



Universidade Federal  
do Rio de Janeiro  

---

Escola Politécnica

MODELAGEM E CONTROLE DE SISTEMAS PÚBLICOS DE  
COMPARTILHAMENTO DE BICICLETAS UTILIZANDO MODELOS EM  
REDES DE PETRI

Braian Igreja de Freitas

Projeto de Graduação apresentado ao Curso de Engenharia de Controle e Automação da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientador: João Carlos dos Santos Basilio

Rio de Janeiro  
Março de 2019

MODELAGEM E CONTROLE DE SISTEMAS PÚBLICOS DE  
COMPARTILHAMENTO DE BICICLETAS UTILIZANDO MODELOS EM  
REDES DE PETRI

Braian Igreja de Freitas

PROJETO DE GRADUAÇÃO SUBMETIDO AO CORPO DOCENTE DO  
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO DA ESCOLA  
POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO  
PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU  
DE ENGENHEIRO DE CONTROLE E AUTOMAÇÃO.

Examinado por:



Prof. João Carlos dos Santos Basilio, Ph.D.



Prof. Lilian Kawakami Carvalho, D.Sc.



Prof. Claudio Miceli de Farias, D.Sc.

RIO DE JANEIRO, RJ - BRASIL  
MARÇO DE 2019

de Freitas, Braian Igreja

Modelagem e controle de sistemas públicos de compartilhamento de bicicletas utilizando modelos em redes de Petri/Braian Igreja de Freitas. – Rio de Janeiro: UFRJ/ Escola Politécnica, 2019.

XIV, 83 p.: il.; 29,7cm.

Orientador: João Carlos dos Santos Basilio

Projeto de Graduação – UFRJ/ Escola Politécnica/  
Curso de Engenharia de Controle e Automação, 2019.

Referências Bibliográficas: p. 82 – 83.

1. Redes de Petri. 2. Estratégia de balanceamento.  
3. Bicicleta. I. Basilio, João Carlos dos Santos. II.  
Universidade Federal do Rio de Janeiro, Escola Politécnica,  
Curso de Engenharia de Controle e Automação. III. Título.

*Aos meus pais, que fizeram de  
tudo para que eu chegasse até  
esta etapa da minha vida.*

*Ao meu irmão, que sempre  
esteve na minha vida como um  
grande amigo.*

*Aos meus avós, que me fizeram  
crescer com suas sabedorias.*

# Agradecimentos

Primeiramente, gostaria de agradecer aos meus pais, Paulo Freitas e Denise Freitas, por sempre estarem dispostos a me ajudar, mesmo nos momentos difíceis. Certamente vocês foram e são fundamentais para a minha vida.

Agradeço ao meu irmão Breno Igreja de Freitas, que mesmo ocupado com suas ambições profissionais, sempre encontra um tempo para ser um grande amigo.

Agradeço às minhas avós, Irma e Adenair, cujas presenças sempre me trazem bastante felicidade. Ao meu avô Januário, que sempre foi um exemplo de pessoa para mim. Ao meu avô Walcy, que, apesar de não estar mais presente para poder ver a minha colação de grau, foi um excelente engenheiro e uma grande pessoa que me motivou e motiva até hoje a ter orgulho da minha futura profissão.

Agradeço à Ingrid Antunes, que se comprometeu a me ajudar neste projeto sempre que eu precisava. Apesar de não estar mais no meio acadêmico, ainda a considero como uma orientadora deste trabalho.

Agradeço ao meu orientador, o professor João Carlos dos Santos Basilio, por ter me ajudado com a continuidade deste projeto, utilizando sua experiência para me ensinar sobre muitos detalhes do meio acadêmico que não sabia. Agradeço também por ter me encorajado a continuar estudando após a minha graduação.

Agradeço à equipe de competição MinervaBots, que me forneceu a oportunidade de trabalhar em projetos com futuros engenheiros de diversas áreas e de ficar em primeiro lugar em uma competição nacional em nome da minha universidade.

Agradeço aos professores, que me mostraram, através das suas excelentes aulas, porque a universidade é considerada uma das melhores do país.

