

**Intervenção em Redes de Regulação Gênica Modelada
como um Processo de Decisão Markoviano Fatorado e Impreciso**

Fabio Alexandre Campos Tisovec

DISSERTAÇÃO APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO TÍTULO
DE
MESTRE EM CIÊNCIAS

Programa: Programa de Mestrado

Orientadora: Profa. Dra. Leliane Nunes de Barros

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro da CAPES

São Paulo, agosto de 2016

Intervenção em Redes de Regulação Gênica Modelado como um Processo de Decisão Markoviano Fatorado

Esta é a versão original da dissertação elaborada pelo
candidato (Fabio Alexandre Campos Tisovec), tal como
submetida à Comissão Julgadora.

Resumo

Tisovec, F. A. C. **Problema de Intervenção em Redes Gênicas Modelado como um Processo de Decisão Markoviano Fatorado e Impreciso**. Dissertação (Mestrado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2016.

Dentro da área de modelagem de sistemas biológicos, o estudo de redes gênicas inclui modelar interações, simular a dinâmica da rede e sugerir possíveis formas de intervir em seu funcionamento. Em particular, o problema de intervenção em redes gênicas é o de encontrar a melhor forma de intervir em sua evolução natural visando evitar cenários indesejáveis (doenças) ou atingir cenários desejáveis (estados sadios ou controláveis). Um modelo matemático comumente usado para representar redes gênicas é a Rede Booleana Probabilística (Probabilistic Boolean Network - PBN), em que os genes de uma dada rede são representados por variáveis booleanas e o sistema progride de um instante de tempo para o seguinte de acordo com funções booleanas, cada qual associada a uma probabilidade. Redes gênicas com ações de intervenção podem ser modeladas por um conjunto de redes booleanas probabilísticas com diferentes dinâmicas, uma para cada ação ou combinação de ações. Porém, a escolha da melhor ação para cada estado é feita resolvendo-se um processo de decisão markoviano com estados enumerados, que quando gerado a partir de uma rede booleana probabilística pode resultar em uma explosão combinatória de estados. Nesse trabalho mostraremos como o problema de intervenção em redes gênicas pode ser modelado como um processo de decisão markoviano fatorado, isto é, um processo de Markov descrito em termos de variáveis de estados que pode ser diretamente extraído de uma Rede Booleana Probabilística. Com tal modelo, podemos usar algoritmos simbólicos (que manipulam conjuntos de estados) capazes de resolver problemas com até centenas de variáveis de estado. Por fim, este trabalho propõe apresentar uma possível extensão deste modelo fatorado, onde admite-se probabilidades imprecisas de transição de estado dos genes, sinalizando um conhecimento impreciso destas transições.

Palavras-chave: redes de regulação gênica, redes gênicas, redes booleanas probabilísticas, processo de decisão markoviano fatorado, MDP fatorado.

Abstract

Tisovec, F. A. C. **Genetic Network Intervention Modeled as a Factored and Imprecise Markovian Decision Process**. Master's Tesis - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2016.

Within the area of modelling biological systems, the study of genetic networks includes modelling genetic interactions, simulate their dynamics and suggest possible means of intervention in their workings. Such study is very important to the public health because it helps prevent undesirable scenarios (diseases) and reach desirable scenarios (healthy states). One mathematical model commonly used to represent a genetic network is a *Probability Boolean Network (PBN)*, where the genes of the network are represented by boolean variables and the system progress from one instant to the next based on a set of boolean functions, each associated to a probability. Genetic networks with possible interventions can be modelled as sets of PBNs, one PBN for each action (intervention). The selection for the best action for each state can be defined by solving a markovian decision process with an enumerated space state, but the size of such space state can grow exponentially if defined based on a probabilistic boolean network.

This work has the objective of presenting a model for genetic network intervention based on probabilistic boolean networks and to show how this model can be solved using a finite horizon factored Markovian Decision Process, where the state of the network is represented by a set of state variables; further, this work intend to present a possible extension of the factored model, where the state transition probabilities are not perfectly known, thus modelled as ranges of probabilities.

Keywords: genetic networks, regulatory genetic networks, probabilistic boolean networks, factored Markovian decision process, factored MDP, MDP-IP.

Sumário

Lista de Figuras	ix
Lista de Tabelas	xi
1 Introdução	1
2 Redes de Regulação Gênica: Modelos de Representação e Intervenção	3
2.1 Redes Gênicas modeladas como Redes Booleanas	3
2.1.1 Redes Booleanas	4
2.1.2 Funções Booleanas Definidas por Ativação e Inibição	4
2.2 Redes Gênicas como Redes Booleanas Probabilísticas	6
2.2.1 Não-determinismo por Perturbações	6
2.2.2 Não-determinismo por Conjuntos de Funções Probabilísticas	7
2.2.3 PBNs e Cadeias de Markov	8
2.3 Métodos de extração de Redes Gênicas como PBNs	9
3 Processos de Decisão Markovianos	11
3.1 Processo de Decisão Markoviano Enumerativo - MDP enumerativo	11
3.2 Processo de Decisão Markoviano fatorado - MDP fatorado	12
3.2.1 Representação de MDPs fatorados por Diagramas de Decisão	13
3.2.2 Programação Dinâmica Simbólica: o algoritmo SPUDD	14
3.3 Processo de Decisão Markoviano com Probabilidades Imprecisas - MDP-IP	15
3.4 Processo de Decisão Markoviano com Probabilidades Fatorado - MDPIP fatorado	16
3.4.1 SPUDD-IP	16
4 Intervenção em Redes Gênicas como um Problema de MDP	17
4.1 Intervenção em Redes Gênicas como um MDP enumerativo	17
4.2 Intervenção em redes gênicas: horizonte, recompensa, custo e estados meta	18
4.3 Intervenção em redes gênicas como um MDP fatorado	20
4.4 Resolvendo o problema de intervenção com programação dinâmica simbólica	21
5 Intervenção em Redes Gênicas como um Problema de MDP-IP	23
5.1 PBNs com transições imprecisas	23
5.2 Especificação de um MDP-IP a partir de uma rede gênica com imprecisões	24

6	Análise Experimental	27
6.1	Escalabilidade do SPUDD-IP com relação ao Crescimento do Número de Transições com Probabilidades Imprecisas	28
6.2	Qualidade da Política encontrada por SPUDD-IP para Redes Gênicas Imprecisas . .	32
6.3	Escalabilidade do SPUDD-IP com relação ao Número de Intervenções	33
7	Trabalhos Correlatos	35
7.1	<i>Can Markov Chain Models Mimic Biological Regulation?</i> [KLD ⁺ 02]	35
7.2	<i>External Control in Markovian Genetic Regulatory Networks</i> [IS02]	35
8	Conclusões	37
	Referências Bibliográficas	39

Lista de Figuras

2.1	Exemplo de rede gênica para o ciclo celular da levedura, figura adaptada de [FLT04]	3
2.2	Representação gráfica de uma rede booleana: nós no grafo dirigido são variáveis booleanas e as arestas indicam suas dependências.	4
2.3	Conjunto de todos os possíveis estados da rede booleana da Figura 2.2, e as possíveis transições de estados.	4
2.4	Diferentes comportamentos de uma rede gênica, cada um sob influência de alguma intervenção. Um problema de intervenção em redes gênicas envolve a escolha da melhor PBN, segundo um dado critério de otimalidade. Note que cada PBN é definida por funções booleanas diferentes, porém elas possuem o mesmo conjunto de genes: para cada estado $\bar{x}(k)$, a transição para o próximo estado $\bar{x}(k + 1)$ depende do comportamento (PBN) em vigor.	9
3.1	Representação em árvore de decisão da função booleana	13
3.2	Redução do BDD da função booleana exemplo	13
6.1	Análise da influência da quantidade de transições imprecisas no tempo de execução do algoritmo SPUDD-IP. O eixo horizontal corresponde ao número de variáveis de estado (genes) de cada instância e o eixo vertical ao tempo de execução.	29
6.2	Influência de Quantidade Fixa de Transições Imprecisas no Tempo de Execução do Resolvedor: o eixo x corresponde ao número de variáveis de estado (genes) de cada instância e o eixo y ao tempo de execução.	31
6.3	Análise da qualidade da política encontrada por SPUDD-IP para redes gênicas imprecisas, com $1/4$, $1/8$, $1/16$, $1/32$, $1/64$ transições imprecisas e sem imprecisão. O eixo horizontal indica o número de variáveis de estado (genes) de cada instância e o eixo vertical, o valor da política encontrada no estado inicial.	32
6.4	Influência da quantidade de ações no te Source Rich Text mpo do resolvedor: o eixo x corresponde ao número de variáveis de estado (genes) de cada instância e o eixo y ao tempo de execução.	33

Lista de Tabelas

2.1	Matriz de interação da rede booleana ilustrada na Figura 2.2	5
2.2	Função booleana de transição de estados da rede booleana listrada na Figura 2.1.	6
2.3	Exemplo de uma Rede Booleana Probabilística (PBN).	7
3.1	Exemplo de função booleana	13
4.1	Exemplo de uma rede gênica com dois genes e uma intervenção [AD04].	18
4.2	Probabilidades não-nulas de transição do MDP gerado pelo exemplo na Tabela 4.1	18
4.3	Tabela de probabilidade condicional para a variável x'_2 , e a intervenção $u_1 = 0$ [AD04].	20
5.1	Exemplo de uma PBN com dois genes, x_1 e x_2 , uma ação u_1 e dois parâmetros de medida de probabilidade p_1 e p_2 [DCBD03].	24
5.2	Tabela de probabilidade condicional para a variável x'_2 e ação $v = u_1$ [AD04].	24
5.3	Tabela de probabilidades condicional para variável x'_2 e ação $v = u_1$ para o caso impreciso	25
6.1	Influência da Quantidade de Transições Imprecisas no Tempo de Execução do Resolvedor: tabela de valores. Entradas com "n/a" representam instâncias para as quais não foi possível concluir a execução.	29
6.2	Influência de Quantidade Fixa de Transições Imprecisas no Tempo de Execução do Resolvedor: tabela de valores.	30
6.3	Análise da qualidade da política encontrada por SPUDD-IP para redes gênicas imprecisas, com 1/4, 1/8, 1/16, 1/32, 1/64 transições imprecisas e sem imprecisão. Entradas com "n/a" representam instâncias para as quais não foi possível encontrar a política ótima em menos de 6 horas de execução do algoritmo SPUDD-IP.	33
6.4	Influência da quantidade de ações no tempo do resolvedor: tabela de valores.	34

Capítulo 1

Introdução

A área de genética moderna evoluiu consideravelmente nas últimas décadas. Dentre tais avanços, este trabalho faz uso de conhecimentos adquiridos em estudos de funcionamento e intervenção em *redes de regulação gênica* (*genetic regulatory networks - GRNs*), também chamadas por *redes gênicas*. Modelos de *Redes Gênicas* representam as interações entre genes com relação a uma determinada função biológica celular. Em sua definição canônica, redes gênicas referem-se à estrutura das relações entre os genes, isto é, como cada gene influencia os demais, e como tal estrutura determina a evolução natural do sistema biológico em questão. Exemplos de funções biológicas celulares são divisão celular [FLT04], diferenciação [BS01] e regulação de concentração molecular no meio celular [RM04, JJTN03]. Em uma GRN, à medida que certos genes passam a ser expressos e outros deixam de ser expressos, o comportamento celular é alterado de acordo com um padrão. Modelos de GRNs representam como os genes, envolvidos em uma dada função celular, interagem e como essas interações determinam padrões de evolução de uma célula.

Uma *rede booleana* (*boolean network - BN*) é um modelo matemático comumente usado para representar um subconjunto de GRNs em que a expressão gênica é simplificada a valores booleanos [Kau69, KLD⁺02]. Em uma BN fazemos a suposição de determinismo, i.e., é possível determinar precisamente qual o estado resultante de um sistema biológico após qualquer número de transições de estado. Dado que um grau de determinismo tão elevado não é realista para GRNs, foram criadas as *redes booleanas probabilísticas* (*probabilistic boolean networks - PBNs*) que assumem a existência de não-determinismo no comportamento do sistema e cuja dinâmica é baseada em uma função de probabilidade de transição [IS02]. Podemos representar diferentes comportamentos de uma GRN por meio de uma PBN definindo um conjunto de probabilidades de transição para cada um deles [DCBD03].

Um dos principais objetivos em modelar uma GRN é projetar e analisar estratégias de intervenções terapêuticas para conduzir o meio celular de um estado indesejável (doente) para um estado desejável (saudável). Intervenção em uma GRN é uma tentativa deliberada de alterar a dinâmica da rede de regulação gênica por meio do uso de drogas medicinais. Dentro de certas suposições, uma PBN pode ser representada por uma cadeia de Markov, o que permite usarmos o arcabouço de decisões de Markov para resolver o problema de intervenção. Assim, diferentes cadeias de Markov definem os diferentes comportamentos de uma GRN: cabe a um sistema de decisão selecionar os comportamentos desejáveis ao longo de um determinado período. A pesquisa na área de intervenção em GRNs modela este problema como um *Processo de Decisão Markoviano* (*Markov Decision Process - MDP*) extraído de uma PBN, que é um processo estocástico comumente usado em tomada de decisão sequencial. Uma solução de um MDP é uma política de ações (intervenções) que maximiza uma função utilidade.

MDPs podem ser classificados como enumerativos ou fatorados [CB99] dependendo da forma como os estados são representados. No primeiro caso, estados do modelo são enumerados, tendo como implicação que os algoritmos desenvolvidos para resolver problemas necessitam analisar todo o espaço de estados. MDPs fatorados, entretanto, representam cada estado como um conjunto de características relevantes para o mesmo, chamadas variáveis de estado, não necessitando enumerar

o espaço de estados inteiro, permitindo que os algoritmos manipulem apenas os estados necessários para determinar a solução para o problema, um conjunto potencialmente muito menor do que o espaço de estados inteiro.

Um MDP fatorado (FMDP) é uma alternativa promissora para representar o problema de intervenção, porque: (1) é capaz de resolver problemas maiores por explorar a estrutura de dependências entre as variáveis de estado, e (2) as redes já são fatoradas. Em FMDPs, a estrutura do modelo é explorada codificando as probabilidades de transição em modelos fatorados como, por exemplo, redes Bayesianas dinâmicas (DBNs) que podem ser resolvidas pelo algoritmo *Programação Dinâmica Simbólica (Symbolic Dynamic Programming - SDP)* [BHD99, TAP10]. Uma questão importante na resolução de um MDP é a chamada maldição da dimensionalidade, que se refere ao crescimento exponencial de tempo e memória requeridos pelo algoritmo devido ao aumento de dimensões nos dados. Neste caso, cada variável de estado (gene) é uma dimensão e, portanto, a cardinalidade do espaço de estados é 2^N para N genes, dado que estes são representados por valores booleanos.

Outra questão que pode surgir na modelagem de GRNs é a dificuldade de se obter probabilidades de transição precisas, devido a imprecisão e insuficiência de dados. Além disso, indivíduos diferentes podem possuir alelos diferentes de cada gene, o que resulta em probabilidades de transição diferentes entre indivíduos de uma população. Além disso, as probabilidades de transição podem ser não-estacionárias devido a informações parciais do modelo, dado que as redes booleanas são construídas por um conjunto selecionado de genes, ao invés de todos os genes do genoma do organismo. Para levar em conta essa incerteza, o modelo MDP com probabilidades imprecisas de transição (MDP-IP) pode ser usado [DSdB11, NEG04, SL70]. O MDP-IP é meramente uma extensão do MDP em que as probabilidades de transição podem ser especificadas imprecisamente. As invés de uma medida de probabilidade sobre o espaço de estados, temos um conjunto de medidas de probabilidades, chamado de *conjunto credal*.

