.0	3.01, 0.0
553	4.11, 0.0	4.11, 0.0	4.11, 0.0	4.11, 0.0	3.91, 0.15
1	5.76, 0.0	5.76, 0.0	5.76, 0.0	5.76, 0.0	5.56, 0.15

Tabela 6.32: Bimatriz de utilidades para $\theta_{1,1}$ e $\theta_{2,2}$ Exemplo 4A

	547	286	546	581	693
243	0.0, 3.53	0.0, 3.86	0.0, 3.36	0.0, 3.52	0.0, 3.27
233	0.0, 3.53	0.0, 3.86	0.0, 3.36	0.0, 3.52	0.0, 3.27
717	0.0, 3.53	0.0, 3.86	0.0, 3.36	0.0, 3.52	0.0, 3.27
553	0.0, 3.53	0.0, 3.86	0.0, 3.36	0.0, 3.52	0.0, 3.27
1	0.0, 3.53	0.0, 3.86	0.0, 3.36	0.0, 3.52	0.0, 3.27

Tabela 6.33: Bimatriz de utilidades para $\theta_{1,2}$ e $\theta_{2,1}$ Exemplo 4A

	547	286	546	581	693
243	4.97, 19.07	19.98, 7.32	4.97, 17.07	4.97, 17.77	4.97, 16.47
233	5.67, 19.07	2.0, 25.76	5.67, 17.07	2.0, 21.21	2.0, 19.91
717	29.78, 0.75	26.1, 4.2	14.78, 13.75	11.1, 17.9	11.1, 16.6
553	14.78, 15.75	11.1, 19.2	14.78, 13.75	11.1, 17.9	11.1, 16.6
1	0.0, 35.26	0.0, 38.56	0.0, 33.59	0.0, 35.21	0.0, 32.71

Tabela 6.34: Bimatriz de utilidades para $\theta_{1,2}$ e $\theta_{2,2}$ Exemplo 4A

	provedor 1	provedor 2
1	[243,243]	[547,547]
2	[233,243]	[286,547]
3	[717,243]	[546,547]
4	[553,243]	[581,547]
5	[1,243]	[693,547]
6	[243,233]	[547,286]
7	[233,233]	[286,286]
8	[717,233]	[546,286]
9	[553,233]	[581,286]
10	[1,233]	[693,286]
11	[243,717]	[547,546]
12	[233,717]	[286,546]
13	[717,717]	[546,546]
14	[553,717]	[581,546]
15	[1,717]	[693,546]
16	[243,553]	[547,581]
17	[233,553]	[286,581]
18	[717,553]	[546,581]
19	[553,553]	[581,581]
20	[1,553]	[693,581]
21	[243,1]	[547,693]
22	[233,1]	[286,693]
23	[717,1]	[546,693]
24	[553,1]	[581,693]
25	[1,1]	[693,693]

Tabela 6.35: Estrategias estendidas Exemplo 4A.

EE	s_{11}	$\cdots$	s_{131}	s_{141}	s_{151}	s_{161}	s_{171}	s_{181}	s_{191}	$\cdots$	s_{251}	EP_1
1	0.0	0.0	0.22	0.78	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5,16
2	0.0	0.0	0.0	0.78	0.0	0.0	0.0	0.22	0.0	0.0	0.0	5,16
3	0.0	0.0	0.22	0.0	0.0	0.0	0.0	0.0	0.78	0.0	0.0	5,16
4	0.0	0.0	0.09	0.0	0.0	0.0	0.0	0.135	0.78	0.0	0.0	5,16
5	0.0	0.0	0.0	0.09	0.0	0.0	0.0	0.22	0.69	0.0	0.0	5,16
EE	s_{12}	$\cdots$	s_{132}	s_{142}	s_{152}	s_{162}	s_{172}	s_{182}	s_{192}	$\cdots$	s_{252}	EP_2
1	0.0	0.0	0.0	0.0	0.0	0.0	0.24	0.0	0.76	0.0	0.0	8,40
2	0.0	0.0	0.0	0.0	0.0	0.0	0.24	0.0	0.76	0.0	0.0	8,40
3	0.0	0.0	0.0	0.0	0.0	0.0	0.24	0.0	0.76	0.0	0.0	8,40
4	0.0	0.0	0.0	0.0	0.0	0.0	0.24	0.0	0.76	0.0	0.0	8,40
5	0.0	0.0	0.0	0.0	0.0	0.0	0.24	0.0	0.76	0.0	0.0	8,40

Tabela 6.36: Exemplo 4A reduzido resolvido pelo método de ARSS.

Programação de Serviços Web por Projeto de Mecanismos

Um mecanismo, no contexto de Teoria de Jogos, pode ser visto como uma instituição ou conjunto de protocolos que prescreve modos particulares de interação entre agentes, de forma a assegurar um resultado socialmente desejado desta interação.

Neste capítulo, o planejador de serviços web assume o papel de planejador social interessado em implementar um conjunto de protocolos que resulte em uma alocação de tarefas socialmente desejável. Como os agentes (provedores) possuem informações privadas a respeito de suas preferências, o planejador precisa extrair (revelar) estas informações. Uma vez extraídas, as informações sobre preferências precisam ser transformadas em um resultado, papel que será desempenhado por uma função de escolha social.

7.1 Mecanismos de Revelação

Nesta seção serão apresentados alguns resultados teóricos sobre projeto de mecanismos, com ênfase em dois tipos de mecanismos de revelação direto e indireto (extração de preferências). O cenário para a formulação de problemas de projeto de mecanismos é o seguinte:

- $N = \{1, 2, \dots, n\}$ é um conjunto de n agentes, racionais e inteligentes;
- X é um dentre alternativas ou resultados. Os agentes devem fazer uma escolha coletiva no conjunto X ;
- Antes de fazer uma escolha coletiva, cada agente privadamente observa suas preferências sobre o conjunto de alternativas em X . Isto é modelado pela suposição de que o agente i privadamente observa um parâmetro ou sinal θ_i que determina suas preferências. O valor de θ_i é conhecido pelo agente i e não pelos outros; θ_i é chamado de tipo do agente i ;
- denota-se por Θ_i o conjunto de valores privados do agente i . O conjunto de todos os perfis de tipos é dado por $\Theta = \Theta_1 \times \dots \times \Theta_n$. Um perfil típico de tipos é representado por $\theta = (\theta_1, \dots, \theta_n)$;
- é assumido que existe uma distribuição comum a priori $\Psi \in \Delta(\theta)$ de tipos. Para manter consistência de crenças, funções de crença individuais p_i que descrevem as crenças que o jo-

gador i tem sobre os tipos dos demais dos jogadores pode ser derivado de uma distribuição a priori;