Agradeço a todos os meus colegas da faculdade que tive o prazer de conhecer, principalmente os da turma T-18. Amy, muito obrigado pelos momentos que você me trouxe desde que te conheci, além de ter me divertido muito, aprendi bastante com você. Patusco, muito obrigado por ter me apresentado tantas oportunidades, além de me apresentar um estágio em que construímos um grande avião, graças a você obtive grandes amizades, inclusive a sua. Victor, obrigado por ter me ensinado sobre mecânica e por ter me ajudado nas lutas que travamos juntos. Finalmente, agradeço ao Vinícius que, além de ser o primeira pessoa com quem trabalhei na faculdade, me motivou muito com o empenho que sempre demonstra nas suas atividades.

Resumo do Projeto de Graduação apresentado à Escola Politécnica/ UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro de Controle e Automação.

MODELAGEM E CONTROLE DE SISTEMAS PÚBLICOS DE  
COMPARTILHAMENTO DE BICICLETAS UTILIZANDO MODELOS EM  
REDES DE PETRI

Braian Igreja de Freitas

Março/2019

Orientador: João Carlos dos Santos Basilio

Curso: Engenharia de Controle e Automação

Uma das diferentes modalidades de compartilhamento de bicicletas possui como princípio permitir que um indivíduo possa retirar uma bicicleta de um lugar e devolvê-la a outro, possibilitando que ele trafegue de um ponto a outro. Desse princípio, originam-se os sistemas públicos de compartilhamento de bicicletas (SPCB), que consistem de estações de bicicletas espalhadas geograficamente em regiões onde os sistemas operam. Nessas estações, os usuários podem adicionar ou remover bicicletas alugadas para se locomoverem entre elas. Essa liberdade dos usuários, porém, pode comprometer a eficiência do sistema, pois dependendo do uso das estações, elas podem ficar sem bicicletas ou lotadas, impedindo, respectivamente, as retiradas e devoluções de bicicletas nas estações com problemas. Uma maneira de solucionar esse problema é implementar estratégias de balanceamento, através de um veículo, para realizar trocas de bicicletas entre as estações, estabilizando-as. Tal estratégia, porém, deve ser planejada atendendo às necessidades atuais e futuras do sistema. Para avaliar a eficiência de diferentes estratégias em um mesmo simulador, neste trabalho, é proposto o modelo de um SPCB genérico em um novo tipo de rede de Petri, a rede de Petri generalizada com controle externo, que permite a atuação de um agente externo nas ações da rede, controlando as estratégias do veículo. Através das equações dinâmicas apresentadas, o modelo possibilita a generalização dos sistemas, permitindo a modelagem de sistemas com diferentes características, como, por exemplo, a quantidade das estações. Para validar o modelo, é feita a implementação e avaliação do projeto Integra UFRJ com a combinação de três tipos diferentes de estratégias de balanceamento formuladas neste trabalho.

Abstract of Undergraduate Project presented to POLI/UFRJ as a partial fulfillment of the requirements for the degree of Engineer.

## MODELING AND CONTROLLING OF PUBLIC BICYCLE SHARING SYSTEMS USING PETRI NET MODELS

Braian Igreja de Freitas

March/2019

Advisor: João Carlos dos Santos Basilio

Course: Automation and Control Engineering

One of the different modalities of bicycle sharing has, as a principle, to allow a person to remove a bicycle from a place and return it to another place, making it possible for them to travel between one point to the other. From that principle, public bicycle sharing systems (PBSS) are originated. They are made of stations of bicycles geographically scattered around regions where the systems operate. In those stations, users can add or remove rented bicycles to move themselves around them. However, that freedom that the users have can compromise the efficiency of the system, because depending on the usage of the stations, they may have all the bicycles removed from the stations or may become full, preventing, respectively, the removal or devolution of bicycles to the stations with problems. One way to solve that matter is to implement balancing strategies, through a vehicle, to exchange bicycles between stations, stabilizing them. However, such strategy must be planned regarding the present and future necessities of the system. To evaluate the efficiency of different strategies in one simulation, in this work is proposed a model of a generic PBSS of a new type of Petri nets, a generalized stochastic Petri net with external control, that enables the operation of an external agent at the actions on the net, controlling the strategies of the vehicle. Through the dynamics equations presented, the model enables the generalization of the systems, allowing the modeling of systems with different characteristics, such as the number of stations. To validate the model, an implementation and evaluation of the project Integra UFRJ is made with the combination of three different types of strategies of balancing formulated on this work.