Embora haja trabalhos sobre intervenção em GRNs abordando probabilidades imprecisas [DVCP09, PDD08, PDD09], a solução robusta trata apenas problemas de tamanhos pequenos. Neste trabalho, mostramos a eficiência da programação simbólica dinâmica (SPD) com o algoritmo de probabilidades imprecisas, SPUDD-IP [DSdB11], para resolução do problema de intervenção em uma GRN, modelada como um MDP-IP fatorado. SPUDD-IP é o método considerado estado-da-arte para solucionar MDP-IPs e portanto pode ser usado em instâncias maiores do problema de intervenção em GRNs.

Capítulo 2

Redes de Regulação Gênica: Modelos de Representação e Intervenção

Com o intuito de entender o padrão de comportamento de uma dada função celular, o estudo de Biologia Celular visou inicialmente entender a relação entre genes dois a dois, ou seja, como um determinado gene ativa ou inibe cada um dos demais, em uma dada função biológica. Embora tal abordagem tenha provido ao longo dos anos inúmeros resultados sobre interações gênicas, observou-se ser insuficiente para inferir o comportamento celular como um todo. Dessa forma, é necessário considerar as várias relações gênicas simultaneamente, surgindo assim o conceito de *Rede de Regulação Gênica*, ou simplesmente chamada por *Rede Gênica*.

2.1 Redes Gênicas modeladas como Redes Booleanas

é dado o nome de *Rede de Regulação Gênica* (ou *Rede Gênica*) a um conjunto de genes que interagem entre si de maneira indireta - pelo uso de seus RNAs transcritos e/ou suas proteínas expressas, juntamente com demais organelas na célula - de forma a regular a expressão dos compostos biológicos, isto é, os produtos de tais genes (proteínas e *RNA mensageiro* - *mRNA*). À medida que o tamanho de uma dada rede gênica aumenta, seu comportamento deixa de ser trivial, e sua modelagem e análise tornam-se cada vez mais complexas. A Figura 2.1 mostra uma representação gráfica da rede de regulação gênica do ciclo celular da levedura [FLT04], indicando seus genes e suas interações. A rede original possui cerca de 800, porém a figura mostra apenas os genes mais relevantes para o processo.

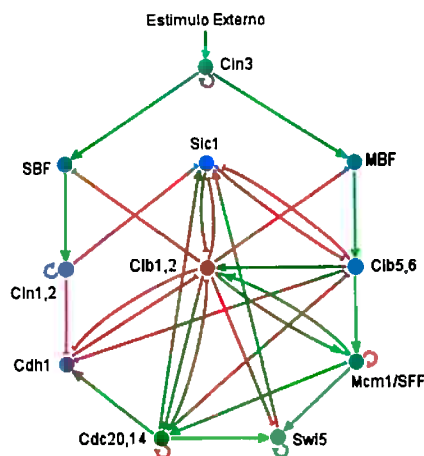


Figura 2.1: Exemplo de rede gênica para o ciclo celular da levedura, figura adaptada de [FLT04]

Foram desenvolvidos diversos modelos computacionais para representar redes gênicas. Dentre os mais difundidos, podemos citar o modelo diferencial [GK73] [JJTN03], redes de Petri [Rei85],

redes booleanas determinísticas [FLT04] e probabilísticas [YZT06] [IS02] e processos de decisão markovianos [TAP10]. Nesta seção, será descrito como uma rede gênica pode ser modelada como uma rede booleana (Seção 2.1.1) e uma rede booleana probabilística (Seção 2.2).

2.1.1 Redes Booleanas

é dado o nome de *rede booleana* [DD07] a um conjunto ordenado de variáveis booleanas x_1, x_2, \dots, x_n que interagem entre si segundo um conjunto de funções booleanas que determinam o valor de cada variável x_i num dado instante de tempo k . Redes gênicas podem ser modeladas como redes booleanas de tempo discreto. Esta representação é feita considerando que cada gene pertencente a uma rede gênica corresponde a uma variável booleana cujos valores podem ser: *ativo* ($x_i = 1$), se o gene está suficientemente expresso em um dado instante de tempo k e *inativo* ($x_i = 0$), caso contrário.

Seja $\vec{x}(k)$ o *perfil de expressão gênica* (*Gene Activity Profile - GAP*) o vetor binário que representa o estado da rede gênica no instante de tempo k , sendo $x_i(k)$ representa o valor do gene i no instante k . Definimos como $n = |\vec{x}|$ o número de genes da rede gênica. Um GAP geralmente é expresso por uma sequência binária, por exemplo, o GAP $\vec{x}(k) = (1, 1, 0, 1, 0)$ é expresso por 11010. Dada uma rede com n genes, concluímos que há no máximo 2^n possíveis GAPs (ou estados) distintos.

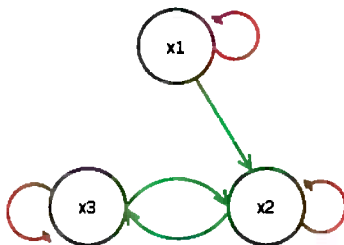


Figura 2.2: Representação gráfica de uma rede booleana: nós no grafo dirigido são variáveis booleanas e as arestas indicam suas dependências.

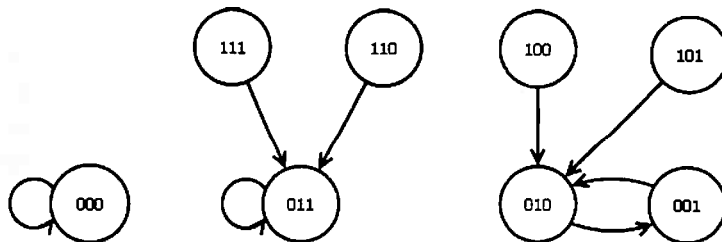


Figura 2.3: Conjunto de todos os possíveis estados da rede booleana da Figura 2.2, e as possíveis transições de estados.

A Figura 2.2 mostra um exemplo de rede booleana representada por um grafo em que nós indicam variáveis booleanas e arestas indicam as dependências entre estas (arestas coloridas indicam diferentes tipos de dependência, como explicaremos a seguir). A Figura 2.3 mostra o espaço de transição de estados da rede booleana representada na Figura 2.2, contendo 2^3 estados e as possíveis transições entre eles. Neste grafo, nós representam GAPs e arestas representa as possíveis transições de estado. Por exemplo, se no instante k a rede booleana está no estado 101, em $k + 1$, a rede estará no estado 010. Caso a rede esteja no estado 000, a rede permanecerá no mesmo estado no instante de tempo $k + 1$ e nos sucessores.

2.1.2 Funções Booleanas Definidas por Ativação e Inibição

Há diversas maneiras de se definir as funções booleanas que determinam o valor das variáveis de uma rede gênica no próximo instante de tempo, isto é, a função de transição de estados como

por exemplo, a ilustrada na Figura 2.3. Uma forma de defini-la foi proposta em [FLT04], em que é atribuído a cada par ordenado de genes, um valor representando o tipo de influência que o primeiro exerce sobre o segundo. Estes valores são representados por uma matriz de interação $A_{N \times N}$, em que $a_{ij} \in \{1, 0, -1\}$ recebe o valor:

- 1, caso o gene j ative o gene i ;
- -1, caso o gene j iniba o gene i , isto é, o gene i se tornará inativo no instante seguinte, caso não haja fatores ativando-o para contrapor tal inibição; e
- e zero, caso não exerça nenhuma influência.

Para calcular a expressão gênica no instante $k + 1$, isto é $\vec{x}(k + 1)$, a partir de $\vec{x}(k)$, definimos a seguinte função booleana:

$$x_i(k + 1) = \begin{cases} 1, & H_i(k) > 0 \\ 0, & H_i(k) < 0 \\ x_i(k), & H_i(k) = 0, \end{cases} \quad (2.1)$$

sendo

$$H_i(k) = \sum_j a_{ij} x_j(k). \quad (2.2)$$

Este modelo de atualização de estados de uma rede gênica pode ser usado de duas formas: (a) assíncrona, em que a cada instante de tempo, um gene é escolhido (aleatoriamente) e tem seu valor atualizado, ou (b) síncrona, em que todos os genes têm seus valores atualizados a cada instante de tempo.

é importante observar que uma rede gênica assim definida é determinística: dado o estado inicial do sistema, é possível saber com precisão em que estado a rede estará num instante de tempo futuro. Além disso, dado que o número de possíveis estados é finito (2^n), necessariamente em algum momento a rede irá passar por algum estado já visitado, e a partir deste ponto o sistema entra em um ciclo (que pode ser composto de um ou mais estados), dado o determinismo das transições.

O conjunto de estados que formam um ciclo é chamado de *atrator* e é possível haver mais de um atrator em uma dada rede booleana. *Bacia de atração* é o termo usado para descrever o conjunto de estados que eventualmente levam a um determinado atrator. Observe que todo estado do sistema pertence a alguma bacia de atração. O atrator no qual o sistema se encontra após um dado número de passos (suficientemente grande para se alcançar um atrator) depende de qual bacia de atração o estado inicial ($\vec{x}(0)$) pertence. Por exemplo, a rede booleana da Figura 2.2 contém 3 atratores $\{000\}$, $\{011\}$ e $\{010, 001\}$. Suas respectivas bacias de atração são $\{000\}$, $\{011, 111, 110\}$ e $\{010, 001, 100, 101\}$.

	x_1	x_2	x_3
x_1	-1	1	0
x_2	0	-1	1
x_3	0	1	-1

Tabela 2.1: Matriz de interação da rede booleana ilustrada na Figura 2.2

Exemplo de cálculo da função booleana. Considere a rede booleana com três variáveis da Figura 2.2 e sua matriz de interação conforme apresentada na Tabela 2.1. Note que essa matriz corresponde à mesma rede booleana representada graficamente na Figura 2.2, em que arestas verdes indicam ativação e arestas vermelhas indicam inibição. Usando a função booleana definida pelas Equações 2.1 e 2.2 e a matriz de interação da Tabela 2.1, podemos calcular a dinâmica de transição de estados, apresentada na tabela 2.2. Note que as colunas $\vec{x}(k)$ e $\vec{x}(k + 1)$ definem as mesmas transições ilustradas na Figura 2.3.

$\vec{x}(k)$	$H_1(k)$	$H_2(k)$	$H_3(k)$	$\vec{x}(k+1)$
000	0	0	0	000
001	0	1	-1	010
010	0	-1	1	001
011	0	0	0	011
100	-1	1	0	010
101	-1	2	-1	010
110	-1	0	1	011
111	-1	1	0	011

Tabela 2.2: Função booleana de transição de estados da rede booleana listrada na Figura 2.1.

2.2 Redes Gênicas como Redes Booleanas Probabilísticas

Apesar do modelo para redes gênicas definido na seção anterior modelar algumas instâncias de redes gênicas (todo: + referencia), por ser determinístico pode não ser adequado para modelar redes gênicas mais complexas. Observe que o fenômeno biológico real que queremos modelar é considerado robusto (todo: + referencia), isto é, mesmo na presença de incerteza nas transições de estado, o sistema transita tente a ir para um atrator. Em outras palavras, sabe-se que existe uma chance não nula do sistema transitar para um estado que não era o esperado, mas ao longo das várias transições de estado é esperado que a rede gênica se auto-regule e volte à sua operação usual.

Assim, para representar mais adequadamente o que realmente ocorre em uma rede gênica, é preciso adotar um modelo não-determinístico. Nas seções seguintes, descrevemos duas formas usadas na literatura para introduzir o não determinismo em redes booleanas: a primeira estende a Equação 2.1 adicionando uma perturbação probabilística no cálculo das transições (Seção 2.2.1) e a segunda considera múltiplas funções booleana e atribui um peso a cada uma (Seção 2.2.2).

2.2.1 Não-determinismo por Perturbações

Uma forma de se introduzir não-determinismo no modelo booleano é introduzindo perturbações durante a transição de estados [YZT06], ou seja, para cada gene existe uma pequena probabilidade de seu valor ser o inverso do esperado.

Desta forma, podemos redefinir a Equação 2.1 analisando as chances da transição ocorrer de acordo com o esperado, dado o valor de $H_i(k)$, a saber:

$H_i(k) > 0$: a probabilidade do gene x_i se tornar ativo é próximo de 1 e de se tornar inativo é próximo de 0;

$H_i(k) < 0$: a probabilidade do gene x_i se tornar ativo é próximo de 0 e de se tornar inativo é próximo de 1;

$H_i(k) = 0$: a probabilidade do gene x_i se manter no mesmo valor é próximo de 1 e de mudar de valor é próximo de 0;

Formalmente, dados os parâmetros α e $\beta \in \mathbb{R}$, podemos calcular essas probabilidades da seguinte forma:

- Se $H_i(k) > 0$ então

$$- Pr[x_i(k+1) = 1 | \vec{x}(k)] = \frac{\exp(\beta H_i(k))}{\exp(\beta H_i(k)) + \exp(-\beta H_i(k))} \approx 1$$

$$- Pr[x_i(k+1) = 0 | \vec{x}(k)] = \frac{\exp(-\beta H_i(k))}{\exp(\beta H_i(k)) + \exp(-\beta H_i(k))} \approx 0$$

- Se $H_i(k) < 0$ então

$$\begin{aligned}
- Pr[x_i(k+1) = 1|\vec{x}(k)] &= \frac{\exp(-\beta H_i(k))}{\exp(\beta H_i(k)) + \exp(-\beta H_i(k))} \approx 0 \\
- Pr[x_i(k+1) = 0|\vec{x}(k)] &= \frac{\exp(\beta H_i(k))}{\exp(\beta H_i(k)) + \exp(-\beta H_i(k))} \approx 1
\end{aligned}$$

- Se $H_i(k) = 0$ então

$$\begin{aligned}
- Pr[x_i(k+1) = x_i(k)|\vec{x}(k)] &= \frac{1}{1 + \exp(-\alpha)} \\
- Pr[x_i(k+1) \neq x_i(k)|\vec{x}(k)] &= 1 - \frac{1}{1 + \exp(-\alpha)}
\end{aligned}$$

Dada a função de probabilidades definida acima para cada gene estar expresso ou não no instante de tempo seguinte, podemos calcular as várias probabilidades do sistema encontrar-se em um determinado estado, dado o estado atual, conforme a equação:

$$Pr[\vec{x}(k+1) = \vec{s}|\vec{x}(k)] = \prod_{i=1}^n Pr[x_i(k+1) = s_i|\vec{x}(k)] \quad (2.3)$$

Na Equação 2.3 denotamos por \vec{y} o valor do GAP $\vec{x}(k+1)$ e y_i o valor de $x_i(k+1)$. Uma consequência interessante desta modelagem é que existe uma probabilidade positiva de qualquer estado ser atingido a partir de qualquer outro estado. Porém, para não descaracterizar a dinâmica de uma rede gênica, escolhe-se valores adequados para α e β que garantam valores adequados de probabilidades de transição [YZT06], isto é, que tais probabilidades tenham valores próximos aos do modelo determinístico apresentado na seção 2.1.1.

2.2.2 Não-determinismo por Conjuntos de Funções Probabilísticas

Outra forma para introduzir não-determinismo no modelo booleano, é modelar múltiplas funções de transição de estados para cada gene [ISO2]. Assim, ao invés de usarmos a Equação (2.1) para calcular $x_i(k+1) = f^{(i)}(\vec{x}(k))$, em que $f^{(i)}$ é uma função definida em $\{0,1\}^n \rightarrow \{0,1\}$, para cada gene i , introduzido O não-determinismo no modelo definindo para cada gene i um conjunto de $l(i)$ tuplas, cada qual composta por uma função $f_j^{(i)} : \{0,1\}^n \rightarrow \{0,1\} | j \in \{1, \dots, l(i)\}$ e sua probabilidade associada $c_j^{(i)}$, em que $\sum_{j=1}^{l(i)} c_j^{(i)} = 1$.