- agentes individuais têm preferências sobre resultados que são representadas pela função utilidade $u_i : X \times \Theta_i \rightarrow R$. Dado que $x \in X$ e $\theta_i \in \Theta_i$, o valor de $u_i(x, \theta_i)$ denota a utilidade que o agente i , tendo tipo θ_i , recebe de uma decisão x . No caso mais geral, u_i depende não somente dos resultados e tipos do jogador i , mas também de todos os tipos dos outros jogadores, e assim $u_i : X \times \Theta \rightarrow R$. Restringe-se a atenção no caso anterior;
- O conjunto de resultados X , o conjunto de jogadores N , os conjuntos de tipos $\Theta_i (i = 1, \dots, n)$, a distribuição comum a priori $\Psi \in \Delta(\theta)$, e as funções utilidade $u_i (i = 1, \dots, n)$ são assumidas de conhecimento comum entre todos os jogadores. O valor específico θ_i observado pelo agente i é informação privada do agente i .

Definição. Função de Escolha Social

Dado um conjunto de agentes $N = \{1, \dots, n\}$, seus conjuntos de tipos $\Theta_1, \dots, \Theta_n$, um conjunto de resultados X , uma função de escolha social é um mapeamento $f : \Theta_1 \times \dots \times \Theta_n \rightarrow X$ que atribui a cada possível perfil de *tipos* $(\theta_1, \dots, \theta_n)$ uma escolha coletiva no conjunto X .

Exemplo. Compra de Produto de dois Fornecedores

Considere uma situação em que exista um agente comprador, identificado por 0 e dois agentes vendedores, identificados por 1 e 2, de um produto de interesse. Neste caso, $N = \{0, 1, 2\}$. Os vendedores conhecem os verdadeiros valores do produto, indicados por θ_1 e θ_2 , com $\theta_i \in [0, 1]$. O agente 0 decide de qual agente deverá comprar, baseado nos valores apresentados, $\hat{\theta}_1$ e $\hat{\theta}_2$, com $\hat{\theta}_i \in [0, 1]$.

O agente 0 faz uma transferência monetária t_i ao agente i . O resultado é representado pelo vetor $x = (y_0, y_1, y_2, t_0, t_1, t_2)$.

Para o agente 0, temos:

$y_0 = 0$, se o agente comprador compra o produto, ou $y_0 = 1$, caso contrário; t_0 é a quantia monetária recebida pelo agente 0. O agente 0 recebe $-t_i$ se pagar $t_i > 0$ pelo produto ao agente i .

Para os agentes 1 e 2, temos:

$y_i = 1$, se o agente i fornece o produto ao comprador, ou $y_i = 0$, caso contrário; t_i é a quantia recebida pelo agente i .

O conjunto dos resultados viáveis é dado por

$$X = \{(y_0, y_1, y_2, t_0, t_1, t_2) | y \in \{0, 1\}, y_0 + y_1 + y_2 = 1, t_0 + t_1 + t_2 \leq 0\}.$$

A utilidade de cada agente vendedor é definida por

$$u_i(x, \theta_i) = u_i(y_0, y_1, y_2, t_0, t_1, t_2; \theta_i) = -y_i \theta_i + t_i, \quad i = 1, 2.$$

Uma possível função de escolha social para esta situação seria a modelada pelo chamado

Leilão do Primeiro Preço $f(\theta) = (y_0(\theta), y_1(\theta), y_2(\theta), t_0(\theta), t_1(\theta), t_2(\theta))$:

$$\begin{aligned} y_0(\theta) &= 0, & \forall \theta \\ y_1(\theta) &= 1, & \theta_1 \leq \theta_2 \\ y_1(\theta) &= 0, & \theta_1 > \theta_2 \\ y_2(\theta) &= 1, & \theta_2 \leq \theta_1 \\ y_2(\theta) &= 0, & \theta_2 > \theta_1 \\ t_1(\theta) &= \theta_1 y_1(\theta), \\ t_2(\theta) &= \theta_2 y_2(\theta), \\ t_0(\theta) &= -(\theta_1 y_1(\theta) + \theta_2 y_2(\theta)). \end{aligned}$$

Vence o leilão o vendedor que oferecer o menor preço.

Para outra situação modelada no Leilão de Segundo Preço, a expressão abaixo seria uma opção de $f(\theta)$:

$$\begin{aligned} y_0(\theta) &= 0, & \forall \theta \\ y_1(\theta) &= 1, & \theta_1 \leq \theta_2 \\ y_1(\theta) &= 0, & \theta_1 > \theta_2 \\ y_2(\theta) &= 1, & \theta_2 \leq \theta_1 \\ y_2(\theta) &= 0, & \theta_2 > \theta_1 \\ t_1(\theta) &= \theta_2 y_1(\theta), \\ t_2(\theta) &= \theta_1 y_2(\theta), \\ t_0(\theta) &= -(\theta_1 y_1(\theta) + \theta_2 y_2(\theta)). \end{aligned}$$

Vence o leilão o vendedor que oferecer o menor preço, recebendo entretanto o maior preço.

Para uma terceira situação, mantendo a alocação anterior, mas mudando os pagamentos, a função $f(\theta)$ seria:

$$\begin{aligned} t_1(\theta) &= \frac{(1 + \theta_1)}{2} y_1(\theta), \\ t_2(\theta) &= \frac{(1 + \theta_2)}{2} y_2(\theta), \\ t_0(\theta) &= -(t_1(\theta) + t_2(\theta)). \end{aligned}$$

■

Como os tipos θ representam informações privadas dos agentes ou jogadores, nem sempre será do interesse destes agentes revelar os seus verdadeiros tipos. Com isto tem-se um problema de extração de preferências, na forma de tipos.