# Sumário

<b>Lista de Figuras</b>	<b>xi</b>
<b>Lista de Tabelas</b>	<b>xiii</b>
<b>1 Introdução</b>	<b>1</b>
<b>2 Fundamentos teóricos</b>	<b>4</b>
2.1 Rede de Petri . . . . .	4
2.1.1 Estrutura da rede de Petri . . . . .	4
2.1.2 Redes de Petri com marcação . . . . .	6
2.1.3 Dinâmicas das redes de Petri . . . . .	6
2.1.4 Conflito em redes de Petri . . . . .	8
2.2 Rede de Petri com arcos inibidores . . . . .	8
2.3 Rede de Petri temporizada-T . . . . .	10
2.4 Redes de Petri estocásticas generalizadas . . . . .	10
2.5 Redes de Petri estocásticas generalizadas com controles externos . . .	14
2.6 Algoritmos de busca por força bruta . . . . .	15
<b>3 Sistema público de compartilhamento de bicicletas</b>	<b>16</b>
3.1 Descrição do sistema . . . . .	16
3.2 Modelo de Antunes et al. implementando uma estratégia de balan- ceamento cíclica . . . . .	18
3.2.1 Sub-rede do fluxo de bicicletas . . . . .	22
3.2.2 Sub-rede do controle de estações . . . . .	24
3.2.3 Sub-rede do circuitos de redistribuição . . . . .	25
3.3 Análise crítica sobre o modelo da Antunes et al. . . . .	26
<b>4 Modelo utilizando rede de Petri estocástica generalizada com con-     trole externo</b>	<b>28</b>
4.1 Modelo proposto . . . . .	28
4.1.1 Sub-rede do fluxo de bicicletas . . . . .	32
4.1.2 Sub-rede do controle de estações . . . . .	33



4.1.3	Sub-rede do fluxo do veículo de redistribuição . . . . .	34
4.2	Matrizes de incidência do modelo proposto para um SPCB com $N$ estações . . . . .	35
4.3	Ordem do modelo proposto . . . . .	42
<b>5</b>	<b>Estudo do sistema de bicicletas compartilhadas Integra UFRJ para a implementação de um modelo para simulações</b>	<b>43</b>
5.1	Descrição do projeto Integra UFRJ . . . . .	43
5.2	Estrutura e uso do Integra UFRJ . . . . .	44
5.3	Comportamento dos usuários . . . . .	46
5.4	Tempos gastos pelo veículo de redistribuição e pelas bicicletas para transitarem pelo sistema . . . . .	50
<b>6</b>	<b>Formulações de algoritmos para serem utilizados no balanceamento do sistema do modelo proposto</b>	<b>53</b>
6.1	Estratégia simples que realiza um balanceamento cíclico . . . . .	53
6.2	Estratégia adaptativa baseado no algoritmo de busca por força bruta	56
6.3	Estratégia preditiva que varia os números de reordenação de acordo com a demanda . . . . .	57
<b>7</b>	<b>Implementação e avaliação do algoritmo de simulação do modelo de um sistema público de compartilhamento de bicicletas com <math>N</math> estações</b>	<b>60</b>
7.1	Algoritmo de simulador . . . . .	61
7.1.1	Transições imediatas que não estão envolvidas em conflitos . .	62
7.1.2	Transições temporizadas estocásticas . . . . .	63
7.1.3	Transições imediatas que estão envolvidas em conflitos solucionados estocasticamente . . . . .	65
7.1.4	Transições imediatas que estão envolvidas em conflitos solucionados por meio de um controle externo . . . . .	66
7.2	Parâmetros de entrada do simulador . . . . .	68
7.3	Indicadores para a avaliação dos desempenhos dos sistemas simulados	70
7.4	Resultados . . . . .	71
7.4.1	Sistema sem balanceamento . . . . .	72
7.4.2	Sistema com balanceamento cíclico e números de reordenação fixos . . . . .	73
7.4.3	Sistema com balanceamento adaptativo e números de reordenação fixos . . . . .	75
7.4.4	Sistema com balanceamento cíclico e números de reordenação preditivos . . . . .	76