A cada transição de estado e para cada gene i , sorteia-se uma função $f_j^{(i)}$ de acordo com $c_j^{(i)}$ para determinar se tal gene estará expresso ou não no instante de tempo seguinte. Observe que isto é equivalente a sortear uma rede booleana determinística R dentre $N = \prod_{i=1}^n l(i)$ possíveis, cada qual com probabilidade $P(R) = \prod_{i=1}^n (c_j^{(i)} | f_R^{(i)} = f_j^{(i)})$, sendo $f_R^{(i)}$ a função booleana escolhida para o gene i na rede R .

x_1	x_2	$f_1^{(1)}$	$f_1^{(2)}$	$f_2^{(2)}$
0	0	0	0	0
0	1	1	0	0
1	0	1	0	0
1	1	0	1	0
$c_j^{(i)}$		1.0	0.5	0.5

Tabela 2.3: Exemplo de uma Rede Booleana Probabilística (PBN).

Redes gênicas modeladas desta forma são chamadas de **redes booleanas probabilísticas** (*Probabilistic Boolean Networks* - PBNs). A Tabela 2.3 mostra um exemplo de PBN com dois

genes x_1 e x_2 . O primeiro tem apenas uma função de transição de estados $f_1^{(1)}$, com probabilidade $c_1^{(1)} = 1$. O segundo gene tem duas funções de transição de estados $f_1^{(2)}$ e $f_2^{(2)}$, com probabilidade $c_1^{(2)} = c_2^{(2)} = 0,5$. A função de transição de estados probabilística de uma PBN, como mostrada na Tabela 2.3, é aplicada da seguinte forma. Por exemplo, dado que o estado atual é $\vec{x}(k) = (1, 1)$ (linha 4 da tabela), isto é $x_1 = 1$ e $x_2 = 1$, no estado $\vec{x}(k + 1)$, x_1 terá valor 0 com probabilidade 1 e x_2 terá valor 1 com probabilidade 0.5 ou valor 0 com probabilidade 0.5.

2.2.3 PBNs e Cadeias de Markov

Dada uma amostra contendo uma sequência de experimentos, por exemplo, observações de expressões gênicas (GAPs), todos os resultados podem influenciar na predição do próximo experimento. Em 1906, Andrei A. Markov começou a estudar um novo tipo de processo, onde o resultado de um dado experimento em um tempo k pode afetar o resultado do próximo experimento no tempo $k + 1$ [Basharin 04]. Este tipo de processo é chamado de *cadeia de Markov*. Uma cadeia de Markov pode ser definida da seguinte forma: seja $S = \{s_1, s_2, \dots, s_r\}$, um conjunto finito de estados. Considere um processo que começa em um desses estados e se move sucessivamente de um estado para outro, sendo cada movimento chamado de *estágio*. Se a cadeia está no estado $s_i \in S$ no instante de tempo k , então no instante de tempo $k + 1$, ela estará no estado $s_j \in S$ com probabilidade p_{ij} (chamada de *probabilidade de transição*). A matriz quadrada P contendo os elementos p_{ij} é chamada matriz de transição.

Numa cadeia de Markov, a probabilidade p_{ij} não depende dos estados anteriormente visitados [Grinstead 97]. Mais formalmente, dizemos que uma sequência de variáveis aleatórias X ,

$$X_0, X_1, \dots, X_k, X_{k+1}$$

é uma cadeia de Markov se a sequência for Markoviana no seguinte sentido, para todo k temos

$$Pr(X_{k+1} = s_i | X_0 = s_j, \dots, X_{k-1} = s_l, X_k = s_r) = Pr(X_{k+1} = s_i | X_k = s_r),$$

sendo $s_i, s_j, s_l, s_r \in S$.

Formalmente, uma *cadeia de Markov* é um modelo matemático com um estado inicial, um espaço de estados e probabilidades de transição de estado, dois a dois. Uma cadeia de Markov \mathcal{C} é definida pela tupla $\mathcal{C}\langle S, s_0, P \rangle$, sendo:

- $S = \{s_1, s_2, \dots, s_r\}$, o espaço de estados que o sistema pode assumir;
- s_0 , o estado inicial do sistema, com $s(0) \in S$; e
- $P(s(k + 1)|s(k))$, a probabilidade do sistema transitar para o estado $s(k + 1)$ dado que no instante k ele se encontra no estado $s(k)$. P respeita a seguinte propriedade: $\sum_{s' \in S} P(s(k + 1) = s' | s(k)) = 1, \forall s(k) \in S, k > 0$.

Observe que podemos montar uma cadeia de Markov a partir de uma rede booleana R , definindo o espaço de estados S como sendo o espaço de estados de R e definindo $P(s(k + 1) = s' | s(k)) = 1$ se $s' = \vec{x}(k + 1) | s(k) = \vec{x}(k + 1)$, ou $P(s(k + 1) = s' | s(k)) = 0$ caso contrário, ou seja, P terá valores binários. Também é possível montar uma cadeia de Markov a partir de uma PBN R , definindo o espaço de estados da mesma forma e $P(s(k + 1)|s(k)) = P(\vec{x}(k + 1)|\vec{x}(k))$.

Desta forma, podemos dizer que uma rede gênica modelada como uma rede booleana probabilística pode ser expressa por uma cadeia de Markov. Isso permite aplicar a vasta teoria de cadeias de Markov [Nor98] ao estudo de redes gênicas [BDS05, SDZ02].

Nesse trabalho, estamos interessados na modelagem e solução de problemas de intervenção em redes gênicas, isto é, queremos controlar a dinâmica de uma rede gênica o que implica em uma escolha entre uma ou mais PBNs (Figura 2.4) o que corresponde a fazer uma escolha entre uma ou mais cadeias de Markov.

Um dos principais usos de cadeias de Markov na área de redes regulatórias gênicas é a extração de redes gênicas a partir de amostras (cadeias de experimentos reais de observações de GAPs).

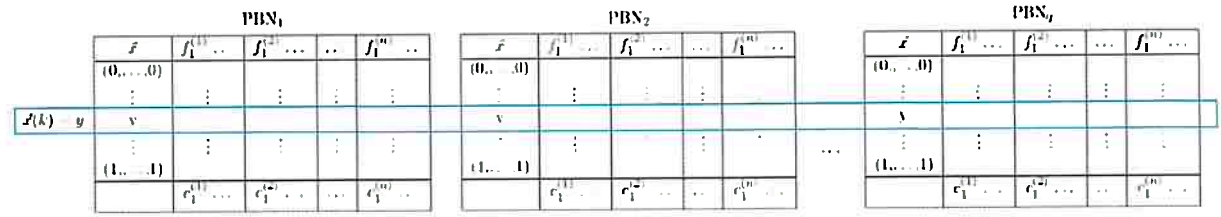


Figura 2.4: Diferentes comportamentos de uma rede gênica, cada um sob influência de alguma intervenção. Um problema de intervenção em redes gênicas envolve a escolha da melhor PBN, segundo um dado critério de otimalidade. Note que cada PBN é definida por funções booleanas diferentes, porém elas possuem o mesmo conjunto de genes: para cada estado $\vec{x}(k)$, a transição para o próximo estado $\vec{x}(k+1)$ depende do comportamento (PBN) em vigor.

2.3 Métodos de extração de Redes Gênicas como PBNs

O processo de definir uma modelagem matemática para uma dada rede gênica a partir de dados amostrais é abordado de diversas formas na literatura [HIG12]. De forma geral, isto envolve:

- observação da expressão gênica (em geral feita para conjuntos de genes, da ordem de dezenas de milhares por amostra, utilizando microarranjos,
- processamento numérico das amostras, como binarização ou discretização dos dados coletados, e
- inferência de um modelo matemático a partir dos dados processados.

Dentre as diversas formas de obtenção de observações, processamento e inferência de modelos, podemos citar [KLD⁺02, FES08, WGH06, Jon02, LFS98, BCM⁺07].

Seja um conjunto de amostras A , cada qual contendo dados amostrais da expressão gênica de um conjunto N de genes. Considere os parâmetros:

- n o tamanho desejado da PBN a ser construída (número de genes na PBN), onde $|N| \gg n$;
- p número máximo de parâmetros em cada função preditora; e
- q número desejado de funções predictoras para cada gene.

Podemos utilizar o algoritmo a seguir para construir uma PBN a partir de A e os parâmetros n , p e q :

- selecione um conjunto inicial de genes G (isso pode ser feito baseado em conhecimentos prévios da rede sendo estudada);
- para cada gene $g \in G$, encontra-se as q funções predictoras de até p parâmetros de genes em G que melhor predizem g nas amostras A ;
- enquanto $|G| < n$, faça:
 - escolha o gene i , tal que $i \in \{N \setminus G\}$ e i é o gene que apresenta a maior melhora das funções predictoras dos genes em $G \cup \{i\}$ nas amostras A ;
 - inclua i em G ;
 - Para cada gene $g \in G$, encontra-se as q funções predictoras de até p parâmetros de genes em G que melhor predizem g nas amostras A .

Para tratar problemas de intervenção em redes gênicas, usaremos cadeias de Markov controladas, também conhecidas como *Processos de Decisão de Markov*, ou *Processos de Decisão Markoviano*, conforme será descrito no próximo capítulo.

Capítulo 3

Processos de Decisão Markovianos

Nesse capítulo definimos processos de decisão markovianos e algumas de suas variantes, a saber: processos de decisão markovianos enumerativos (MDP enumerativo), em que os possíveis estados do sistema são enumerados; processos de decisão markovianos fatorados (MDP fatorado), em que os estados e funções são descritos em termos de variáveis de estado; e finalmente os processos de decisão markovianos fatorados e com probabilidades imprecisas (MDPIP fatorado).

3.1 Processo de Decisão Markoviano Enumerativo - MDP enumerativo

Um *Processo de Decisão Markoviano* (*Markovian Decision Process - MDP*) é um sistema estocástico capaz de modelar sistemas com transição probabilística de estados. Num MDP, a cada instante de tempo, um agente escolhe uma ação (com efeitos probabilísticos), que produzirá um estado futuro, uma recompensa e/ou um custo. O objetivo do agente é maximizar a recompensa esperada acumulada num horizonte de h escolhas de ação. A solução de um MDP é uma política π dada por uma função que associa a cada estado do sistema uma ação a ser executada.

Formalmente, um MDP é definido como uma tupla $\mathcal{M} = \langle Z, V, R, P \rangle$, em que:

- Z é o conjunto de estados possíveis do domínio.
- V é o conjunto de possíveis ações.
- $R : Z \times V \rightarrow \mathbb{R}$ é uma função que atribui uma recompensa a cada par (estado, ação).
- P determina probabilidades de transição de estado, em que $\sum_{z'} P(z'|z, v) = 1$, sendo $z, z' \in Z$;
- $E = (k_1, k_2, \dots, k_h)$ define um conjunto de h estágios de decisão.

Num MDP é importante conhecer o número de estágios de decisão que o processo deverá realizar, isto é, o horizonte do MDP, que pode ser de três tipos:

- *MDP de horizonte finito*: em que, o agente deve executar um número finito de ações, $h = k_{max}$. Neste caso, o agente muda o seu comportamento a medida que se aproxima de seus últimos estágios de decisão, o que implica em políticas não-estacionárias $\pi : Z \times E \rightarrow A$ (isto é, políticas que prescrevem uma ação para cada par estado e estágio);
- *MDP de horizonte infinito*: em que, o agente deve executar um número infinito de ações, $h = \infty$. Neste caso, o agente tende a manter o mesmo comportamento ao longo do tempo, o que implica em políticas estacionárias $\pi : Z \rightarrow A$ (isto é, uma política que prescreve uma ação para cada estado, independente do estágio);

- *MDP de horizonte indeterminado*: em que o agente deve parar ao alcançar algum estado identificado como estado meta, também chamado de MDP orientados a meta (*Goal driven MDP*) cuja subclasse mais conhecida é o MDP de *Caminho Mínimo Estocástico* (*Stochastic Shortest Path MDP - SSP MDP*).

Podemos representar a probabilidade do sistema atingir o estado z_j ao se executar uma ação $v \in V$ no estado z_i por uma matriz $M_{N \times N}$, sendo $N = |V|$, em que cada valor $M_{ij} = P(z_j|z_i, a)$. Tal matriz é chamada de Matriz de Transição do MDP, sendo que é dada uma matriz para cada ação do domínio. Esse modelo é chamado de MDP enumerativo uma vez que trabalha com estados enumerados, isto é, $Z = (z_1, z_2, \dots, z_N)$.

Solução de um MDP enumerativo. O valor da política no estágio $1 \leq k \leq k_{max}$ começando no estado z e executando π , denotado por $J_\pi(z, k)$, é definido como:

$$J_\pi(z, k) = E_\pi \left[\sum_{i=0}^{k_{max}-k} R_{i+k} | z_0 = z \right] \quad (3.1)$$

em que R_k é a recompensa obtida no estágio k .

Uma política é considerada ótima (π^*) quando maximiza a soma esperada da recompensa em um determinado estágio k do horizonte finito, ou seja, $J_{\pi^*}(z, k) \geq J_{\pi'}(z, k), \forall z \in Z, \forall \pi'$.

Seja J^* a função valor ótima determinada por qualquer política ótima. Em um problema de horizonte k_{max} , $J^*(z, k_{max}) = \max_{v \in V} \{R(z, v)\}$. Para os demais valores de k , J^* necessariamente precisa satisfazer a equação de Bellman [Bel57]:

$$J^*(z, k) = \max_{v \in V} \left\{ R(z, v) + \sum_{z' \in Z} P(z'|z, v) J^*(z', k+1) \right\}$$

Além disso, a política ótima $\pi^*(z, k)$ correspondente a função valor ótima é dada por:

$$\operatorname{argmax}_{v \in V} \left\{ R(z, v) + \sum_{z' \in Z} P(z'|z, v) J^*(z', k+1) \right\}$$

3.2 Processo de Decisão Markoviano fatorado - MDP fatorado

Num MDP fatorado, cada estado $z \in Z$ é descrito por um vetor de variáveis, isto é $z = \vec{x} = (x_1, x_2, \dots, x_n)$ e as funções de probabilidade de transição de estados P , para cada par estado-ação, são definidas por um grafo dirigido acíclico em duas camadas, o que define uma rede bayesiana dinâmica (*Dynamic Bayesian Network - DBN*), sendo que a primeira camada representa os valores das variáveis X no estado atual e a segunda representa os valores das variáveis X no estado seguinte (denotadas por X'). Além disso, as funções recompensa e custo também são definidas por variáveis de estado. Formalmente, um MDP fatorado é definido pela tupla $\mathcal{M} = \langle X, V, \mathcal{R}, \mathcal{P} \rangle$, em que X é um conjunto de variáveis de estados, P e R são funções também fatoradas. Sejam os vértices da DBN de um MDP fatorado, x_i e x'_i , e seja $pa(x'_i)$ os vértices pais de x'_i no grafo. Por construção, cada variável x'_i é condicionalmente dependente de seus pais, isso implica que a função de transição de probabilidade condicional P pode ser escrita como:

$$P(\vec{x}' | \vec{x}, v) = \prod_{i=1}^n \mathcal{P}(x'_i | pa(x'_i), v).$$

Com isto, pode-se reescrever a Equação de Bellman para um MDP fatorado da seguinte forma:

$$J^*(\vec{x}, k) = \max_{v \in V} \left\{ R(\vec{x}, v) + \sum_{\vec{x}' \in Z} \prod_{i=1}^n \{ \mathcal{P}(x'_i | pa(x'_i), v) \} J^*(\vec{x}', k+1) \right\} \quad (3.2)$$

3.2.1 Representação de MDPs fatorados por Diagramas de Decisão

A vantagem do modelo fatorado para MDPs é fazer uso de características estruturais do problema, com o intuito de reduzir o custo computacional para encontrar uma solução. Isto torna-se possível utilizando representações que permitem armazenar e manipular o problema tomando vantagem de suas características estruturais. Este é o motivo da função $\mathcal{P}(x'_i|pa(x'_i), v)$ ser definida por uma rede bayesiana e CPTs. Observe que nada impede de utilizar uma Matriz de Transição $|Z| \times |Z|$, porém isto não apenas causa um uso desnecessário de espaço de armazenamento, como faz com que os algoritmos efetuem cálculos desnecessários. Uma representação mais compacta para $\mathcal{P}(x'_i|pa(x'_i), v)$ é usar diagramas de decisão binários (*Binary Decision Diagrams - BDDs*) e sua extensão, os diagramas de decisão algébricos (*Algebraic Decision Diagrams - ADD*).