Existem duas formas principais para os mecanismos de extração de tipos: mecanismos indiretos e diretos. Na definição a seguir, assumimos que o conjunto de agentes N , o conjunto de resultados X , os conjuntos de tipos $\Theta_1, \dots, \Theta_n$, uma distribuição de probabilidades com sobre $\Theta = \Theta_1 \times \dots \times \Theta_n$ e as funções utilidades $u_i : X \times \Theta_i \rightarrow R$ são dadas e de conhecimento comum dos agentes.

Definição. Mecanismo Direto

Dado uma função de escolha social $f : \Theta_1 \times \dots \times \Theta_n \rightarrow X$, um mecanismo de revelação direta consiste da tupla $(\Theta_1, \dots, \Theta_n, f(\theta))$.

Definição. Mecanismo Indireto

Um mecanismo de revelação indireto consiste de uma tupla $(S_1, \dots, S_n, g(s))$, onde S_i é o conjunto de todas as possíveis ações para o agente i ($i = 1, \dots, n$) e $g : S_1 \times \dots \times S_n \rightarrow X$ é uma função que mapeia cada perfil estratégico em um resultado em X . A idéia em um mecanismo indireto é obter de cada agente uma escolha dentre um conjunto de ações, e então especificar um resultado para cada perfil de ações selecionado. Desta forma, induz-se um “jogo” entre os agentes, e as estratégias adotadas pelos jogadores em um equilíbrio deste jogo refletirão indiretamente os seus tipos privados.

7.2 Implementação da Função de Escolha Social por Mecanismos Indiretos

Um função de escolha social é implementável se a sua adoção torna ser de interesse dos agentes revelarem os seus tipos. Por um mecanismo de revelação direta, a implementação seria:

1. cada agente revela o seu tipo Θ_i ;
2. dados os tipos anunciados $(\hat{\theta}_1, \dots, \hat{\theta}_n)$, escolhe-se o resultado $x = f(\hat{\theta}_1, \dots, \hat{\theta}_n) \in X$.

Uma abordagem alternativa seria implementar a função de escolha social de forma indireta. Para a descrição desta abordagem alternativa, faz-se necessário introduzir o conceito de estratégia fracamente dominada.

Definição. Estratégia Fracamente Dominada.

Para um jogo $\Gamma = (N, (S_i)_{i \in N}, (u_i)_{i \in N})$, uma estratégia $s_i \in S_i$ é fracamente dominada por uma estratégia $s'_i \in S_i$ do jogador i se $u_i(s'_i, s_{-i}) \geq u_i(s_i, s_{-i}) \forall s_{-i} \in S_{-i}$ e $u_i(s'_i, s_{-i}) > u_i(s_i, s_{-i})$ para algum s_{-i} .

Definição. Estratégia Fracamente Dominante.

Uma estratégia s_i^* é uma estratégia fracamente dominante do jogador i se s_i^* domina fracamente qualquer outra estratégia $s_i \in S_i$.

Definição. Equilíbrio em Estratégias Fracamente Dominantes.

Dado um jogo $\Gamma = (N, (S_i)_{i \in N}, (u_i)_{i \in N})$, um perfil estratégico $s^* = (s_1^*, \dots, s_n^*)$ é um equilíbrio em estratégias fracamente dominantes se para $i = 1, \dots, n$ a estratégia s_i^* é uma estratégia fracamente dominante do jogador i .

Passamos agora a implementação da função de escolha social em termos de estratégias dominantes.

Definição. Implementação de Função de Escolha Social em Estratégias Dominantes.

Dizemos que um mecanismo $M = (S_1, \dots, S_n, g(s_1, \dots, s_n))$ implementa a função de escolha social $f(\theta)$ em equilíbrio em estratégias dominantes se existe um equilíbrio em estratégias fracamente dominante $s^*(\theta) = (s_1^*(\theta), \dots, s_n^*(\theta))$ do jogo Γ^b induzido por M tal que

$$g(s_1^*(\theta), \dots, s_n^*(\theta)) = f(\theta_1, \dots, \theta_n) \forall (\theta_1, \dots, \theta_n) \in \Theta.$$

7.3 Compatibilidade de Incentivo e o Teorema da Revelação

A noção de compatibilidade de incentivo e o Teorema da Revelação formam a base para o estudo de Projeto de Mecanismos.

O Projeto de Mecanismos envolve os problemas de extrair (revelar) preferências e de agregar preferências em uma função de escolha social. Para extrair informações dos agentes sobre os seus tipos de forma confiável, torna-se necessário fazer com que a revelação confiável dos tipos seja a melhor resposta dos agentes, e isto seja compatível com a racionalidade e a inteligência dos agentes.

Definição. Compatibilidade de Incentivo em Estratégias Dominantes (CIED)

Uma função de escolha social $f : \Theta_1 \times \dots \times \Theta_n \rightarrow X$ é incentivo-compatível em estratégias dominantes (ou verdadeiramente implementável em estratégias dominantes) se o mecanismo de revelação direta $D = (\Theta_1, \dots, \Theta_n, f(\theta))$ tem um equilíbrio em estratégias fracas dominantes $s^*(\theta) = (s_1^*(\theta), \dots, s_n^*(\theta))$ no qual $s_i^*(\theta_i) = \theta_i$. A condição necessária e suficiente para uma função de escolha social $f(\cdot)$ ser incentivo-compatível em estratégias dominantes é:

$$u_i(f(\theta_i, \theta_{-i}), \theta_i) \geq u_i(f(\hat{\theta}_i, \theta_{-i}), \theta_i), \forall i \in N, \forall \theta_i, \hat{\theta}_i \in \Theta_i, \forall \theta_{-i} \in \Theta_{-i}. \quad (7.3.1)$$

O Princípio da Revelação para equilíbrios em estratégias dominantes estabelece uma relação entre os mecanismos direto e indireto de uma função de escolha social, e garante que apenas o mecanismo de revelação direta seja necessário. Os dois teoremas a seguir são importantes para a caracterização do leilão Vickrey-Clarke-Groves (Narahari, Narayanam, Garg & Prakash 2009).

Teorema. *Teorema da Revelação em Estratégias Dominantes.*

Suponha que exista um mecanismo $M = (S_1, \dots, S_n, g(\theta))$ que implementa a função de escolha social $f(\cdot)$ por equilíbrio em estratégias dominantes. Então $f(\cdot)$ é incentivo-compatível em estratégias dominantes.