7.4.5	Sistema com balanceamento adaptativo e números de reordenação preditivos . . . . .	78
<b>8</b>	<b>Conclusão e trabalhos futuros</b>	<b>80</b>
	<b>Referências Bibliográficas</b>	<b>82</b>

# Lista de Figuras

2.1	Exemplo de um grafo de uma rede de Petri. . . . .	5
2.2	Exemplo de um grafo de uma rede de Petri com um arco de peso maior que 1. . . . .	5
2.3	Exemplo de uma rede de Petri com marcação. . . . .	6
2.4	Exemplo de uma rede de Petri com transição não habilitada. . . . .	7
2.5	Rede de Petri da figura 2.3 após o disparo da transição $t_1$ . . . . .	7
2.6	Conflito estrutural entre as transições $t_1$ e $t_2$ em relação ao lugar $p_1$ . . . . .	8
2.7	Rede de Petri com conflito efetivo entre as transição $t_1$ e $t_2$ (a) e sem conflito efetivo (b). . . . .	8
2.8	Rede de Petri com transição $t_1$ habilitada (a) e com a mesma transição não habilitada (b). . . . .	9
2.9	Exemplo de uma transição temporizada $t_1$ . . . . .	10
2.10	Exemplo de uma rede de Petri estocástica generalizada. . . . .	13
3.1	Fluxo de bicicletas em um SPCB de três estações com um veículo de redistribuição. . . . .	18
3.2	Rede de Petri estocástica generalizada com arcos inibidores da estação $S_1$ de um SPCB de três estações proposto em [1, 2] . . . . .	19
4.1	Rede de Petri estocástica generalizada com controle externo da estação $S_1$ de um SPCB com três estações. . . . .	29
4.2	Sub-rede do fluxo de bicicletas da estação $S_1$ . . . . .	32
4.3	Sub-rede do controle de estações da estação $S_1$ . . . . .	33
4.4	Sub-rede do fluxo do veículo de redistribuição da estação $S_1$ e do centro de redistribuição. . . . .	34
5.1	Mapa com as oito estações de bicicletas do Integra UFRJ na Cidade Universitária. . . . .	44
6.1	Sub-rede demonstrando o trajeto do veículo na estratégia cíclica. . . . .	55
7.1	Gráfico da distribuição de Poisson para um intervalo de 0 a 20 e $\lambda = 10$ . . . . .	67

7.2	Gráfico da quantidade de bicicletas em cada estação durante a simulação do sistema sem balanceamento. . . . .	72
7.3	Gráfico da quantidade de bicicletas em cada estação (azul) em relação ao número de reordenação (laranja), da quantidade de bicicletas no veículo e das estações que foram visitadas pelo veículo ao longo da simulação utilizando estratégia de balanceamento cíclico com números de reordenação constantes. . . . .	74
7.4	Gráfico da quantidade de bicicletas em cada estação (azul) em relação ao número de reordenação (laranja), da quantidade de bicicletas no veículo e das estações que foram visitadas pelo veículo ao longo da simulação utilizando estratégia de balanceamento adaptativo com números de reordenação constantes. . . . .	75
7.5	Gráfico da quantidade de bicicletas em cada estação (azul) em relação ao número de reordenação (laranja), da quantidade de bicicletas no veículo e das estações que foram visitadas pelo veículo ao longo da simulação utilizando estratégia de balanceamento cíclico com números de reordenação preditivos. . . . .	77
7.6	Gráfico da quantidade de bicicletas em cada estação (azul) em relação ao número de reordenação (laranja), da quantidade de bicicletas no veículo e das estações que foram visitadas pelo veículo ao longo da simulação utilizando estratégia de balanceamento adaptativo com números de reordenação preditivos. . . . .	78