Diagramas de decisão podem ser interpretados diretamente como funções matemáticas para definir operações como soma, multiplicação, máximo e mínimo, e operações de conjuntos, como união, intersecção e complemento. Também é conveniente definir a operação unária de marginalização (em uma dada variável), conforme: $\sum_x F = F_{|x=1} \oplus F_{|x=0}$.

Considere a função booleana F dada pela Tabela 3.1. Observe que esta mesma informação pode ser representada em formato de uma árvore de decisão (Figura 3.1). Esta árvore apresenta várias sub-árvores repetidas e, portanto, não é a forma mais compacta de representar a função F . Diz-se que um diagrama de decisão é *canônico* quando todos os seus sub-DAGs são distintos, ou seja, não se pode encontrar sub-grafos idênticos.

x_1	x_2	x_3	$F(x_1, x_2, x_3)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Tabela 3.1: Exemplo de função booleana



Figura 3.1: Representação em árvore de decisão da função booleana

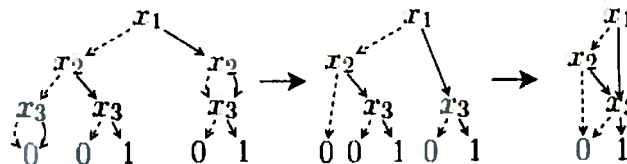


Figura 3.2: Redução do BDD da função booleana exemplo

É chamado de *redução* o processo de encontrar a forma canônica de um dado diagrama de decisão. Uma maneira de realizar este processo é verificar por sub-grafos idênticos a partir das folhas, alterando as referências de forma que seus pais apontem para apenas um dentre os sub-grafos idênticos. A qualquer momento deste processo, se ambas arestas de um dado vértice \mathcal{V} apontem para um mesmo vértice \mathcal{V}' , podemos concluir que \mathcal{V} é desnecessário para a decisão e pode

ser eliminado do BDD, fazendo o pai de \mathcal{V} apontar diretamente para \mathcal{V}' . A Figura 3.2 mostra as etapas de redução da árvore de decisão da Figura 3.1.

3.2.2 Programação Dinâmica Simbólica: o algoritmo SPUDD

Seja $\mathcal{M} = \langle X, V, \mathcal{R}, \mathcal{P} \rangle$ um MDP fatorado de horizonte finito. O algoritmo SPUDD (*Stochastic Planning Using Decision Diagrams*) [JH99] é a versão fatorada do algoritmo iteração de valor. Dada a Equação (3.2) e as funções \mathcal{R} , \mathcal{P} e J representadas por ADDs, denotadas por R_{DD} , P_{DD} e J_{DD} , respectivamente, SPUDD calcula a função valor ótima usando programação dinâmica simbólica:

$$J_{DD}^*(\vec{x}, k) = \max_{v \in V} \left\{ R_{DD}(\vec{x}, v) + \sum_{\vec{x}' \in Z} \bigotimes_{i=1}^n P_{DD}(x'_i | pa(x'_i), v) J_{DD}^*(\vec{x}', k+1) \right\} \quad (3.3)$$

Para realizar esse cálculo de maneira eficiente, SPUDD usa o algoritmo de eliminação de variáveis [ZP94] na marginalização sobre todos os estados sucessores, isto é, $\sum_{\vec{x}'}$. Por exemplo, se x'_1 não é dependente de qualquer outro x'_i para $i \neq 1$, podemos *empurrar* na Equação (3.3) a soma sobre x'_1 para dentro para obter:

$$\sum_{x'_i (i \neq 1)} \bigotimes_{i=1 (i \neq 1)}^n \{ P_{DD}(x'_i | pa(x'_i), v) \} \sum_{x'_1} P_{DD}(x'_1 | pa(x'_1), v) J_{DD}^*(\vec{x}', k+1)$$

O algoritmo SPUDD (Algoritmo 1) para horizonte finito, primeiro atribui à $J_{DD}^{k_{max}}$ o valor da recompensa no limite k_{max} (linha 1) e calcula J_{DD}^i dado J_{DD}^{i+1} (linhas 2 a 13), para i decrescente de $k_{max} - 1$ até 0. A função *SubstituiVariaveis* na linha 5 devolve um novo diagrama de decisão com estrutura idêntica a J_{DD}^{i+1} , em que todas as variáveis x_i são substituídas por x'_i . A eliminação de variáveis por marginalização é feita nas linhas 6 a 8. Na linha 9, $R_{DD|\vec{u}}^i$ denota o ADD obtido a partir de R_{DD}^i , onde as variáveis u_i tem seus valores definidos pelo vetor \vec{u} .

Algoritmo 1 SPUDD

Parâmetros: \vec{R}_{DD} , \vec{P}_{DD} , k_{max} , $|\vec{u}|$

```

1:  $J_{DD}^{k_{max}} \leftarrow R_{DD}^{k_{max}}$ 
2: for  $i = (k_{max} - 1) \rightarrow 0$  do
3:   for  $v = 1 \rightarrow 2^{|\vec{u}|}$  do
4:      $\vec{u} \leftarrow y^{-1}(v)$ 
5:      $val_{DD} \leftarrow \text{SubstituiVariaveis}(J_{DD}^{i+1}, \vec{x}, \vec{x}')$ 
6:     for  $l = 1 \rightarrow n$  do
7:        $val_{DD} \leftarrow val_{DD} \otimes P_{DD}(x'_l | \vec{x}, \vec{u})$ 
8:        $val_{DD} \leftarrow \sum_{x'_l} val_{DD}$ 
9:      $val_{DD} \leftarrow val_{DD} \oplus R_{DD|\vec{u}}^i$ 
10:    if  $v = 1$  then
11:       $J_{DD}^i \leftarrow val_{DD}$ 
12:    else
13:       $J_{DD}^i \leftarrow \max\{J_{DD}^i, val_{DD}\}$ 
14: devolva  $\vec{J}_{DD}$ 

```

3.3 Processo de Decisão Markoviano com Probabilidades Imprecisas - MDP-IP

Um MDP *com probabilidades imprecisas* (MDP-IP) generaliza um MDP permitindo que a função de transição probabilística seja especificada com imprecisão. Isto é, ao invés de uma medida de probabilidade $P(\cdot|z, v)$ sobre todos os estados Z , temos um *conjunto* de medidas de probabilidades $K(z'|z, v)$, chamado de *conjunto credal* [Coz00], sendo que $P(\cdot|z, v) \in K(z'|z, v)$.

Podemos assumir que todos os conjuntos credais são fechados e convexos, uma suposição frequentemente feita na literatura, e usada na maioria das aplicações práticas [Wal91]. Podemos ainda assumir estacionaridade das transições por conjuntos credais $K(z'|z, v)$; isto é, essas transições não dependem do instante de tempo t .

Formalmente, um MDP-IP é definido por $\mathcal{M}_{IP} = (Z, V, K, R)$. Essa definição é idêntica ao MDP, exceto que a função de transição de estados P é substituída pela função de transição de conjuntos credais K . Representaremos conjunto de distribuições de probabilidades de transição K implicitamente como o conjunto de probabilidades de transição consistentes com o conjunto de restrições de inequações lineares C sobre parâmetros de probabilidades.

Existem vários critérios de otimização que podem ser usados para definir o valor ótimo de uma política para um MDP-IP. No contexto de horizonte infinito descontado [DSdB11], sempre existe uma política ótima estacionária que é *maximin* ótima [SL70] (isto é, não existe outra política com valor maior sobre a suposição que a Natureza seleciona uma probabilidade de transição $P(z'|z, v)$ adversariamente ao valor minimal); além disso, dada a suposição que o conjunto de ações V é finito e o conjunto credal K é fechado, essa política induz uma função de valor ótimo J^* que é a solução de ponto-fixa única da Equação de Bellman para MDP-IPs:

$$J^*(z) = \max_{v \in V} \min_{P \in K} \left\{ R(z, v) + \gamma \sum_{z' \in Z} P(z'|z, v) J^*(z') \right\}, \quad (3.4)$$

sendo γ um fator de desconto entre o intervalo $]0, \dots, 1[$,

Existem vários algoritmos para resolver MDP-IPs enumerados baseados em programação dinâmica [SL70, WED94]. Uma solução clássica é o algoritmo de iteração de valor para MDP-IPs [SL70], dado pela atualização de Bellman:

$$J^t(z) = \max_{v \in V} \min_{P \in K} \left\{ R(z, v) + \gamma \sum_{z' \in Z} P(z'|z, v) J^{t-1}(z') \right\}, \quad (3.5)$$

que inicializa J^0 com algum valor heurístico e atualiza J^t usando J^{t-1} com a Equação 3.5 até convergência. Iteração de valor para MDP-IPs é similar à iteração de valor para MDPs, exceto que agora, para todo estado z , otimizamos a escolha de ação $v \in V$ com respeito à distribuição de *pior-caso* $P \in K$, aquele que minimiza o valor futuro esperado (supondo a Natureza como adversária). Assim nós garantimos que a função valor resultante e a política são robustas com relação à pior escolha que a Natureza possa fazer diante dos possíveis valores futuros $J^{t-1}(z')$ que o agente espera alcançar.

MDP-IPs enumerativos também sofrem da maldição de dimensionalidade de Bellman dos MDPs, isto é, o problema cresce exponencialmente com o número de variáveis usadas para representar os estados (para variáveis booleanas $|Z| = 2^{\text{no. de variáveis de estado}}$). Isso pode ser evitado com uma representação fatorada e solução simbólica, como é feito no algoritmo SPUDD-IP [DSdB11].

Neste trabalho, estamos interessados em resolver MDP-IPs fatorados de horizonte finito e encontrar a política ótima sem fator de desconto, como descrita a seguir.

3.4 Processo de Decisão Markoviano com Probabilidades Fatorado - MDPIP fatorado

Como no MDP fatorado, um MDP-IP fatorado representa cada estado por um vetor de variáveis $\vec{x} = (x_1, x_2, \dots, x_n)$ e as funções \mathcal{P} e \mathcal{R} de forma fatorada. A função de probabilidade de transição \mathcal{P} também é definida por uma rede bayesiana dinâmica, porém as CPTs são parametrizadas, ou seja, alguns valores de probabilidade são dados por parâmetros e um conjunto de restrições sobre eles.

3.4.1 SPUDD-IP

O algoritmo SPUDD-IP [DSdB11] é a versão do algoritmo SPUDD adaptado para resolver MDP-IPs fatorados. Para representar de forma compacta as CPTs de um MDP-IP, foi proposto uma extensão de ADDs chamados *ADDs parametrizados (parameterized ADDs - PADDs)* [DSdB11] dado que as folhas são expressões parametrizadas. PADDs não apenas permitem a representação compacta das CPTs para MDP-IPs fatorados, mas também permitem cálculos eficientes nas operações de iteração de valor.

Começamos por expressar a iteração de valor para MDP-IP de (3.5) na forma fatorada representando a função de transição por PADDs e as funções valor e recompensa por ADDs.

$$V_{DD}^t(\vec{x}) = \max_{a \in A} \left\{ R_{DD}(\vec{x}, a) \oplus \gamma \min_{\vec{p}} \left(\sum_{\vec{x}'} \bigotimes_{i=1}^n P_{DD}(x'_i | p_{a_i}(x'_i), a) V_{DD}^{t-1}(\vec{x}') \right) \right\} \quad (3.6)$$

Devido às CPTs no MDP-IP conter parâmetros \vec{p} , tais CPTs precisam ser representadas em formato de PADDs ($P_{DD}(x'_i | p_{a_i}(x'_i), a) \in K_a(x'_i | p_{a_i}(x'_i))$). A recompensa $R_{DD}(\vec{x}, a)$ pode ser representada por um ADD dado que contem apenas valores constantes (quanto às operações, ADDs são um caso particular de PADDs). Embora pareça que $V_{DD}^t(\vec{x})$ precise ser representado por um PADD, ressalta-se que os parâmetros \vec{p} são excluídos pela minimização $\min_{\vec{p}} \square$ em (3.6) ($\min_{\vec{p}} \square$ é a operação *MinParameterOut* em PADDs que faz a minimização sobre os parâmetros chamando um resolvidor não-linear para cada folha e retorna um ADD). Assim, o $V_{DD}^t(\vec{x})$ resultante do cálculo $\max_{a \in A}$ tem folhas com valores constantes e pode ser representado por um ADD.

Para explicar a avaliação eficiente de (3.6) com mais detalhes, exploramos o algoritmo de eliminação de variáveis [ZP94] na operação de marginalização $\sum_{\vec{x}'}$. Por exemplo, se x'_1 não é dependente de nenhum outro x'_i para $i \neq 1$, podemos “empurrar” a soma sobre x'_1 para dentro, obtendo:

$$V_{DD}^t(\vec{x}) = \max_{a \in A} \left\{ R_{DD}(\vec{x}, a) \oplus \gamma \min_{\vec{p}} \left(\sum_{x'_i (i \neq 1)} \bigotimes_{i=1 (i \neq 1)}^n P_{DD}(x'_i | p_{a_i}(x'_i), a) \sum_{x'_1} P_{DD}(x'_1 | p_{a_1}(x'_1), a) V_{DD}^{t-1}(\vec{x}') \right) \right\} \quad (3.7)$$

Então podemos continuar este processo com x'_2 , multiplicando este resultado por $P_{DD}(x'_2 | p_{a_2}(x'_2), a)$, somando sobre x'_2 , e repetindo para todo x'_i para calcular \square . Depois disso, \square não contem mais nenhuma variável x'_i , mas apenas variáveis x_i .

A solução de iteração de valor SPUDD-IP para MDP-IPs factorados devolve uma política ótima robusta e com frequência resulta em uma melhora sobre a iteração de valor clássica para MDP-IPs enumerativos. Em [DSdB11] é mostrado que para grandes espaços de estados (2^{12} to 2^{22}), Iteração de Valor não é capaz de encontrar uma solução por limitações de tempo e memória, enquanto que SPUDD-IP pode resolve-los.

Capítulo 4

Intervenção em Redes Gênicas como um Problema de MDP

Muitas redes gênicas são estáveis no sentido que, em geral, elas permanecem em conjuntos de estados estáveis, isto é, estados com pouca sensibilidade a perturbações. Isso corresponde a fazer transições para atratores de uma rede booleana. No trabalho de [SDZ02] a ideia de intervenção em redes gênicas está relacionada a encontrar um ou mais genes candidatos a intervenção (isto é, mudar sua expressão) de forma a aumentar a chance da rede entrar em um atrator desejado: dada uma PBN cujo comportamento seja transitar por um conjunto B de "estados indesejáveis", queremos que esta passe a transitar por um conjunto B' de "estados desejáveis". Assim, queremos ser capazes de intervir na expressão do menor número de genes possível de maneira a maximizar a chance da rede alcançar um atrator desejado. Essa abordagem permitirá identificar medicamentos para, por exemplo, o tratamento de câncer ou outras doenças relacionadas a funções genéticas.