Demonstração. Se $M = (S_1, \dots, S_n, g(\cdot))$ implementa $f(\theta)$ em estratégias dominantes, então existe um perfil estratégico $s^*(\cdot) = (s_1^*(\cdot), \dots, s_n^*(\cdot))$ tal que

$$g(s_1^*(\theta_1), \dots, s_n^*(\theta_n)) = f(\theta_1, \dots, \theta_n) \quad \forall (\theta_1, \dots, \theta_n) \in \Theta \quad (7.3.2)$$

e

$$\begin{aligned} u_i(g(s_i^*(\theta_i), s_{-i}(\theta_{-i}), \theta_i) &\geq u_i(g(s_i'(\theta_i), s_{-i}(\theta_{-i}), \theta_i) \\ \forall i \in N, \forall \theta_i \in \Theta_i, \forall \theta_{-i} \in \Theta_{-i}, \forall s_i'(\cdot) \in S_i, \forall s_{-i} \in S_{-i}. \end{aligned} \quad (7.3.3)$$

A condição (7.3.3) implica, em particular, que (trocando-se s_{-i} por s_{-i}^* e trocando-se $s_i'(\theta_i)$ por $s_i(\hat{\theta}_i)$)

$$\begin{aligned} u_i(g(s_i^*(\theta_i), s_{-i}^*(\theta_{-i}), \theta_i) &\geq u_i(g(s_i(\hat{\theta}_i), s_{-i}^*(\theta_{-i}), \theta_i) \\ \forall i \in N, \forall \theta_i \in \Theta_i, \forall \hat{\theta}_i \in \Theta_i, \forall \theta_{-i} \in \Theta_{-i}. \end{aligned} \quad (7.3.4)$$

As condições (7.3.3) e (7.3.4) juntas implicam que

$$u_i(f(\theta_i, \theta_{-i}), \theta_i) \geq u_i(f(\hat{\theta}_i, \theta_{-i}), \theta_i), \forall i \in N, \forall \theta_i \in \Theta_i, \forall \hat{\theta}_i \in \Theta_i, \forall \theta_{-i} \in \Theta_{-i}.$$

Mas isto é precisamente a condição (7.3.1), ou seja, a condição de $f(\cdot)$ ser verdadeiramente implementável em estratégias dominantes. $\square$

No restante deste capítulo, assumiremos que o projeto de mecanismos se dá no chamado ambiente quasi-linear. Neste ambiente, uma alternativa $x \in X$ é um valor da forma $x = (k, t_1, \dots, t_n)$, em que k é um elemento de um conjunto K , chamado de conjunto de escolhas de projetos ou conjunto de alocação; K é geralmente um conjunto finito. O termo $t_i \in R$ representa a transferência monetária ao agente i . Se $t_i > 0$, o agente i recebe t_i ; e $t_i < 0$, o agente i paga t_i . Assume-se que $\sum_{i=1}^n t_i \leq 0$, significando que não existe fonte externa de financiamento, e que a soma das quantias transferidas pode ser menor que zero, situação em que haveria uma perda para um elemento intermediário. O conjunto de alternativas a ser considerado é:

$$X = \{(k, t_1, \dots, t_n) | k \in K, t_i \in R, \sum t_i \leq 0\}.$$

A função de escolha social no ambiente quasi-linear assume a forma

$$f(\theta) = (k(\theta), t_1(\theta), \dots, t_n(\theta)),$$

onde $k(\theta) \in K$ e $\sum t_i(\theta) \leq 0$ para todo $\theta \in \Theta$. Para um mecanismo de revelação direta $D = ((\theta_i)_{i \in N}, f(\cdot))$, a função utilidade do agente i assume a forma quasi-linear

$$u_i(x, \theta_i) = u_i(k, t_1, \dots, t_n; \theta_i) = v_i(k, \theta_i) + m_i + t_i,$$

onde m_i é a dotação inicial e $v_i(\cdot)$ é a função de valoração do agente i . As funções $v_i(\cdot, \theta_i)$ são de conhecimento comum de todos os jogadores.

Exemplo. Considere o problema de um projeto público com dois agentes, $N = \{1, 2\}$. Assuma que o custo do projeto é de 50 unidades monetárias, e que os tipos dos agentes são

$$\Theta_1 = \Theta_2 = \{20, 60\}.$$

Cada agente possui uma baixa disposição de pagamento (20), ou uma alta disposição de pagamento (60). O conjunto de escolhas do projeto são

$$K = \{0, 1\},$$

com 0 indicando que o projeto será abandonado, e com 1 indicando que o projeto será desenvolvido. Assume-se que se $k = 1$, então os dois agentes dividirão igualmente o custo do projeto, pagando 25 cada. Se $k = 0$, os agentes não pagam nada. Uma forma possível para as funções de valoração seria

$$v_i(k, \theta_i) = k(\theta_i - 25).$$

A função de alocação poderia ser definida como

$$\begin{aligned} k(\theta_1, \theta_2) &= 0, \quad \theta_1 = \theta_2 = 20, \\ &= 1, \quad \text{caso contrário.} \end{aligned}$$

A Tabela 7.1 resume os valores de $v_1(k, \theta_1)$ e $v_2(k, \theta_2)$ para diferentes alocações pela função $k(\theta_1, \theta_2)$. Da tabela, infere-se que $k(\theta_1, \theta_2)$ maximiza a soma $v_1(k, \theta_1) + v_2(k, \theta_2)$ para os tipos $\theta_1 \in \Theta_1$ e $\theta_2 \in \Theta_2$.

(θ_1, θ_2)	k	$v_1(0, \theta_1)$	$v_2(0, \theta_2)$	$v_1(1, \theta_1)$	$v_2(1, \theta_2)$
(20,20)	0	0	0	-5	-5
(20,60)	1	0	0	-5	35
(60,20)	1	0	0	35	-5
(60,60)	1	0	0	35	35

Tabela 7.1: Exemplo.

Definição. Eficiência Alcativa (EA)

Uma função de escolha social $f(\theta) = (k(\theta), t_1(\theta), \dots, t_n(\theta))$ é alcativamente eficiente se, para cada $\theta \in \Theta$, a alocação $k(\theta)$ satisfaz a seguinte condição:

$$k(\theta) \in \arg \max_{k \in K} \sum_{i=1}^n v_i(k, \theta_i). \quad (7.3.5)$$

Esta definição implica que para todo $\theta \in \Theta$, a alocação $k(\theta)$ irá maximizar a soma dos valores dos jogadores. Cada alocação é um valor maximizante: os objetos são alocados aos jogadores que mais valorizam os objetos.