# Lista de Tabelas

3.1	Interpretações dos parâmetros da rede de Petri proposta em [1, 2]. . .	19
3.2	Interpretações dos lugares da rede de Petri proposta em [1, 2]. . . . .	20
3.3	Interpretações das transições da rede de Petri proposta em [1, 2]. . .	21
4.1	Interpretações dos parâmetros da rede de Petri proposta. . . . .	29
4.2	Interpretações dos lugares da rede de Petri proposta. . . . .	30
4.3	Interpretações das transições da rede de Petri proposta. . . . .	31
5.1	Número e local físico de cada estação. . . . .	45
5.2	Porcentagem de retirada de bicicletas de cada estação. . . . .	46
5.3	Porcentagem de retirada de bicicletas ao longo do dia. . . . .	47
5.4	Porcentagem de devoluções de bicicletas à cada estação. . . . .	48
7.1	Indicadores médios das 50 simulações do sistema sem balanceamento.	73
7.2	Indicadores médios das 50 simulações do sistema com balanceamento cíclico com números de reordenação constantes. . . . .	74
7.3	Indicadores médios das 50 simulações do sistema com balanceamento adaptativo com números de reordenação constantes. . . . .	76
7.4	Indicadores médios das 50 simulações do sistema com balanceamento cíclico com números de reordenação preditivos. . . . .	77
7.5	Indicadores médios das 50 simulações do sistema com balanceamento adaptativo com números de reordenação preditivos. . . . .	79

# Lista de Símbolos

$\mathbb{Z}_+^*$	Conjunto dos números inteiros positivos e não nulos
$\mathbb{Q}_+$	Conjunto dos números racionais positivos
$P$	Conjuntos de lugares de uma rede de Petri
$T$	Conjuntos de transições de uma rede de Petri
$T_i$	Conjuntos de transições imediatas de uma rede de Petri
$T_d$	Conjuntos de transições temporizadas de uma rede de Petri
$n_P$	Números de elementos do conjunto de lugares $P$
$n_T$	Números de elementos do conjunto de transições $T$
$p_i$	Lugar de uma rede de Petri
$t_j$	Transição de uma rede de Petri
$Pre(p_i, t_j)$	Função de um arco ordinário que conecta um lugar $p_i$ a uma transição $t_j$
$Post(t_j, p_i)$	Função de um arco ordinário que conecta uma transição $t_j$ a um lugar $p_i$
$Inhib(p_i, t_j)$	Função de um arco inibidor que conecta um lugar $p_i$ a uma transição $t_j$
$I(t_j)$	Conjunto dos lugares de entrada de $t_j$
$O(t_j)$	Conjunto dos lugares de saída de $t_j$
$I(p_i)$	Conjunto das transições de entrada de $p_i$
$O(p_i)$	Conjunto das transições de saída de $p_i$
$IH(t_j)$	Conjunto dos lugares inibidores de $t_j$
$\underline{m}$	Vetor contendo a marcação de uma rede de Petri
$\underline{m}_0$	Vetor contendo a marcação inicial de uma rede de Petri
$m(p_i)$	Número de fichas do lugar $p_i$
$A$	Matriz de incidência de uma rede de Petri
$\sigma_k$	Vetor de disparos das transições em um instante $k$
$C(p_i, T_k)$	Conflito estrutural entre as transições do conjunto $T_k$ em relação à marcação de um lugar $p_i$
$d_j$	Intervalo de tempo de retardo de disparo da transição $t_j$ .