4.1 Intervenção em Redes Gênicas como um MDP enumerativo

A intervenção em redes gênicas considera: um conjunto de genes X , um conjunto de PBNs relacionada a X como um conjunto de possíveis intervenções (ações) \vec{u} , na expressão gênica, num dado instante de tempo.

Em [AD04] é utilizado um Processo de Decisão Markoviano enumerativo (Seção 3.1) para resolver problemas de intervenção em redes booleanas. Nesta seção, apresentamos as principais ideias da descrição de um MDP enumerativo de horizonte finito baseado em recompensa de estados e custo nas ações, a partir de um conjunto dado de PBNs.

Para enumerar os estados é definida uma função bijetora $y : \mathbb{B}^n \rightarrow \mathbb{N}$ que transformar um dado vetor booleano em um número inteiro positivo, ou seja:

$$y(\vec{x}(k)) = 1 + \sum_{j=1}^n 2^{n-j} x_j(k) \quad (4.1)$$

Nesta forma, y é um valor inteiro positivo que define o estado do sistema no instante k . Além disso, é possível resgatar o valor de cada coordenada j do vetor $\vec{x}(k)$ a partir do valor y . Como decorrência desta transformação entre vetores booleanos e números inteiros, é possível observar que o resultado da função y não pode ser inferior a 1 e nem pode ser superior a 2^n , ou seja, seu domínio é $\{1, \dots, 2^n\}$.

Em [AD04] é considerado um horizonte finito com k_{max} estágios e um vetor R de recompensas para o último passo, $R_{k_{max}} = -z(\vec{x})$. Além disso, os custos de intervenção a cada passo k são modelados como recompensas negativas, dadas por $R_k = \sum_{u_j \in \vec{U}} \begin{cases} -1 & , u_j = 1 \\ 0 & , u_j = 0 \end{cases}$.

Podemos também aplicar a função y para enumerar o vetor \vec{u} de ações, isto é $v = y(\vec{u})$. Finalmente, o problema de intervenção pode ser resolvido usando o algoritmo de iteração de valor

[Put94].

A Tabela 4.1 mostra um exemplo de PBN com dois genes x_1 e x_2 e uma intervenção u_1 . Note que ao modelar a ação u_1 como um gene (isto é, como uma variável booleana) podemos interpretar que a ação ocorreu ($u_1 = 1$) ou que a ação não foi executada ($u_1 = 0$), porém, numa rede gênica, a não ocorrência de qualquer ação implica que a rede gênica se comportará segundo sua dinâmica natural, ou seja, livre de intervenções. Na Tabela 4.1, o gene x_1 tem apenas uma função de transição de estado e o gene x_2 tem duas. Observe que as três funções tem como parâmetros os dois genes, x_1 e x_2 , e a intervenção u_1 (interpretada como um gene de controle).

Dado o conjunto de PBNs, podemos calcular a função de probabilidade de transição do MDP enumerativo, computando as probabilidades conjuntas a partir das funções booleanas das PBNs, isto é, $f_j^{(i)}$.

A Tabela 4.2 mostra apenas as probabilidades conjuntas não-nulas de transição do estado z para o estado z' . Note que as colunas v , z e z' são calculadas respectivamente por $y(\vec{u})$, $y(\vec{x})$ e $y(\vec{x})'$ (Equação 4.1).

u_1	x_1	x_2	$f_1^{(1)}$	$f_1^{(2)}$	$f_2^{(2)}$	$R_{k_{max}}$
0	0	0	0	0	0	0
0	0	1	1	0	0	-1
0	1	0	1	0	0	-2
0	1	1	0	1	0	-3
1	0	0	1	0	0	
1	0	1	1	1	0	
1	1	0	0	1	0	
1	1	1	1	1	1	
$c_j^{(i)}$			1.0	0.5	0.5	

Tabela 4.1: Exemplo de uma rede gênica com dois genes e uma intervenção [AD04].

u_1	v	x_1x_2	z	$x'_1x'_2$	z'	$P(z' z, v)$
0	1	00	1	00	1	1,0
0	1	01	2	10	2	1,0
0	1	10	3	10	2	1,0
0	1	11	4	00	1	0,5
0	1	11	4	01	2	0,5
1	2	00	1	10	3	1,0
1	2	01	2	10	3	0,5
1	2	01	2	11	4	0,5
1	2	10	3	00	1	0,5
1	2	10	3	01	2	0,5
1	2	11	4	11	4	1,0

Tabela 4.2: Probabilidades não-nulas de transição do MDP gerado pelo exemplo na Tabela 4.1

4.2 Intervenção em redes gênicas: horizonte, recompensa, custo e estados meta

Estados meta podem, naturalmente, serem relacionados a estados sadios da rede gênica. Para isso, podemos listar os estados de atratores associados a fenótipos sadios como o conjunto de estados meta de um MDP orientado a meta ou um SSP MDP. Isso que fará com que a sequência ideal de intervenções seja realizada até que se um estado pertencente a um destes atratores seja alcançado.

Dependendo da modelagem, é possível incluir no conjunto de estados meta não apenas os atratores, mas também todos os estados pertencentes a suas bacias de atração, o que possibilita reduzir a profundidade da busca, uma vez que quando a rede encontra-se em uma destas bacias, há a garantia de que eventualmente chegará a um atrator sadio.

O horizonte de um MDP quando o problema é o de intervenção em redes gênicas, pode ser interpretado da seguinte forma:

1. para o tratamento de doenças comuns, dados já existentes de pacientes podem permitir uma estimativa de duração do tratamento. Portanto pode-se modelar o problema de intervenção em redes gênicas como um problema de cadeias de Markov controladas em que o número de passos de controle é previamente definido. A cada passo, decidimos se fazemos ou não uso de um medicamento, alterando a expressão dos genes influenciados por este. Fica implícito que ao final do número escolhido de passos a rede volta a operar inteiramente de acordo com sua evolução natural;
2. para tratamentos em que não se possui informação suficiente para definir *a priori* sua duração, é necessário utilizar uma modelagem em que o número de passos é indeterminado e definir um critério de parada. Usualmente isto é feito definindo um conjunto de estados meta. Se a qualquer momento o sistema encontra-se em algum estado da meta, entende-se que o tratamento atingiu seu objetivo; e
3. para tratamentos em que não é possível manter a rede em estados sadios consideramos a intervenção como um problema de cadeias de Markov controladas de horizonte infinito de passos. Podemos citar doenças crônicas como exemplo desta categoria, pois é sabido que sem demais intervenções a rede gênica eventualmente voltará a um estado doente.

Um mecanismo de diferenciação entre os estados mais relevantes, bem como os menos relevantes, para se manter um paciente é a atribuição de recompensas e/ou custos nos estados e ações. Em casos de doenças crônicas, sem haver diferenciação entre os estados desejáveis e indesejáveis, não é possível estabelecer uma estratégia (política) de intervenções que mantenha o paciente em estados sadios. Uma abordagem comum para isto é atribuir um valor a cada estado: positivo (recompensa) para os estados desejáveis e negativo (custo) para os indesejáveis. Dada esta modelagem, é possível fazer uma busca por estratégias de intervenção que buscam maximizar a recompensa obtida. Esta abordagem permite também estabelecer diferenciações entre o conjunto de estados sadios e entre o conjunto de estados doentes, bastando apenas aumentar ou diminuir a recompensa de cada estado individualmente caso haja informações suficientes para afirmar que alguns são mais desejáveis que outros.

Outra prática comum é atribuir um custo a cada medicamento, uma abordagem que não é auto-exclusiva com a anteriormente citada. Este custo por medicamento pode representar não apenas seu custo financeiro, mas também impactos adversos que costuma apresentar no paciente. O objetivo desta modelagem é incluir na busca por uma política de intervenções, o custo do medicamento, ou seja, em estados em que mais de um tratamento seja ideal, escolhe-se o que tem menos efeitos colaterais, o menor custo, ou uma combinação destes.

Naturalmente, a junção destas duas práticas leva a uma discussão bastante delicada que é a comparação da saúde e, em última instância, a vida do paciente em relação ao custo do medicamento. É importante observar que as recompensas definidas para cada estado e os custos associados aos medicamentos não precisam sequer estar nas mesmas ordens de grandeza. Atribuindo-se recompensas suficientemente grandes aos estados sadios em comparação aos custos, é possível conduzir a busca por uma estratégia ideal de intervenções que prese em primeira instância pela saúde do paciente, e apenas em casos em que dois tratamentos tenham eficácias semelhantes é que seja levado em conta o custo associado a cada um.

A modelagem do problema de intervenção em redes gênicas como um MDP enumerativo não é possível para redes com um número grande de genes devido à explosão combinatória: o número de estados cresce exponencialmente com o número de genes. Uma possível solução para isso é modelar

esse problema como um MDP fatorado e resolver o problema de intervenção com programação dinâmica simbólica (Seção 3.1.2).

4.3 Intervenção em redes gênicas como um MDP fatorado

Para modelar o problema de intervenção em redes gênicas como um MDP fatorado (Seção 3.2), dada uma rede gênica G por um conjunto de PBNs, as variáveis de estado correspondem aos genes de G , e as tabelas de probabilidade condicional (CPTs) são calculadas a partir das funções booleanas de G e suas respectivas probabilidades. Isso é feito através da seguinte equação:

$$P(x'_i = 1 | pa(x'_i), v) = \sum_{j=1}^{l_i} \begin{cases} c_j^{(i)} & , f_j^{(i)}(pa(x'_i), v) = 1 \\ 0 & , f_j^{(i)}(pa(x'_i), v) = 0 \end{cases} \quad (4.2)$$

sendo $P(x'_i = 0 | pa(x'_i), v)$ seu complemento, isto é, $1 - P(x'_i = 1 | pa(x'_i), v)$.

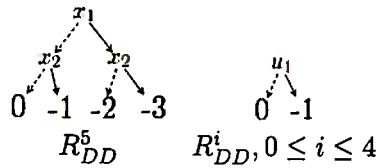
A Tabela 4.3 mostra a CPT para a variável x'_2 e a intervenção $u_1 = 0$ da Tabela 4.1. Essa tabela mostra todas as combinações de atribuições para os pais de x'_2 , isto é, $pa(x'_2)$. A probabilidade $P(x'_2 = 1 | \{x_1 = 1, x_2 = 1\}, u_1 = 0)$ é calculado usando a Equação 4.2, obtendo o valor 0.5 e $P(x'_2 = 0 | \{x_1 = 1, x_2 = 1\}, u_1 = 0)$ é o complemento.

x_1	x_2	$P(x'_2 = 1 \vec{x}, u_1 = 0)$	$P(x'_2 = 0 \vec{x}, u_1 = 0)$
0	0	0.0	1.0
0	1	0.0	1.0
1	0	0.0	1.0
1	1	0.5	0.5

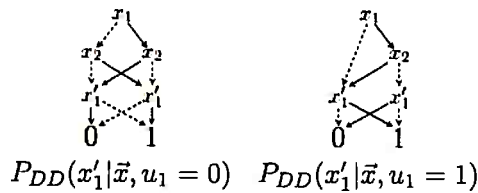
Tabela 4.3: Tabela de probabilidade condicional para a variável x'_2 , e a intervenção $u_1 = 0$ [AD04].

Observe que o modelo definido na Seção 4.1 pode ser facilmente mapeado para um MDP fatorado sem a necessidade de enumerar todos os estados, ou seja, $Z = \vec{x}$, ao invés de $Z = z(\vec{x})$. Tomando o exemplo dado na tabela 4.1, pode-se montar os diagramas de decisão para recompensa e para as probabilidades condicionais.

Podemos representar a função R de recompensas para o último passo, $R_{k_{max}} = -z(\vec{x})$ em que $k_{max} = 5$ e $R_k = \sum_{u_j \in \vec{U}} \begin{cases} -1 & , u_j = 1 \\ 0 & , u_j = 0 \end{cases}$ para $0 \leq k < k_{max}$ pelos ADDs R_{DD}^5 e R_{DD}^k a seguir:



A função de probabilidade de transição também pode ser representada por ADDs. Para o exemplo temos ADDs de probabilidade condicional para o gene x'_1 :



Em seguida fazemos a marginalização das variáveis x'_1 e x'_2 :

$$\begin{aligned}
 J_{DD}^4 &= \max \left\{ 0 \oplus \sum_{x'_2} \begin{array}{c} x_1 \\ \swarrow \quad \searrow \\ x'_2 \quad x'_2 \\ \downarrow \quad \downarrow \\ 0 \quad 1 \quad 0,5 \end{array} \otimes \begin{array}{c} x_1 \\ \swarrow \quad \searrow \\ x'_2 \quad x'_2 \\ \downarrow \quad \downarrow \\ -1 \quad 0 \quad -2 \quad -3 \end{array}, -1 \oplus \sum_{x'_2} \begin{array}{c} x_1 \\ \swarrow \quad \searrow \\ x'_2 \quad x'_2 \\ \downarrow \quad \downarrow \\ 0 \quad 0,5 \quad 1 \end{array} \otimes \begin{array}{c} x_1 \\ \swarrow \quad \searrow \\ x'_2 \quad x'_2 \\ \downarrow \quad \downarrow \\ -2 \quad -3 \quad 0 \quad -1 \end{array} \right\} \\
 J_{DD}^4 &= \max \left\{ 0 \oplus \sum_{x'_2} \begin{array}{c} x_1 \\ \swarrow \quad \searrow \\ x'_2 \quad x'_2 \\ \downarrow \quad \downarrow \\ -1 \quad 0 \quad -0,5 \end{array}, -1 \oplus \sum_{x'_2} \begin{array}{c} x_1 \\ \swarrow \quad \searrow \\ x'_2 \quad x'_2 \\ \downarrow \quad \downarrow \\ -1,5 \quad -1 \quad -2 \quad 0 \quad -0,5 \end{array} \right\} \\
 J_{DD}^4 &= \max \left\{ 0 \oplus \begin{array}{c} x_1 \\ \swarrow \quad \searrow \\ x'_2 \quad x'_2 \\ \downarrow \quad \downarrow \\ 0 \quad -2 \quad -0,5 \end{array}, -1 \oplus \begin{array}{c} x_1 \\ \swarrow \quad \searrow \\ x'_2 \quad x'_2 \\ \downarrow \quad \downarrow \\ -2,5 \quad -2 \quad -0,5 \quad -1 \end{array} \right\} \\
 J_{DD}^4 &= \begin{array}{c} x_1 \\ \swarrow \quad \searrow \\ x'_2 \quad x'_2 \\ \downarrow \quad \downarrow \\ 0 \quad -2 \quad -1,5 \quad -0,5 \end{array}
 \end{aligned}$$

As funções valor dos estágios 3, 2 e 1 são calculadas de maneira análoga a J_{DD}^4 :

$$J_{DD}^3 = \begin{array}{c} x_1 \\ \swarrow \quad \searrow \\ x'_2 \quad x'_2 \\ \downarrow \quad \downarrow \\ 0 \quad -1,5 \quad -0,5 \end{array}, J_{DD}^2 = \begin{array}{c} x_1 \\ \swarrow \quad \searrow \\ x'_2 \quad x'_2 \\ \downarrow \quad \downarrow \\ 0 \quad -1,5 \quad -0,75 \end{array} \text{ e } J_{DD}^1 = J_{DD}^2$$

Não é necessário calcular V_{DD}^0 para este exemplo, pois a função valor já convergiu no cálculo anterior, desta forma $J_{DD}^0 = J_{DD}^1$.