Teorema. Teorema de Groves.

Seja uma função de escolha social $f(\theta) = (k^*(\theta), t_1(\theta), \dots, t_n(\theta))$ alcativamente eficiente. Então $f(\cdot)$ é incentivo-compatível em estratégias dominantes se satisfaz a seguinte estrutura de pagamento, chamado pagamento Groves:

$$t_i(\theta) = \left[\sum_{j \neq i} v_j(k^*(\theta), \theta_j) \right] + h_i(\theta_{-i}), \quad (7.3.6)$$

onde $h_i : \Theta_{-i} \rightarrow R$ é qualquer função que garante a condição de factibilidade $\sum_i t_i(\theta) \leq 0$.

Demonstração. Ver (Narahari et al. 2009). □

O teorema anterior será aplicado na próxima seção para a resolução do problema de programação de serviços web.

7.4 Mecanismo de Vickrey-Groves-Clarke (VGC)

O chamado Mecanismo de Vickrey-Groves-Clarke produz uma solução em estratégias dominantes para o problema de projeto de mecanismos. Nesta seção, o mecanismo VGC será aplicado ao problema de programação de serviços web.

Considere um processo de negócios com m tarefas podendo ser executadas por n provedores, ou seja, as tarefas podem ser escolhidos do conjunto $A = \{1, \dots, m\}$ e os provedores podem ser escolhidos do conjunto $N = \{1, \dots, n\}$. Neste cenário, o **comprador** é o dono do processo de negócios e os **vendedores** são os provedores de serviços web. O leilão ocorre em benefício do comprador, e os vendedores competem entre si para ganhar o direito de executar uma ou mais tarefas do processo de negócios. O leilão envolve os seguintes elementos de modelagem.

Alocação y_{ij}

Atribui-se $y_{ij} = 1$, se o provedor i executa a tarefa j e $y_{ij} = 0$, caso contrário. O vetor y contém todas as variáveis y_{ij} e tem o papel similar à variável de alocação k .

Pagamento p_{ij} e $p_i(y)$

O valor p_{ij} é o pagamento realizado pelo sistema de leilão ao provedor i pela execução da tarefa j .

O total de pagamentos recebidos pelo provedor i é $p_i = \sum_{j=1}^m p_{ij}y_{ij}$.

Custo real de execução EC_{ij} e $EC_i(y)$

O custo de executar a tarefa j pelo provedor i é EC_{ij} . O custo de todas as tarefas executadas pelo provedor i é $EC_i = \sum_{j=1}^m EC_{ij}y_{ij}$.

Orçamento v_0

Este é o valor que o proprietário dispõe para o processo de negócios.

Utilidade dos provedores $u_i(y, p_i)$

A utilidade $u_i(y, p_i) = p_i(y) - EC_i(y)$ é o valor líquido entre o valor total pago p_i pelo sistema de leilão e o valor total de custo EC_i .

Regra de Alocação Vickrey y_{ij}

Para satisfazer o requisito de eficiência, devemos obter a solução com o máximo valor para os provedores.

Dado v_0 , um orçamento disponível, e $\sum_{i=1}^n EC_i(y)$, a soma de todos os custos de execução de todos os provedor i , a sua diferença é a sobra em recursos monetários para o proprietário do processo de negócios. A alocação ótima y_{ij}^* é obtida resolvendo-se o problema de otimização (7.4.1), onde $freq_l$ é a frequência de execução e EC_l custo do caminho de execução l :

$$y_{ij}^* = \arg \left| \begin{array}{l} \max_{y_{ij}} \\ s.a \end{array} \right| \begin{array}{l} [v_0 - \sum_{l \in L} freq_l EC_l] \\ restrições do problema. \end{array} \quad (7.4.1)$$

Regra de Pagamento Vickrey $p_i(y)$

A regra de pagamento é implementada modificando-se o modelo com preços do plano ótimo y_{ij}^* para valores não competitivos. O pagamento aos vencedores deve ser a segunda menor diferença entre receita e custo total:

$$\left| \begin{array}{l} \max_{y_{ij}} \\ s.a \end{array} \right| \begin{array}{l} [v_0 - \sum_{i,j \notin \{y_{ij}^*\}, l \in L} EC_l] \\ restrições do problema. \end{array} \quad (7.4.2)$$

A regra de Vickrey satisfaz o requisito de racionalidade individual

$$u_i(y, p_i) \geq 0, \quad (7.4.3)$$

ou seja, o pagamento sempre excede o custo interno.

A regra também é incentivo compatível, pois demonstra-se

$$u_i(y, p_i) \geq u_i(y, \hat{p}_i),$$

ou seja, a utilidade para p_i , calculado por (7.4.2), é sempre maior que a utilidade para qualquer outro $\hat{p}_i$.

7.5 Resultados Numéricos

Considere uma versão simplificada do problema, no geral o processo de negócios contém apenas uma única tarefa. Também considere que não haja restrições de tempo, reputação, confiabilidade e disponibilidade.

Sejam quatro provedores $\{1, \dots, 4\}$ com custos em R\$ de execução: $TD_{11} = 20$, $TD_{21} = 21$, $TD_{31} = 19$, $TD_{41} = 18$. O orçamento em R\$ para este serviço web é $v_0 = 25$. De acordo com a regra (7.4.1), o provedor vencedor é computado por meio de $\max_{1 \leq i \leq 4} \{25 - TD_{i1}\} = \max\{25 - 20, 25 - 21, 25 - 19, 25 - 18\} = 7$

e o provedor 4 executaria a única tarefa. O pagamento do provedor 4 é calculado sem o vencedor da alocação, por meio de (7.4.2):

$$p_4 = 25 - \max_{1 \leq i \leq 3} \{25 - TD_{i1}\} = 25 - \max\{25 - 20, 25 - 21, 25 - 19\} = 25 - 6 = 19.$$

Portanto, o jogador 4 recebe R\$ 19.

Considere agora a versão completa de processo de negócios com várias tarefas não sequenciais. Além do mais, todas as restrições de tempo, reputação, confiabilidade e disponibilidade são consideradas no problema de programação de serviços web.