$\Lambda_d(t_j)$	Função estocástica de uma transição temporizada $t_j$
$\Lambda_i(t_j, p_i)$	Função estocástica de uma transição imediata $t_j$ envolvida em um conflito em relação ao lugar $p_i$
$f_{dp_{t_j}}(x)$	Função densidade de probabilidade de uma variável aleatória $x$ e de uma transição $t_j$ .
$PRC(t_j p_i)$	Probabilidade de $t_j$ ser sorteada em relação a um conflito em relação ao lugar $p_i$
$Esc(t_j, p_i)$	Ação de um agente externo sob o disparo da transição $t_j$ envolvida em um conflito relativo ao lugar $p_i$
$blockdiag\{a_1, a_2, \dots, a_k\}$	Matriz diagonal cujos elementos da diagonal são $a_1, a_2, \dots, a_k$

# Capítulo 1

## Introdução

Os sistemas públicos de compartilhamento de bicicletas (SPCB) tem como princípio permitir que um indivíduo possa retirar uma bicicleta de um lugar e devolvê-la a outro, possibilitando que ele trafegue de um ponto a outro. Desse princípio, originam-se SPCB, que consistem de estações de bicicletas espalhadas geograficamente onde os usuários podem adicionar ou remover bicicletas alugadas para se locomoverem entre elas.

Como uma modalidade de transporte, os sistemas públicos de bicicletas trazem uma variedade de vantagens em relação a outros meios de transporte. Por ser um transporte de bicicletas, os usuários podem trafegar pelas regiões sem emitir poluição gasosa e sonora como fazem os veículos de combustão. Além disso, como elas são movidas pelo esforço físico do usuário, o uso do sistema faz com que o usuário se exercite, promovendo o esporte de ciclismo.

Pode-se afirmar que, graças às vantagens que eles trazem, atualmente os SPCBs se tornaram meios de transporte comuns em diversas regiões do mundo, totalizando mais de 1600 programas de bicicletas compartilhadas em diferentes países, cujas quantidades de bicicletas totalizam aproximadamente dois milhões de bicicletas. Tal progresso ocorreu devido ao incentivo por alguns países no programa, como por exemplo a China, que, com o declínio da utilização de bicicletas que começou nos anos noventa e o aumento dos congestionamentos devido ao crescimento populacional, incentivou o início de programas de bicicletas compartilhadas em 2008. Um ano após o início dos programas, aproximadamente 250000 pessoas usavam o sistema diariamente [3]. Atualmente, são feitas aproximadamente 42 milhões de viagens no país por dia.

Apesar da China atualmente utilizar sistemas *dockless*, que não necessitam de estações, muitos programas de outros países ainda utilizam sistemas da terceira geração, baseado em estações. Por esse motivo, neste trabalho, são estudados apenas sistemas da terceira geração. Nesse tipo de sistema, as viagens realizadas pelos usuários são normalmente utilizadas por eles para se deslocarem de um ponto ao outro,



sem retornarem ao ponto original logo após o uso. Isto significa que cada usuário do sistema contribui para desbalancear a quantidade de bicicletas das estações do sistema.

Devido ao desbalanceamento das estações causados pelos usuários e o número limitado de vagas de bicicletas em cada estação, podem ocorrer dois problemas nessas estações: as estações podem se tornar vazias em determinados momentos do dia, impedindo os usuários de retirar bicicletas, ou as estações podem se tornar cheias de bicicletas, impedindo que os usuários devolvam as bicicletas alugadas a essas estações. Ambos os problemas reduzem a eficiência do sistema, o que pode diminuir o interesse dos usuários utilização de bicicletas públicas.

Mesmo sendo possível solucionar o problema alterando a capacidade das estações e a quantidade de bicicletas do sistema, existe uma outra solução, de curto prazo, que não altera a estrutura do sistema. Essa solução consiste na adição de um veículo de redistribuição capaz de transportar bicicletas entre as estações, diminuindo ou aumentando a quantidade de bicicletas nelas. Para que veículo seja capaz de balancear as estações com êxito, o caminho realizado pelo veículo entre estações deve ser planejado de tal forma que ele não visite uma estação vazia enquanto ele estiver sem bicicletas ou uma estação cheia enquanto ele estiver cheio.