Capítulo 5

Intervenção em Redes Gênicas como um Problema de MDP-IP

Como foi dito anteriormente, uma das principais dificuldades na modelagem de uma GRN é a obtenção das probabilidades de transição, devido à imprecisão dos dados, bem como a quantidade insuficiente de dados. Além disso, indivíduos diferentes podem possuir alelos diferentes de cada gene, o que resulta em probabilidades de transição diferentes entre indivíduos de uma população. Uma outra fonte de imprecisão está relacionada ao fato das probabilidades de transição poderem ser não-estacionárias devido a informações parciais do modelo (modelo aproximado), dado que as redes booleanas são construídas por um conjunto selecionado de genes, ao invés de todos os genes do genoma do organismo. Para levar em conta essa incerteza, o modelo MDP com probabilidades imprecisas de transição (MDP-IP) (Seção 3.2).

Embora haja trabalhos sobre intervenção em GRNs abordando probabilidades imprecisas [DVCP09, PDD08, PDD09], a solução robusta trata apenas problemas de tamanhos pequenos. Neste trabalho, mostramos a eficiência da programação simbólica dinâmica (*SPD*) com o algoritmo de probabilidades imprecisas, SPUDD-IP [DSdB11], para resolução do problema de intervenção em uma GRN, modelada como um MDP-IP fatorado. SPUDD-IP é o método considerado estado-da-arte para solucionar MDP-IPs e portanto pode ser usado em instâncias maiores do problema de intervenção em GRNs.

5.1 PBNs com transições imprecisas

Dependendo da GRN sendo estudada, pode não ser viável coletar dados suficientes para definir com precisão a estrutura de sua PBN. Nestes casos, pode ser mais adequado fazer uso de um modelo mais geral, em que as probabilidades associadas às funções de transição de estados são representadas por parâmetros de medida de probabilidade, restritos por conjuntos de equações lineares.

Dado um conjunto de restrições C sobre os parâmetros p_1 e p_2 , por exemplo: $C = \begin{cases} p_1 + p_2 = 1.0, \\ p_1 > p_2, \\ 0.0 \leq p_2 \leq 0.3, \end{cases}$

a rede gênica dada como exemplo na Tabela 5.1 define um conjunto de PBNs, uma para cada par (p_1, p_2) de valores que satisfazem C .

Extração de PBNs com transições imprecisas a partir de amostras. Os métodos utilizados na biocomputação para extrair uma rede gênica a partir de dados reais, possuem intrinsecamente imprecisão na definição da transição de estados. Existem algumas formas de se estimar a imprecisão das probabilidades de transição calculadas da forma como foi descrito na Seção 2.3. Uma maneira é fazer um *K-fold* com as q funções escolhidas e encontrar $q - 1$ conjuntos de probabilidades para as funções predictoras; a partir disto pode-se determinar um conjunto credal de probabilidades para cada transição e resolver o problema com uso de um resolvidor probabilístico capaz de resolver problemas com tal especificação. Pode-se ainda, a partir dos $q - 1$ conjuntos de probabilidades,

u_1	x_1	x_2	$f_1^{(1)}$	$f_1^{(2)}$	$f_2^{(2)}$	$R_{k_{max}}$
0	0	0	0	0	0	0
0	0	1	1	0	0	-1
0	1	0	1	0	0	-2
0	1	1	0	1	0	-3
1	0	0	1	0	0	
1	0	1	1	1	0	
1	1	0	0	1	0	
1	1	1	1	1	1	
$c_j^{(i)}$			1.0	p_1	p_2	

Tabela 5.1: Exemplo de uma PBN com dois genes, x_1 e x_2 , uma ação u_1 e dois parâmetros de medida de probabilidade p_1 e p_2 [DCBD03].

calcular valores máximos e mínimos de probabilidade para cada transição, e posteriormente resolver o problema com um desenhado para trabalhar com problemas cuja imprecisão nas probabilidades de transição são descritas por valores máximos e mínimos. Outra opção é, a partir dos conjuntos credais das probabilidades de transição, definir funções de densidade do probabilidades que representem esta incerteza em um espaço contínuo entre zero e um. Para os testes realizados neste trabalho escolheu-se ter um controle maior de quanto de imprecisão nas probabilidades de transição há em cada instância testada, calculando-se um único valor de probabilidade, sem o uso do método *K-fold*, e posteriormente adicionando-se uma faixa de incerteza centralizada no valor calculado, sendo o tamanho desta faixa escolhido para cada teste.

5.2 Especificação de um MDP-IP a partir de uma rede gênica com imprecisões

A Tabela 5.2 define a função de probabilidade condicional de um MDP fatorado, também chamada *Tabela de Probabilidade Condicional (Conditional Probability Tables - CPTs)*.

x_1	x_2	$P(x'_2 = 1 \vec{x}, v = u_1)$	$P(x'_2 = 0 \vec{x}, v = u_1)$
0	0	0.0	1.0
0	1	0.0	1.0
1	0	0.0	1.0
1	1	0.5	0.5

Tabela 5.2: Tabela de probabilidade condicional para a variável x'_2 e ação $v = u_1$ [ADO4].

Podemos extrair as CPTs a partir das DBNs para cada par variável-ação, dada a seguinte equaç:

$$P(x'_i = 1 | pa(x'_i), v) = \sum_{j=1}^{l_i} \begin{cases} c_j^{(i)} & , f_j^{(i)}(pa(x'_i), v) = 1 \\ 0 & , f_j^{(i)}(pa(x'_i), v) = 0 \end{cases} \quad (5.1)$$

onde $P(x'_i = 0 | pa(x'_i), v)$ é seu complemento, ou seja, $P(x'_i = 0 | pa(x'_i), v) = 1 - P(x'_i = 1 | pa(x'_i), v)$.

A CPT para a variável x'_2 e ação $v = u_1$ é dada na tabela 5.2 do exemplo dado na tabela 4.1. Essa tabela mostra todas as combinações de valores dos pais de x'_2 , ou seja, $pa(x'_2)$. Por exemplo, a probabilidade $P(x'_2 = 1 | \{x_1 = 1, x_2 = 1\}, v = u_1)$ pode ser calculada usando a Equação 5.1, resultando no valor 0.5 e $P(x'_2 = 0 | \{x_1 = 1, x_2 = 1\}, v = u_1)$ é seu complemento.

A representação de MDPs fatorados [BHD99] também tem sido estendida para representar de maneira compacta MDP-IPs fatorados [DSdB11]. Isso requer modificar a representação de transição da DBN para considerar a incerteza quanto à exatas probabilidades de transição.

Conforme a definição anterior de um MDP-IP com estados enumerados, o conjunto de de todas as distribuições de transição viáveis para um MDP-IP fatorado é definido também por um *conjunto credal* K , mas agora fatorado sobre o vetor de variáveis \vec{x} . O desafio era especificar um conjunto credal de uma maneira fatorada e compacta. Para isso foi proposto o uso de uma *Rede Credal Dinâmica* (*Dynamic Credal Network - DCN*), um caso particular de rede [Coz00, Coz05], como uma forma apropriada de expressar conjuntos credais para transições fatoradas [DSdB11]. Assim, a equação 5.2 pode ser reescrita em forma fatorada:

$$J^*(\vec{x}) = \max_{v \in V} \min_{P \in K} \left\{ R(\vec{x}, v) + \gamma \sum_{\vec{x}' \in X} P(\vec{x}' | pa(\vec{x}), v) J^*(\vec{x}') \right\}. \quad (5.2)$$

x_1	x_2	$P(x'_2 = 1 \vec{x}, v = u_1)$	$P(x'_2 = 0 \vec{x}, v = u_1)$
0	0	0.0	1.0
0	1	0.0	1.0
1	0	0.0	1.0
1	1	p_1	p_2

Tabela 5.3: Tabela de probabilidades condicional para variável x'_2 e ação $v = u_1$ para o caso impreciso

Assim como [TAP10] demonstrou como construir um MDP a partir de uma PBN, é possível fazer uso de um método análogo para construir um MDP-IP a partir de uma PBN com transições imprecisas.

Capítulo 6

Análise Experimental

O objetivo da análise experimental realizada neste trabalho é explorar os limites e restrições do SPUDD-IP quando aplicado a MDP-IPs gerados a partir de redes gênicas modeladas por PBNS com imprecisão. Foram feitas as seguintes análises:

- Escalabilidade do algoritmo SPUDD-IP com relação ao crescimento do número de transições com probabilidades imprecisas.
- Análise da qualidade da política encontrada pelo algoritmo SPUDD-IP com relação ao crescimento do número de transições com probabilidades imprecisas.
- Escalabilidade do algoritmo SPUDD-IP com relação ao número de intervenções.

Para realizar os experimentos, foram geradas redes booleanas artificiais com base nos métodos utilizados na biocomputação a partir de dados reais. Para tanto, foi criado um gerador randômico de PBNS e MDP-IPs correspondentes, conforme descrito a seguir.

Gerador randômico de PBNS e MDP-IPs. Foi implementado o método de extração de redes gênicas como PBNS a partir de amostras de GAPS, com transições imprecisas, englobando as seguintes etapas:

1. geração aleatória de amostras;
2. extração de PBNS e
3. conversão de PBNS para MDPs/MDP-IPs.

Para a geração de amostras, o programa recebe cinco parâmetros:

1. número de amostras D ,
2. número máximo de genes $|G|$,
3. número de intervenções A ,
4. e valores mínimos e máximos de frequência que cada gene em D pode estar ativo, $minF$ e $maxF$.

O programa gera uma matriz de 3 dimensões $M \in \mathbb{B}_{D \times G \times A}$, onde cada posição representa o estado do gene g em uma dada amostra d e intervenção a . Além disso, para cada par (gene, intervenção), o gerador garante que a proporção de amostras em que tal gene está ativo está entre $minF$ e $maxF$, ou seja, $\forall g \in G, \forall a \in A : minF \leq ((\sum_{d \in D} m_{d,g,a})/D) \leq maxF$.

Para a extração de PBNS, o programa recebe como parâmetros:

1. um gene inicial g_0 ,

2. um valor p para o número máximo de genes preditores em cada função preditora,
3. q , o número máximo de funções booleanas preditoras para cada gene,
4. e uma lista L estritamente crescente, de valores inteiros positivos menores que $|G|$, onde para cada valor o algoritmo deve gerar uma instância de MDP com tal número de genes.

Começando com uma PBN R de tamanho $n = 1$ contendo apenas g_0 , a cada iteração o programa inclui um novo gene em R . Caso o tamanho da rede resultante $|R|$ esteja em L , é gerada uma instância de MDP baseada em R , e caso já tenha sido gerada uma instância para cada valor em L a execução é encerrada. Desta forma, o algoritmo gera uma sequência de $|L|$ redes, de tamanho crescente em número de genes e estruturalmente relacionadas entre si.

A escolha do gene a ser incluído a cada iteração é feita da seguinte forma: para cada gene g_i ausente à rede R , faz-se uma iteração entre todas as intervenções e entre todos os genes presentes à R escolhendo-se a função preditora mais adequada. é escolhido o g_i que apresenta a maior melhora no conjunto de funções preditoras escolhidas.

Dada uma função booleana $f(g_{i_1}, g_{i_2}, \dots, g_{i_l}) = g_i : \mathbb{B}^{l_i} \rightarrow \mathbb{B}$, preditora do gene g_i , é possível calcular o *Coefficiente de Determinância* de f em M por:

$$CoD(f, M) = \left(\sum_{a \in A, d \in D} \begin{cases} 1, & f(m_{i_1, a, d}, m_{i_2, a, d}, \dots, m_{i_l, a, d}) = m_{i, a, d} \\ 0, & f(m_{i_1, a, d}, m_{i_2, a, d}, \dots, m_{i_l, a, d}) \neq m_{i, a, d} \end{cases} \right) / CoD_{g_i} \quad (6.1)$$

$$CoD_{g_i} = \max \left\{ \sum_{a \in A, d \in D} m_{i, a, d}; A * D - \sum_{a \in A, d \in D} m_{i, a, d} \right\} / A * D \quad (6.2)$$

Seja $GEN(R_{it}) = \{g_{R_{it_1}}, g_{R_{it_2}}, \dots, g_{R_{it_{|R_{it}|}}}\}$ o conjunto de genes em R_{it} . Queremos encontrar o gene $g_j \in G \setminus GEN(R_{it})$ que oferece a maior melhora para as funções preditoras para a rede $R'_{it} = R_{it} \cup \{g_j\}$. Para isso, iteramos entre todos os genes $g_j \in G \setminus GEN(R_{it})$ e selecionamos q funções preditoras de g_j em $GEN(R_{it})$ com o maior CoD, iterando entre todas as funções possíveis com até p genes preditores. O gene g_j com a maior média de CoD entre suas q funções escolhidas é incluso em R_{it} , gerando R'_{it} .

Para atribuir probabilidades a cada uma das q funções preditoras escolhidas, basta normalizar seus respectivos coeficientes de determinância.

6.1 Escalabilidade do SPUDD-IP com relação ao Crescimento do Número de Transições com Probabilidades Imprecisas

Nesse experimento, queremos analisar como a quantidade de transições imprecisas influencia no tempo de execução do SPUDD-IP, isto é, no tempo para calcular a política ótima de cada instância analisada.

Geração de instâncias com transições imprecisas . A partir de um conjunto artificial de amostras foram extraídas 7 instâncias, de 5 a 11 genes, cada qual com 4 ações possíveis. A partir deste conjunto de instâncias, foram derivados outros 5 conjuntos, cada qual com 7 instâncias. No primeiro conjunto de 7 instâncias, foram introduzidas imprecisões em 1/64 do total de transições, no segundo foram 1/32 (incluindo as imprecisões anteriores), no terceiro foram 1/16, no quarto foram 1/8 e no quinto foram 1/4. As transições imprecisas contêm uma faixa de incerteza de 12,5%, centrada no valor de probabilidade precisas inicialmente gerada. Todas as instâncias foram resolvidas para um horizonte de tamanho cinco. Os resultados podem ser observados na Figura 6.1 e na tabela 6.1, ambas contendo a mesma informação.

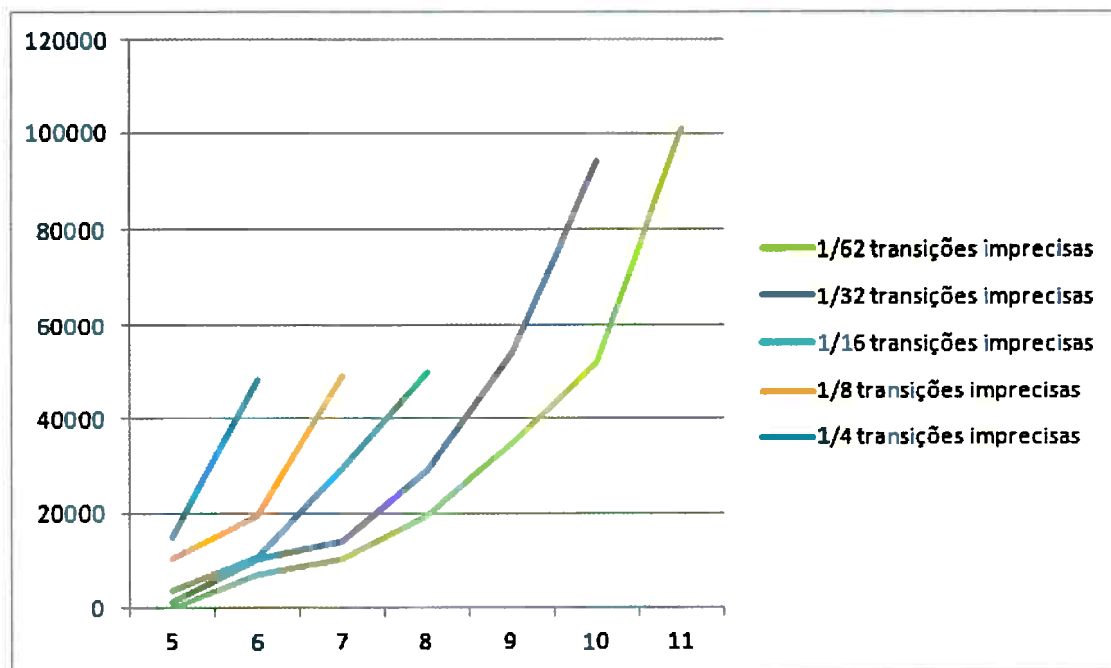


Figura 6.1: Análise da influência da quantidade de transições imprecisas no tempo de execução do algoritmo SPUDD-IP. O eixo horizontal corresponde ao número de variáveis de estado (genes) de cada instância e o eixo vertical ao tempo de execução.

variáveis	transições imprecisas				
	1/62	1/32	1/16	1/8	1/4
5	78	1281	3934	10212	14777
6	7038	10232	10994	19587	48413
7	10315	14184	29379	49218	n/a
8	19361	28859	50278	n/a	n/a
9	34730	54096	n/a	n/a	n/a
10	52182	94158	n/a	n/a	n/a
11	101278	n/a	n/a	n/a	n/a

Tabela 6.1: Influência da Quantidade de Transições Imprecisas no Tempo de Execução do Resolvedor: tabela de valores. Entradas com "n/a" representam instâncias para as quais não foi possível concluir a execução.