Os passos principais a serem seguidos são:

- Resolva o problema de otimização (7.4.1). Obtenha por meio do *Solver* a alocação ótima detalhada y_{ij}^* , determinando qual provedor i executa a tarefa j .
- Para obter a alocação y_{ij}^{-i} , sem o provedor vencedor i , atribui-se nas estruturas de dados do *Solver* custos muito grandes para o provedor i , por exemplo, R\$ 99999. Resolve-se o problema de otimização (7.4.2) para obter os pagamentos (segunda melhor alocação).

Como exemplo, considere um problema com duas tarefas sequenciais e dois provedores $\{P1, P2\}$. Na Tabela 7.2, apresenta-se um processo de negócios com duas tarefas T0 e TFinal sequenciais, onde os preços são em R\$ e os tempos de execução são em unidades de segundos. O orçamento máximo do comprador dos serviços é de R\$ 150,00. Note que o comprador quer maximizar a diferença entre seu orçamento máximo e a soma dos custos das tarefas.

preços	P1	P2	tempos	P1	P2
T0	R\$ 88,00	R\$ 87,00	T0	5,23	5,06
TFinal	R\$ 35,00	R\$ 44,00	TFinal	7,96	1,10

Tabela 7.2: Preços e tempos de execução do Exemplo 14.

Executando-se o *Solver* uma primeira vez, obtem-se a alocação de tarefas descrita na Tabela 7.3.

tarefas	provedor
T0	P2
TFinal	P1

Tabela 7.3: Plano de atribuição de tarefas do Exemplo 14.

Na Tabela 7.4, apresenta-se os preços modificados das tarefas específicas para auxiliar na determinação dos leilão de segundo preço.

preços	P1	P2
T0	R\$ 88,00	R\$ 99999,00
TFinal	R\$ 99999,00	R\$ 44,00

Tabela 7.4: Preços modificados do Exemplo 14.

Executando-se o *Solver* uma segunda vez, obtem-se as receitas referentes ao leilão de segundo preço, conforme apresentado na Tabela 7.5.

provedor	receita
P1	R\$ 88,00
P2	R\$ 44,00

Tabela 7.5: Pagamentos pelo segundo maior preço do Exemplo 14.

Exemplo. Exemplo das seis tarefas

Este exemplo é composto de seis tarefas não sequenciais com duas decisões. As tarefas são T1, T2, T3, T4, T5 e T6. São três caminhos de execução denominados ep_0 , ep_1 e ep_2 com frequências de execução 0,35, 0,35 e 0,30, respectivamente. Orçamento disponível ao comprador é de R\$ 20,00.

Os preços em R\$ e os tempos de execução em segundos são mostrados na Tabela 7.6.

preços	P1	P2	tempos	P1	P2
T1	3,50	4,40	T1	0,20	0,25
T2	2,50	1,90	T2	0,15	0,20
T3	0,50	1,00	T3	0,90	0,91
T4	2,10	1,20	T4	0,45	0,35
T5	4,00	3,20	T5	0,60	0,30
T6	1,50	1,00	T6	0,50	0,25

Tabela 7.6: Preços e tempos de execução do Exemplo 3.

Após a primeira execução do *Solver*, obteve-se o plano de atribuição de tarefas conforme Tabela 7.7. O processo de negócios pagaria aos provedores apenas R\$ 6,94 no caso de **leilão** de primeiro preço ou **menor preço**. A máxima diferença entre a receita v_0 e os gastos ponderados

por caminho de execução $freq_i EC_i$ é R\$ 20,00-R\$ 6,94= R\$ 13,06, ou seja, a sobra este valor para o dono do processo de negócios para este leilão.

tarefas	provedor		freq.	custo por ep_i	custo pond.
T1	P1				
T2	P2	ep_0	0.35	4,00	1,40
T3	P1	ep_1	0.35	7,60	2,66
T4	P2	ep_2	0.30	9,60	2,88
T5	P2	total			R\$ 6,94
T6	P2				

Tabela 7.7: Plano de atribuição de tarefas do Exemplo 3.

Antes de executar o *Solver* uma segunda vez, os preços correspondentes às alocações ótimas foram alteradas para R\$ 99999, na Tabela 7.8. Por exemplo, como o provedor P1 executa as tarefas T1 e T3 então os seus preços são alterados para R\$ 99999,00; o provedor P2 executa as tarefas T2, T4, T5 e T6 portanto os seus preços são alterados para R\$ 99999,00.

Tarefas	Provedor		Preços (R\$)	P1	P2
T1	P1		T1	99999,00	4,40
T2	P2		T2	2,50	99999,00
T3	P1	...	T3	99999,00	1,00
T4	P2		T4	2,10	99999,00
T5	P2		T5	4,00	99999,00
T6	P2		T6	1,50	99999,00

Tabela 7.8: Preços modificados do Exemplo 3.

O *Solver* produz um relatório de pagamentos do leilão de segundo menor preço, conforme mostrado na Tabela 7.9. Pelo leilão de segundo menor preço, o processo de negócios paga R\$ 9,29 aos provedores de serviços web. A sobra de receita do dono do processo de negócios é R\$ 20,00 - R\$ 9,29=R\$10,71. O teorema de Groves estabelece que neste caso os provedores revelariam seus preços reais.

	freq.	pagam. por ep_i	pagam. pond.		
ep_0	0.35	5,40	1,89	provedor	receita
ep_1	0.35	10,50	3,67	P1	R\$ 4,54
ep_2	0.30	12,40	3,72	P2	R\$ 4,75
total			R\$ 9,29		

Tabela 7.9: **Pagamentos pelo segundo menor preço** do Exemplo 3.

A abordagem do problema de programação de serviços web como um problema de projeto de mecanismos, proposto neste capítulo é simples, mas pode ser aprimorada com a consideração de resultados baseados em leilões combinatório (de tarefas).

Conclusões

Esta tese abordou o problema da programação de serviços web por meio de técnicas de Otimização Multiobjetivo e da Teoria de Jogos. O problema central da designação de provedores a tarefas foi formulado como um problema de programação linear-inteira multiobjetivo, e resolvido por uma técnica de programação alvo. Uma designação que minimiza os desvios dos objetivos de qualidade dos serviços web em relação a metas ideais mostrou-se viável e – o autor acredita – consistente com a forma como muitos serviços web compostos são projetados.