Assim, este trabalho será uma abordagem para descrever o funcionamento dos SPCBs por meio de modelos a eventos discretos, o que permitirá, posteriormente, a simulação e avaliação do sistema. Para obter o modelo discreto de um SPCB, as diferentes localizações da bicicleta e do veículo no sistema são associadas a espaços de estados discretos, enquanto ações, como a escolha para qual estação um usuário se desloca ou a adição de uma bicicleta a uma estação pelo veículo, causam mudanças no estado do sistema ao longo do tempo. Como os usuários tomam suas decisões de como agir no sistema independentemente, algo que pode ser descrito estocasticamente, e os espaço de estado discreto é composto de recursos (bicicletas e o veículo de redistribuição), o formalismo mais indicado para o sistema é o de redes de Petri. Portanto, neste trabalho, os modelos de SPCBs são compostos de formalismos de redes de Petri.

Anteriormente a este trabalho, foi proposto, por Antunes et al. [1, 2], uma modelagem em redes de Petri estocásticas generalizadas de um SPCB com um veículo que executa, através de rondas, balanceamentos cíclicos, visitando as estações em uma ordem sequencial, e realizando os balanceamentos em momentos específicos do dia. Mesmo que o modelo tenha apresentado resultados satisfatórios, a estratégia de balanceamento implementada é simples e, em algumas situações, ineficiente, pois como a estratégia faz o veículo visitar todas as estações sem considerar seus estados, ele pode visitar estações que não precisam de balanceamento ou que não podem ser balanceadas no momento, desperdiçando tempo que poderia ser utilizado para

balancear outras estações.

O objetivo deste trabalho, portanto, é ser a continuação aos trabalhos realizados em [1, 2], alterando o seu modelo de SPCB em redes de Petri estocástica generalizada para, além de otimizá-lo, permitir que ele possa ser usado para implementar e testar diferentes tipos de estratégias de balanceamento. Nesse sentido, para que seja possível implementar diferentes tipos de estratégias de balanceamento, torna-se necessário desenvolver alguma forma de modelar uma rede de Petri que seja capaz de permitir que diferentes algoritmos externos que sejam capazes de alterar o estado do modelo possam ser implementados. Para isso ser possível, é proposto um novo tipo de formalismo, as redes de Petri estocásticas com controles externos, que permite modelar um sistema a eventos discretos com redes de Petri e algoritmos que interagem entre si.

Para validar o modelo apresentado, será usado o SPCB recentemente implementado na Cidade Universitária do Rio de Janeiro, o projeto Integra UFRJ. Com o modelo do projeto feito, serão feitos cinco tipos de simulações para testar cinco combinações diferentes de estratégias de balanceamento, permitindo não somente validar a modelagem proposta neste trabalho, como também comparar os rendimentos e tirar conclusões sobre a eficiência de cada estratégia.

Este trabalho é estruturado da seguinte forma. No capítulo 2, serão revistos os principais conceitos de redes de Petri, necessários para o entendimento do modelo, e será introduzido um novo tipo de rede de Petri e apresentada uma breve discussão sobre algoritmos de busca por força bruta, a ser utilizado em um dos algoritmos de balanceamento. Em seguida, no capítulo 3, serão apresentados os conceitos básicos de SPCBs e o modelo de SPCBs em redes de Petri estocásticas generalizadas proposto em [1, 2]. No capítulo 4, será apresentado o modelo para SPCBs em redes de Petri estocásticas generalizadas com controles externos, proposto por este trabalho. Em seguida, no capítulo 5, será feito um estudo das características do sistema do projeto Integra UFRJ, a ser utilizado na validação do modelo proposto. Após isso, no capítulo 6, serão propostos três tipos de algoritmos de balanceamento para serem utilizados em conjunto com o modelo. A partir de todas as informações dos capítulos anteriores, no capítulo 7, será implementado um algoritmo genérico que simula o modelo e as estratégias propostas, que é utilizado para realizar diferentes simulações do sistema do projeto Integra UFRJ. Finalmente, no capítulo 8, serão apresentadas as conclusões do trabalho e sugestões para futuros trabalhos.