Os resultados da Figura 6.1 e Tabela 6.1 sugerem que com o aumento da fração de transições imprecisas, o tempo que o algoritmo SPUDD-IP consome para encontrar políticas ótimas cresce exponencialmente: instâncias com 1/4, 1/8 e 1/16 transições imprecisas resolvem instâncias até 6, 7 e 8 variáveis, respectivamente; instâncias com 1/32 e 1/64 transições imprecisas resolvem instâncias até 10 e 11 variáveis, respectivamente. Note que para instâncias com o mesmo número de variáveis, por exemplo as instâncias com 8 variáveis, o aumento de transições imprecisas implica num crescimento exponencial no tempo gasto para encontrar a políticas ótima. Na Tabela 6.1, experimentos marcados com n/a não encontraram uma solução em 6 horas de execução do algoritmo.

variáveis	transições imprecisas					
	1	2	4	8	16	32
5	77	4913	7526	13609	16149	18741
6	66	4908	7590	13979	25737	35802
7	70	4885	7577	12894	15268	26283
8	63	1541	3063	6593	6608	9068
9	63	1548	3032	7725	7815	18906
10	62	1546	3045	7836	7887	15392
11	63	1548	3029	6492	6703	13919
12	63	1529	2999	6465	6416	6591
13	63	1530	2984	6607	6452	6556
14	64	1546	3070	6306	15210	31082
15	67	1537	4196	9162	9316	23838
16	67	1534	4100	8694	8957	23489
17	65	1536	4110	9102	9051	23567
18	66	1538	4089	8881	9059	24215
19	72	1533	4085	8924	8838	23934
20	66	1529	4093	8854	8792	23910
21	68	1537	4089	8903	9029	24510
22	71	1553	4095	9052	9140	23859
23	69	1525	4116	9002	8837	23793
24	71	1541	4120	9044	9023	24000
25	69	1546	4112	9000	9042	23594
26	71	1545	4130	9187	8978	23495
27	71	1536	4112	8971	9150	24017
28	72	1539	4112	9147	9233	24076
29	71	1539	4130	8853	8967	23467
30	73	1549	4145	9140	8830	9255
31	72	1537	4126	8861	8844	8874
32	69	1532	4103	8841	9051	8917

Tabela 6.2: *Influência de Quantidade Fixa de Transições Imprecisas no Tempo de Execução do Resolvedor: tabela de valores.*

Outra forma de avaliar o impacto no desempenho do resolvedor pela quantidade de imprecisão manipulada é, ao invés de se manter constante a porcentagem de transições com imprecisão, manter constante o número absoluto de transições imprecisas à medida que se aumenta o tamanho das instâncias. Para tanto, a partir de um conjunto artificial de amostras foram extraídas vinte e oito instâncias, de cinco a trinta e dois genes, cada qual com quatro ações possíveis. A partir deste conjunto de vinte e oito instâncias foram gerados outros cinco conjuntos, cada qual também com vinte e oito instâncias, o primeiro tendo uma única transição com probabilidade imprecisa por instância, o segundo com duas, o terceiro com quatro, o quarto com oito e o quinto com dezesseis. A cada uma destas transições foi adicionada uma faixa de incerteza de 12,5%, centrada no valor de probabilidade inicialmente gerado. Todas as instâncias foram resolvidas para um horizonte de

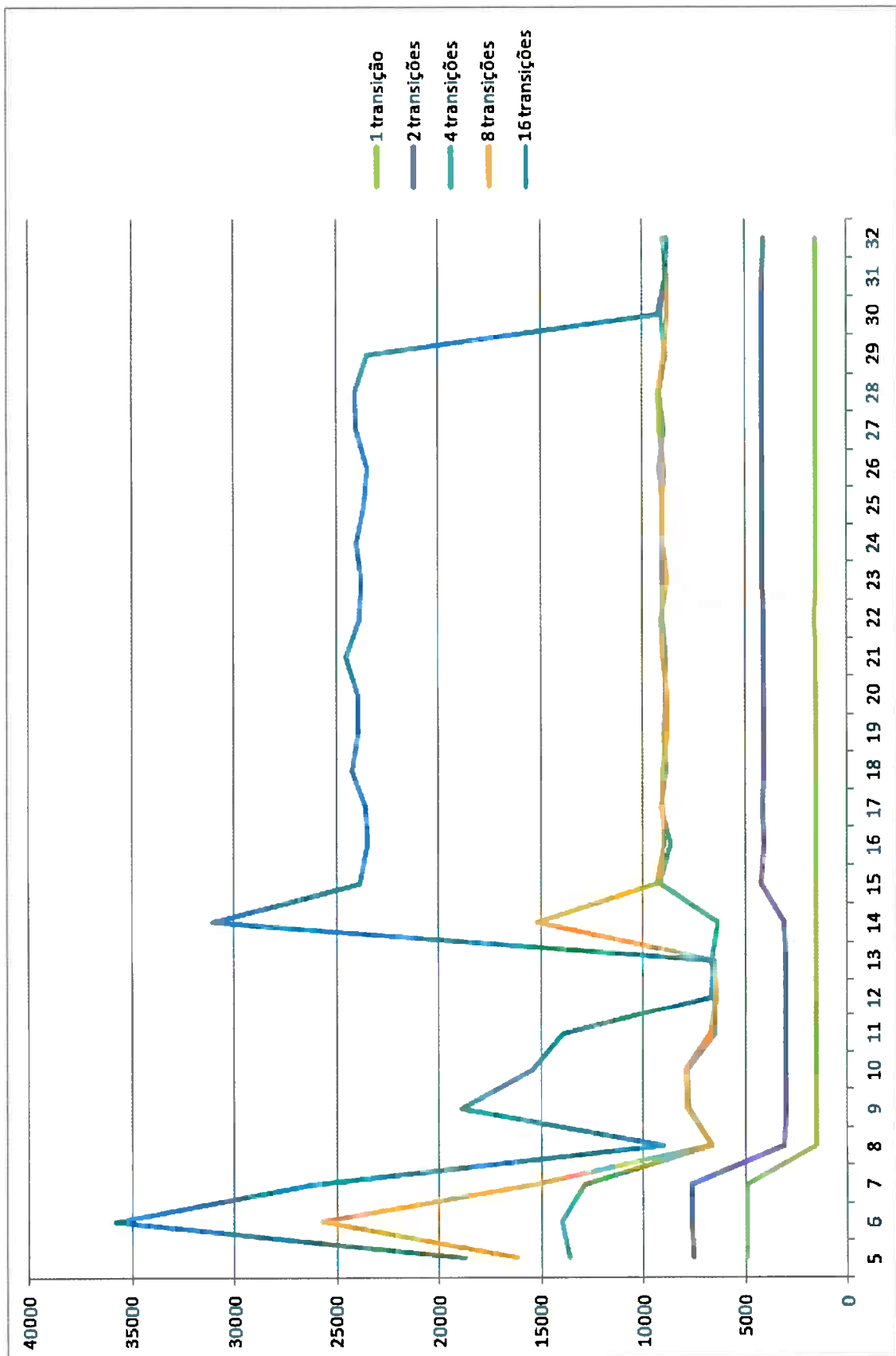


Figura 6.2: Influência de Quantidade Fixa de Transições Imprecisas no Tempo de Execução do Resolvedor: o eixo x corresponde ao número de variáveis de estado (genes) de cada instância e o eixo y ao tempo de execução.

tamanho cinco. Os resultados podem ser observados na figura 6.2 e na tabela 6.2, ambas contendo a mesma informação.

6.2 Qualidade da Política encontrada por SPUDD-IP para Redes Gênicas Imprecisas

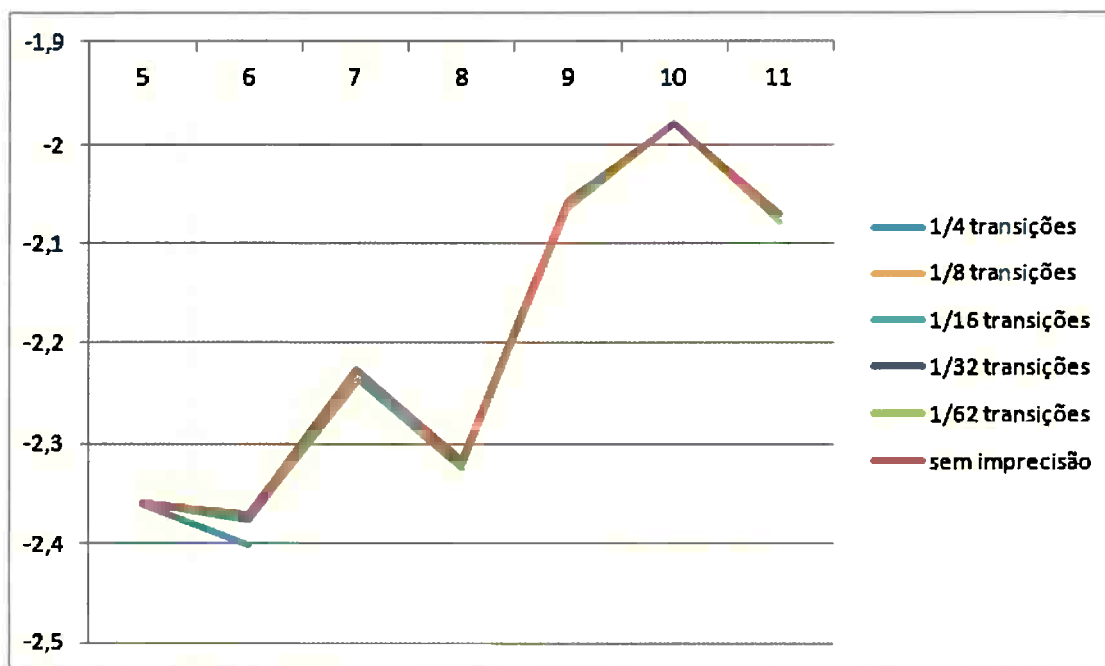


Figura 6.3: Análise da qualidade da política encontrada por SPUDD-IP para redes gênicas imprecisas, com 1/4, 1/8, 1/16, 1/32, 1/64 transições imprecisas e sem imprecisão. O eixo horizontal indica o número de variáveis de estado (genes) de cada instância e o eixo vertical, o valor da política encontrada no estado inicial.

Nessa seção, queremos analisar como o aumento de imprecisão na Rede Gênica afeta a qualidade da política encontrada usando o algoritmo SPUDD-IP. Uma motivação para isso é usar essa análise para avaliar a qualidade da rede gênica imprecisa em função da qualidade da política encontrada. As instâncias com transição imprecisa foram geradas da mesma forma como descrito na seção anterior.

O gráfico da Figura 6.3 e a Tabela 6.3 mostram o valor das políticas encontradas pelo SPUDD-IP para diferentes instâncias de redes gênicas imprecisas. Para cada instância foram geradas redes gênicas com diferentes frações de transições imprecisas: 1/4, 1/8, 1/16, 1/32 de 1/64 de transições imprecisas do total de transições de cada instância. Também foi resolvido instâncias sem imprecisão. Os resultados da Figura 6.3 mostram que, o SPUDD-IP encontrou políticas ótimas para todas as instâncias sem transições imprecisas (curva vermelha) e para todas as instâncias com 1/64 de transições imprecisas (curva verde), com valores muito próximos, o que indica que essa fração de transições imprecisas não afeta a qualidade das políticas encontradas.

Os resultados mostram que para instâncias com frações de transições imprecisas entre 1/4 até 1/32, o algoritmo SPUDD-IP não foi capaz de encontrar políticas ótimas para todas as instâncias (experimentos marcados com n/a na Tabela 6.3). Isso ocorre porque com o aumento de transições imprecisas SPUDD-IP faz mais chamadas ao solver linear o que resulta no aumento de custo computacional. No entanto, os valores das políticas encontradas ainda são muito próximos, o que indica que essas frações de transições imprecisas não afetou significativamente a qualidade das políticas encontradas.

vars	transições imprecisas					
	sem imprecisão	1/64	1/32	1/16	1/8	1/4
5	-2,35810	-2,35810	-2,35810	-2,35810	-2,35815	-2,36182
6	-2,37068	-2,37114	-2,37233	-2,37592	-2,37695	-2,40179
7	-2,22767	-2,22928	-2,22799	-2,23554	-2,24039	n/a
8	-2,31703	-2,32143	-2,32004	-2,32419	n/a	n/a
9	-2,05692	-2,06409	-2,06027	n/a	n/a	n/a
10	-1,97818	-1,97900	n/a	n/a	n/a	n/a
11	-2,06949	-2,07787	n/a	n/a	n/a	n/a

Tabela 6.3: Análise da qualidade da política encontrada por SPUDD-IP para redes gênicas imprecisas, com 1/4, 1/8, 1/16, 1/32, 1/64 transições imprecisas e sem imprecisão. Entradas com "n/a" representam instâncias para as quais não foi possível encontrar a política ótima em menos de 6 horas de execução do algoritmo SPUDD-IP.

6.3 Escalabilidade do SPUDD-IP com relação ao Número de Intervenções

Para verificar a escalabilidade da solução robusta com o número de intervenções na instância do MDP-IP influencia no tempo de execução do resolvidor. Para tanto, a partir de um conjunto artificial de amostras foram extraídas 9 instâncias, de 5 a 13 genes, cada qual com 4 ações possíveis. Vinte por cento das transições de estado de cada uma destas instâncias foram selecionadas de forma determinística para adicionar-se imprecisão às probabilidades de transição. A cada uma destas transições foi adicionada uma faixa de incerteza de 10% acima e 10% abaixo do valor de probabilidade inicialmente gerado.

Dadas estas nove instâncias com 4 ações, foram definidas outras instâncias análogas com 3 e 2 ações, obtendo-se 9 com 3 ações e 9 com apenas 2 ações. Todas as instâncias foram resolvidas para um horizonte de tamanho 5. Os resultados podem ser observados na Tabela 6.4, cujo gráfico está na Figura 6.4.