A tese também abordou a modelagem de situações nas quais os provedores disputam o direito de executar tarefas. A hipótese de que os provedores podem fixar os parâmetros de fornecimento de serviços de forma estratégica conduziu naturalmente o trabalho à Teoria de Jogos. A modelagem do problema de programação de serviços web pela Teoria de Jogos, combinada com uma formulação linear-inteira multiobjetivo detalhada do problema de designação de provedores a tarefas, é a principal contribuição da tese.

Uma noção clássica de equilíbrio em Teoria de Jogos, o Equilíbrio Correlacionado, foi usado com sucesso. O cálculo do equilíbrio é muito mais simples e eficiente quando comparado ao cálculo do Equilíbrio de Nash. Existe a premissa de que os provedores aceitam a intervenção de um agente que sinaliza a eles quais estratégias adotar. Esta abordagem pode ser considerada ilegal pelo sistema jurídico vigente. Contudo, abordagem proposta explicita a correlação e pode ser usada para prever situações de conclusão.

No que diz respeito à modelagem da interação entre provedores de serviços, duas linhas principais foram exploradas, ambas originárias da ideia de que cada provedor possui informações privadas – parâmetros como custos e tempos de execução das tarefas – não acessíveis aos demais provedores. Na primeira linha, informações privadas dão origem a um jogo de informação incompleta, também conhecido por jogo Bayesiano. A tese mostrou como modelar o problema de programação de serviços web, e por meio de experimentos computacionais, como obter os equilíbrios Bayes-Nash do jogo.

Em uma segunda linha de pesquisa, o problema de programação de serviços foi visto como um mecanismo de leilão a ser projetado. A sub-área da Teoria de Jogos conhecida como Projeto de Mecanismos fornece o suporte teórico necessário à modelagem do problema como um Mecanismo de Vickery-Groves-Clarke. Acreditamos que nesta segunda linha de pesquisa concentram-se as perspectivas mais promissoras para trabalhos futuros. Em particular, a natureza combinatória do problema de programação de serviços web tem motivado a adoção de várias alternativas de pagamentos aos provedores, e o chamado Leilão Ótimo vem sendo objeto de intensa pesquisa na área.

A compatibilização das metodologias propostas nesta tese com as exigências e as restrições operacionais de sua implementação em tempo real também é um tema a ser explorado futuramente.

Bibliografia

- Akbar, M. M., Rahman, M. S., Kaykobad, M., Manning, E. G. & Shoja, G. C. (2006). Solving the multidimensional multiple-choice knapsack problem by constructing convex hulls, *Computers and Operations Research* **33**: 1259–1273.
- Ardagna, D. & Pernici, B. (2005). Global and local qos guarantee in web service selection, *Proceedings of Business Process Management Workshop Performance Management* pp. 36–50.
- Ardagna, D. & Pernici, B. (2007). Adaptive service composition in flexible processes, *IEEE Transactions on Software Engineering* **33**(6): 369–384.
- Avis, D., Rosenberg, G., Savani, R. & von Stengel, B. (2010). Enumeration of nash equilibria for two-player games, *Game Theory* **42**: 9–37.
- Bell, M. (2008). *Service Oriented Modeling*, first edn, Wiley Publishing, New York.
- Brown, P. (2008). *Implementing SOA: Total Architecture in Practice*, first edn, Addison Wesley Professional, New York.
- Eiselt, H., Sandblom, C.-L., Spielberg, K., Richards, E., Smith, B., Laporte, G. & Boffey, B. (2000). *Integer Programming and Network Models*, first edn, Springer, New York.
- Fan, X. Q., Fang, X. W. & Jiang, C. J. (2011). Research on web service selection based on cooperative evolution, *Expert Systems with Applications* **38**: 9736–9743.
- Fontanini, W. & Ferreira, P. A. V. (2013). Multiobjective mixed-integer optimization scheduling of web services, *Anais do Simpósio Brasileiro de Automação Inteligente (SBAI 2013)*.
- Fontanini, W. & Ferreira, P. A. V. (2014). Game-theoretic approach for the web services scheduling problem, *Expert Systems with Applications* **to appear**.
- Gembicki, F. & Haimes, Y. (1975). Approach to performance and sensitivity multiobjective optimization: the goal attainment method, *IEEE Transactions on Automatic Control* **AC-20**(6): 769–771.
- Gnu (2008). Glpk (gnu linear programming kit), www.gnu.org/software/glpk/.
- Huang, A. F., Lan, C.-W. & Yang, S. J. (2009). An optimal qos-based web service selection scheme, *Information Sciences* **179**: 3309–3322.

-
- Khosravifar, B., Bentahar, J., Mizouni, R., Otrok, H., Alishahi, M. & Thiran, P. (2013). Agent-based game-theoretic model for collaborative web services: Decision making analysis, *Expert Systems with Applications to appear*.
- Narahari, Y., Narayanam, R., Garg, . & Prakash, H. (2009). *Game Theoretic Problems in Network Economics and Mechanism Design Solutions*, first edn, Springer Press, London, UK.
- Nash, J. (1951). Non-cooperative games., *Annals of Mathematics* **54**: 286–295.
- Nemhauser, G. L. & Wolsey, L. A. (1999). *Integer and Combinatorial Optimization*, John Wiley and Sons, New York, USA.
- OMG (2008). Business process management notation from object management group, www.bpmn.org.
- Osborne, M. J. (2003). *An Introduction to Game Theory*, Oxford Press, New York, USA.
- Papadimitriou, C. H. (2007). Complexity of finding nash equilibria, *Algorithmic Game Theory* .
- Papadimitriou, C. & Roughgarden, T. (2008). Computing correlated equilibria in multiplayer games, *Journal of the ACM* **55**(3).
- Papazoglou, M. (2008). *Web Services: Principles and Technology*, first edn, Pearson Education, New York.
- Shoham, Y. & Leyton-Brown, K. (2009). *Multiagent Systems. Algorithmic, Game-Theoretic and Logical Foundations*, Cambridge University Press, New York, USA.
- Sinnen, O. (2007). *Task Scheduling for Parallel Systems*, first edn, Wiley-Interscience, New York, USA.
- Steuer, R. (1989). *Multiple Criteria Optimization: Theory, Computation and Application*, Krieger Pub. Co., New York, USA.
- Wolsey, L. A. (1998). *Integer Programming*, first edn, Wiley-Interscience, New York.
- Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J. & Sheng, Q. (2003). Quality-driven web services composition, *Proceedings of International World Wide Web Conference (WWW)* pp. 411–421.
- Zeng, L., Benatallah, B., Ngu, A., Dumas, M., Kalagnanam, J. & Chang, H. (2004). Qos-aware middleware for web services composition, *IEEE Transactions on Software Engineering* **30**(5): 311–327.