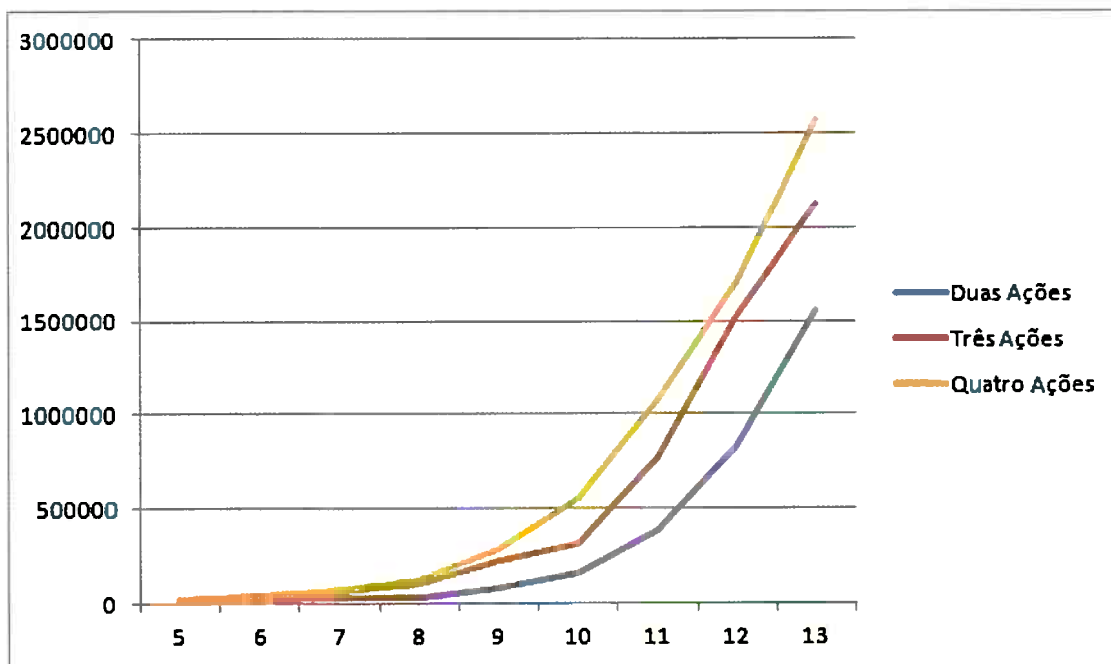


Figura 6.4: Influência da quantidade de ações no tempo de execução do resolvidor: o eixo x corresponde ao número de variáveis de estado (genes) de cada instância e o eixo y ao tempo de execução.

variáveis	duas ações	três ações	quatro ações
5	15291	19424	24691
6	25226	37370	46107
7	27542	59022	75357
8	34686	103777	125840
9	78920	228649	285502
10	168028	319248	549734
11	387890	765135	1073777
12	826754	1525757	1707257
13	1558896	2127197	2568577

Tabela 6.4: *Influência da quantidade de ações no tempo do resolvidor: tabela de valores.*

Capítulo 7

Trabalhos Correlatos

Este capítulo inclui breves exposições dos principais trabalhos relacionados com o tema tratado nesta dissertação.

7.1 *Can Markov Chain Models Mimic Biological Regulation?* [KLD⁺02]

O trabalho [KLD⁺02] apresenta um método para construir uma PBN a partir de dados de amostras. O trabalho é dividido nas seguintes partes:

- **Introdução:** faz uma descrição em alto nível da teoria de redes gênicas, dos modelos de redes booleanas e PBNs, sua relação com cadeias de Markov e menciona o objetivo do trabalho, que é encontrar um modelo matemático para uma rede gênica associada ao melanoma cutâneo.
- **Conjunto de dados:** descreve o conjunto de dados coletados a serem usados para inferir a rede, bem como o método usado para realizar a seleção inicial dos genes mais relevantes ao problema tratado. Os dados incluem a expressão gênica de 6,971 em 31 amostras diferentes. A seleção inicial reduz este número para 587.
- **Seleção de genes para simulação de cadeia de Markov:** descreve o porque não é correto tentar inferir uma PBN com milhares de genes para uma função biológica, sendo necessário, portanto, fazer uso de uma metodologia para selecionar apenas os genes com maior associação ao objeto de estudo, da ordem de dezenas de genes. Descreve os métodos estatísticos e computacionais para selecionar 41 genes que demonstram alto grau de interatividade mútua dentre os 587 previamente selecionados.
- **Cadeia de Markov:** faz uma breve introdução da teoria de cadeias de Markov, descreve como a rede gênica será modelada como tal dado os genes selecionados e faz consideração quanto à implementação que foi usada na análise experimental.
- **Resultados da simulação:** descreve os critérios de seleção de 10 genes dentre os 41 anteriores e como a rede se comportou durante a execução dos testes (estatísticas quanto aos estados mais visitados e expressão média dos genes).

7.2 *External Control in Markovian Genetic Regulatory Networks* [IS02]

O trabalho [IS02] define um modelo de intervenção em redes gênicas probabilísticas, efetivamente transformando uma PBN em um MDP enumerado. O trabalho é dividido nas seguintes partes:

- **Introdução:** expõe o problema tratado e faz um resumo da abordagem exposta pelo trabalho
- **Revisão da teoria de PBNs:** faz uma breve recapitulação do modelo de redes booleanas e em seguida expõe o modelo de redes booleanas probabilísticas em mais detalhes.

- **Formulação do problema de controle em PBNs:** demonstra como pode-se transformar uma PBN em um problema de controle assumindo que um ou mais dos genes da rede gênica é perfeitamente controlável pelo uso de efeitos externos à rede (medicamentos). Assume horizonte finito para o problema e custos nos tratamentos e estados da rede ao final do horizonte, embora não estabeleça nenhum método para definir esses custos.
- **Exemplos:** demonstra como aplicar a teoria exposta na seção anterior em um problema artificial com uma variável de controle e dois genes. Em seguida, faz uma adaptação da rede gênica extraída no trabalho [KLD⁺02] de forma a incluir o problema de controle. Com esta formulação, é demonstrado que é possível exercer controle indireto do estado do gene WNT5a por uso de controle direto de outros genes, quando comparado à transição natural da rede gênica sem uso que qualquer controle.
- **Conclusões:** faz uma recapitulação da teoria apresentada pelo trabalho e os resultados alcançados.

Capítulo 8

Conclusões

O objetivo desse trabalho foi tratar o problema de intervenção em redes gênicas como um problema de decisão sequencial, usando técnicas eficientes da área de planejamento probabilístico baseadas em processos de decisão markovianos e programação dinâmica simbólica.

Foram apresentados os fundamentos da teoria de redes gênicas modeladas como redes booleanas probabilísticas, bem como os fundamentos de processos de decisão markoviano e suas variantes: processos de decisão markovianos enumerativos (MDP enumerativo), em que os possíveis estados do sistema são enumerados; processos de decisão markovianos fatorados (MDP fatorado), em que os estados e funções são descritos em termos de variáveis de estado; e finalmente os processos de decisão markovianos fatorados e com probabilidades imprecisas (MDPIP fatorado).

Apesar de existirem outros trabalhos nessa área, muitos não apresentam de forma clara a modelagem de MDPs a partir de PBNs, bem como não apresentam uma discussão sobre a adequação dos diferentes aspectos da aplicação de redes gênicas quando representados como um processador e decisão markoviano. Além disso, esse é o primeiro trabalho que apresenta uma solução para redes gênicas com probabilidades imprecisas de forma fatorada, usando o algoritmo SPUDD-IP.

Foram apresentados resultados de experimentos em redes gênicas com probabilidades imprecisas modeladas como MDP-IPs, usando o algoritmo SPUDD-IP. Esta análise experimental teve por objetivo explorar quais as limitações e restrições desta aplicação conseguindo, em certos casos, resolver problemas de até 32 genes, sendo que, trabalhos anteriores mostram a solução de problemas de até 10 genes [PDD08], um aumento significativo de eficiência para um problema de complexidade exponencial.

Devido à dificuldade de obter exemplos reais de redes gênicas para a realização da análise experimental, este trabalho elaborou um método de geração de redes gênicas artificiais, buscando manter as características que são esperadas de redes gênicas reais. Isso foi feito com base nos métodos existentes de modelagem de redes gênicas [KLD⁺02, HIG12] que envolvem inferir PBNs a partir de dados amostrais.

Referências Bibliográficas

- [AD04] M. L. Bittner E. R. Dougherty A. Datta, A. Choudhary. External control in markovian genetic regulatory networks: the imperfect information case. *Computational Biology and Bioinformatics*, 20(6):924–930, 2004. xi, 17, 18, 20, 24
- [BCM⁺07] Junior Barrera, Roberto M. Cesar, David C. Martins, Ricardo Z. N. Vêncio, Emilio F. Merino, Márcio M. Yamamoto, Florencia G. Leonardi, Carlos A. de B. Pereira e Hernando A. del Portillo. *Constructing Probabilistic Genetic Networks of Plasmodium falciparum from Dynamical Expression Signals of the Intraerythrocytic Development Cycle*, páginas 11–26. Springer US, Boston, MA, 2007. 9
- [BDS05] Marcel Brun, Edward R. Dougherty e Ilya Shmulevich. Steady-state probabilities for attractors in probabilistic boolean networks. *Signal Process.*, 85(10):1993–2013, Outubro 2005. 8
- [Bel57] R. E. Bellman. Dynamic programming. *Princeton University Press*, 1957. 12
- [BHD99] Craig Boutilier, Steve Hanks e Thomas Dean. Decision-theoretic Planning: Structural Assumptions and Computational Leverage. *JAIR*, 11:1–94, 1999. 2, 24
- [BS01] Séraphin B. Becskei, A. e L. Serrano. Positive feedback in eukaryotic gene networks: cell differentiation by graded to binary response conversion. *The EMBO Journal*, 20(10):2528–2535, May 2001. 1
- [CB99] Steve Hanks Craig Boutilier, Thomas Dean. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94., Jan 1999. 1
- [Coz00] F. G. Cozman. Credal Networks. *Artificial Intelligence*, 120:199–233, 2000. 15, 25
- [Coz05] F. G. Cozman. Graphical Models for Imprecise Probabilities. *International Journal of Approximate Reasoning*, 39(2-3):167–184, 2005. 25
- [DCBD03] A. Datta, A. Choudhary, M. L. Bittner e E. R. Dougherty. External control in markovian genetic regulatory networks. *Kluwer Academic Publishers*, 56(1-2):169–191, 2003. xi, 1, 24
- [DD07] A. Datta e E.R. Dougherty. *Introduction to genomic signal processing with control*. CRC Press, 2007. 4
- [DSdB11] Karina Valdivia Delgado, Scott Sanner e Leliane Nunes de Barros. Efficient Solutions to Factored MDPs with Imprecise Transition Probabilities. *Artif. Intell.*, 175(9-10):1498–1527, 2011. 2, 15, 16, 23, 24, 25
- [DVCP09] S Z Denic, B Vasic, C D Charalambous e R Palanivelu. Robust control of uncertain context-sensitive probabilistic Boolean networks. *IET systems biology*, 3(4):279–95, Julho 2009. 2, 23

- [FES08] Matthias Dehmer Frank Emmert-Streib. *Analysis of Microarray Data: A Network-Based Approach*. Wiley-VCH Verlag GmbH & Co. KGaA, 2008. 9
- [FLT04] Y. Lu Q. Ouyang F. Li, T. Long e C. Tang. The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences of the United States of America*, 101(14):4781–4786, Apr 2004. ix, 1, 3, 4, 5
- [GK73] L. Glass e S. A. Kauffman. The logical analysis of continuous, non-linear biochemical control networks. *Journal of theoretical biology*, 39(1):103–129, Apr 1973. 3
- [HIG12] Carlos Henrique Aguenta HIGA. *Inferência de redes de regulação gênica utilizando o paradigma de crescimento de sementes*. Tese de Doutorado, Instituto de Matemática e Estatística, University of São Paulo, 2012. 9, 37
- [IS02] S. Kim W. Zhang I. Shmulevich, E.R. Dougherty. Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261–274, Feb 2002. viii, 1, 4, 7, 35
- [JH99] A. Hu C. Boutilier J. Hoey, R. St-Aubin. Spudd: Stochastic planning using decision diagrams. *Proceeding of the Fifteenth Conference on Uncertainty in Artificial Intelligences*, páginas 279–288, 1999. 14
- [JJTN03] K. C. Chen J. J. Tyson e B. Novak. Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell. *Current Opinion in Cell Biology*, 15(2):221–231, Apr 2003. 1, 3
- [Jon02] Hidde De Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9:67–103, 2002. 9
- [Kau69] S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3):437–67, Mar 1969. 1
- [KLD⁺02] Seungchan Kim, Huai Li, Edward R. Dougherty, Nanwei Cao, Yidong Chen, Michael Bittner e Edward B. Suh. Can markov chain models mimic biological regulation? *Journal of Biological Systems*, 10(4):337–357, 2002. viii, 1, 9, 35, 36, 37
- [LFS98] Shoudan Liang, Stefanie Fuhrman e Roland Somogyi. Reveal, a general reverse engineering algorithm for inference of genetic network architectures, 1998. 9
- [NEG04] Arnab Nilim e Laurent El Ghaoui. Robustness in Markov Decision Problems with Uncertain Transition Matrices. Em *Neural Information Processing Systems (NIPS)*, 2004. 2
- [Nor98] James R. Norris. *Markov chains*. Cambridge series in statistical and probabilistic mathematics. Cambridge University Press, 1998. 8
- [PDD08] Ranadip Pal, Aniruddha Datta e Edward R. Dougherty. Robust intervention in probabilistic boolean networks. *IEEE Transactions on Signal Processing*, 56(3):1280–1294, 2008. 2, 23, 37
- [PDD09] Ranadip Pal, Aniruddha Datta e Edward R. Dougherty. Bayesian robustness in the control of gene regulatory networks. *IEEE Transactions on Signal Processing*, 57(9):3667–3678, 2009. 2, 23
- [Put94] Martin L. Puterman. *Markov Decision Processes*. John Wiley and Sons, New York, 1994. 18
- [Rei85] W. Reisig. Petri nets: an introduction. *Current Opinion in Cell Biology*, 1985. New York, NY, USA: Springer-Verlag New York, Inc ISBN:0-387-13723-8. 3

- [RM04] Martin Gollery Frank Van Breusegem Ron Mittler, Sandy Vanderauwera. Reactive oxygen gene network of plants. *Trends in plant science*, 9(10):490–498, Nov 2004. 1
- [SDZ02] I. Shmulevich, E. R. Dougherty e W. Zhang. From boolean to probabilistic boolean networks as models of genetic regulatory networks. *Proceedings of the IEEE*, 90(11):1778–1792, 2002. 8, 17
- [SL70] Jay K. Satia e Roy E. Lave Jr. Markovian Decision Processes with Uncertain Transition Probabilities. *Operations Research*, 21:728–740, 1970. 2, 15
- [TAP10] Mehmet Tan, Reda Alhajj e Faruk Polat. Automated large-scale control of gene regulatory networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 40(2):286–297, 2010. 2, 4, 25
- [Wal91] P. Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, London, 1991. 15
- [WED94] Chelsea C. White III e Hank K. El-Deib. Markov Decision Processes with Imprecise Transition Probabilities. *Operations Research*, 42(4):739–749, 1994. 15
- [WGH06] Adriano V. Werhli, Marco Grzegorzczuk e Dirk Husmeier. Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical gaussian models and bayesian networks. *Bioinformatics*, 22(20):2523–2531, 2006. 9
- [YZT06] Q. Ouyang M. Deng F. Li Y. Zhang, M. Qian e C. Tang. Stochastic model of yeast cell-cycle network. *Physica D Nonlinear Phenomena*, 219(14):35–39, Jul 2006. 4, 6, 7
- [ZP94] N. L. Zhang e D. Poole. A simple approach to Bayesian network computations. Em *Proc. of the Tenth Canadian Conference on Artificial Intelligence*, páginas 171–178, 1994. 14, 16