Apêndice: Algoritmos

Algoritmo Não-Cooperativo

Algoritmo .1 Principal Nao-Cooperativo

```
/* input: n,p,d,tmin[n,p],tmax[n,p],pmin[n,p],pmax[n,p]
/* output: best_response[maxLines,n],
           M[maxLines,n],
           MU[maxLines,n]
/* local variables:
maxLines=(d+1)^(n*p);
M=Generate_Strategies(n,p,d,tmin,tmax);

write_to_file(M,"Otima2.strategies");

[file1,...,filen]=
    solve_optim_problems(n,p,maxLines,M,pmin,pmax,tmin,tmax);

best_response[maxLines,n],MU[maxLines,n]
    =calculate_Nash_equilibria(n,p,d);

/* print solution reports
PRINT M, MU, best_response;
```

Algoritmo .2 Generate_Strategies(n,p,d,tmin,tmax,)

```
/* input: n,p,d,tmin[n,p],tmax[n,p]
/* output: maxLines,M[maxLines,n*p]

for j=1 to n do
    for k=1 to p do
        { step_s[j,k]= (tmax[j,k]-tmin[j,k])/d;
          s[j,k] = tmin[j,k];
        }

maxLines=0;
Repeat
    maxLines++;
    for j=1 to n do
        for k=1 to p do M[maxLines,(j-1)*p+k] = s[j,k];

    // Test for end of processing
    done=true;
    for j=1 to n do
        for k=1 to p do
            done = done and s[j,k]==tmax[j,k];
    // Generate the next strategic profile
    if not done
        for j=1 to n do
            for k=1 to p do {
                s[j,k] = s[j,k] + step_s[j,k];
                if s[j,k] <= tmax[j,k] goto out;
                else s[j,k]=tmin[j,k];
            }
out:
until done

return maxLines,M;
```

Algoritmo .3 solve_optim_problems (n,p, maxLines,M,pmin,pmax,tmin,tmax)

```

/* input: n,p,maxLines,M[maxLines,n*p],pmin[n,p],pmax[n,p],
/*          tmin[n,p],tmax[n,p]
/* output: files with names "OTIMA2.maxLines.data" and
/*          "OTIMA2.maxLines.out"

for index=1 to maxLines do
{  file_name="OTIMA2."+index+".data"; open(file_name);
  Data = "data;";

  Data += "set WS_PROVIDERS :="; for j=1 to n do Data += "P"+j;
  Data += "set Tasks :="; for k=1 to p do Data += "T"+k;

  /* p-sequential tasks in path ep0 with only one subpath sp0
  Data += "set EP := ep0; set SUBP := sp0;";
  Data += "set subp[ep0,sp0]:="; for k=1 to p do Data += "T"+k;
  Data += "set ep[ep0]:="; for k=1 to p do Data += "T"+k;

  /* execution frequency of path 0
  Data += "param freq := ep0 1.0 ;";

  /* sequential mapping of tasks
  for k=1 to p-1 do Data += "set nextTasks [T"+k+"]:=T"+(k+1);
  Data += "set nextTasks [T"+p+"]:= ;";

  Data += "param WSP_Tasks_exectime default 9999999";
  for j=1 to n do Data += "P"+j;
  for k=1 to p do
    Data += "T"+j;
    for j=1 to n do Data += M_s[index][(j-1)*p+k]);

  Data += "param WSP_Tasks_price default 9999999";
  for j=1 to n do Data += "P"+j;
  for k=1 to p do
    Data += "T"+j;
    for j=1 to n do
      Data += linear_function(M[index][(j-1)*p+k],
                             pmin[j,k],pmax[j,k],tmin[j,k],tmax[j,k]);

  write_to_file(file_name,Data); close(file_name);

  exec("glpsol.exe --math --model OTIMA2.sol"+"--data file_name"
      +"--output file_name.out");
}

return file_name1.out,...,file_nameMaxLines.out;

```

Algoritmo .4 calculate_Nash_equilibria(n,p,d)

```
/* input: n,p,d, file otima2.strategies ,
/* output: best_response[maxLines,n],maxLines
/* local: current_strategies_indexes[n], b_r_i[max_strategies],
           MU[maxLines,n],strategies[n,max_strategies]
MU[maxLines,n]= read_utilities_from_file("otima2.out");
strategies[n,max_strategies]= read_strategies_from_file("otima2.strategies");
strategic_profiles[maxLines]= read_profiles_from_file("otima2.strategies");

for i_player=1 to n do {
  for ic=1 to n do current_strategies_indexes[ic]=1;
  done=false;
  while (not done) {
    max=0; max_profile=""; best_response_index=0;

    for ia=1 to max_strategies do {
      cs="";
      for ic=1 to n do
        if ic==i_player cs= cs + strategies[i_player,ia];
        else cs = cs + strategies[ic,current_strategies_indexes[ic]];

      if max_profile==" {
        max_profile=cs;
        for best_response_index=1 to maxLines do
          if strategic_profiles[best_response_index]==cs break;
        b_r_i[ia]=best_response_index;
        max=MU[best_response_index,i_player]; }
      else {
        for ii=1 to maxLines do
          if strategic_profiles[ii]==cs break;
        b_r_i[ia]=ii;
        if max < MU[ii,i_player] {
          best_response_index=ii; max = MU[ii,i_player];
          max_profile = cs; }
      }
    } /* end for ia
  /* The Main result is here
  best_response[best_response_index,i_player]=max;
  for ia=1 to max_strategies do
    if MU[b_r_i[ia],i_player]==max
      best_response[b_r_i[ia],i_player]=max;
  /* test for end of processing
  done=true;
  for iz=1 to n do if iz != i_player
    done = done AND curren_strategies_indexes[iz]=max_strategies;
  /* Generate the next strategic profile
  if not done
    for other_player=1 to n do if other_player != i_player {
      current_strategies_indexes[other_player]++;
```