# Capítulo 2

## Fundamentos teóricos

Neste capítulo, serão apresentados os conceitos básicos de redes de Petri e dos algoritmos de busca por força bruta, ambos utilizados na elaboração deste trabalho. Para isso, este capítulo é estruturado da seguinte forma: na seção 2.1, a rede de Petri será definida e algumas das suas propriedades serão apresentadas. Na seção 2.2, será mostrada a rede de Petri com arcos inibidores. Na seção 2.3, será apresentado o conceito de redes de Petri temporizadas-T e, após isso, na seção 2.4 será explicada a rede de Petri estocástica generalizada, que será utilizada para a definição da rede de Petri estocástica generalizada com controle externo, apresentada na seção 2.5 e utilizada na modelagem do SPCBs deste trabalho. Por fim, na seção 2.6, serão demonstrados os princípios de algoritmos de busca por força bruta, que servirá como base para o algoritmo proposto em uma das estratégias de balanceamento consideradas neste trabalho.

### 2.1 Rede de Petri

A rede de Petri é um formalismo capaz de representar uma grande variedade de sistemas a eventos discretos. Além disso, as redes de Petri possuem uma quantidade significativa de técnicas de análise desenvolvidas para auxiliar nos seus estudos [4].

#### 2.1.1 Estrutura da rede de Petri

Uma rede de Petri é um grafo bipartido com dois tipos diferentes de vértices conectados por arcos: os lugares e as transições. Os lugares são representados graficamente por círculos e representam os diferentes estados contidos no sistema. As transições são representadas por barras e são associadas, na maioria dos casos, aos eventos. Os arcos são representados por setas orientadas de acordo com as relações entre os lugares e as transições. Os arcos nunca ligam vértices do mesmo tipo, isto é, lugares somente são conectados a transições e vice-versa. Formalmente, as redes de Petri

são definidas como [5]:

**Definição 2.1** (*Rede de Petri*) Uma rede de Petri é uma quádrupla

$$\mathcal{N} = (P, T, Pre, Post),$$

em que:

- $P = (p_1, p_2, \dots, p_{n_P})$  é o conjunto finito de lugares;
- $T = (t_1, t_2, \dots, t_{n_T})$  é o conjunto finito de transições;
- $Pre: (P \times T) \rightarrow \mathbb{Z}_+^*$  é a função de arcos ordinários que conectam lugares a transições e
- $Post: (T \times P) \rightarrow \mathbb{Z}_+^*$  é a função de arcos ordinários que conectam transições a lugares.

Quando um arco tem origem em um lugar  $p_i \in P$  e destino em uma transição  $t_j \in T$ ,  $p_i$  é chamado de lugar de entrada de  $t_j$  e quando um arco tem origem em uma transição  $t_j \in T$  e destino em um lugar  $p_i \in P$ , este é chamado de lugar de saída de  $t_j$ . Os conjuntos de lugares de entrada e de saída de uma transição  $t_j \in T$  são denotados por  $I(t_j)$  e  $O(t_j)$ , respectivamente. Analogamente, os conjuntos de transições de entrada e de saída de um lugar  $p_i \in P$  são denotados por  $I(p_i)$  e  $O(p_i)$ , respectivamente. Repare que  $I(t_j)$  é formado de lugares  $p_i \in P$  tais que  $Pre(p_i, t_j) > 0$  e que  $O(t_j)$  é formado de lugares  $p_i \in P$  tais que  $Post(t_j, p_i) > 0$ . Da mesma forma,  $I(p_i)$  e  $O(p_i)$  são formados de transições  $t_j \in T$  tais que  $Post(t_j, p_i) > 0$  e  $Pre(p_i, t_j) > 0$ , respectivamente.

Um exemplo de um grafo de redes de Petri é mostrado na figura 2.1. Sua estrutura é definida por  $P = \{p_1, p_2\}$ ,  $T = \{t_1\}$ ,  $Pre(p_1, t_1) = 2$  e  $Post(t_1, p_2) = 1$ . Os dois arcos que ligam o lugar  $p_1$  à transição  $t_1$  podem ser substituídos por um único arco contendo a informação da quantidade de arcos, a qual é definida como peso, cujo valor é igual a  $Pre(p_1, t_1)$ . Essa substituição torna a visualização do grafo mais simples, como mostrado na figura 2.2.

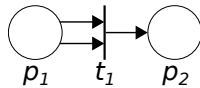


Figura 2.1: Exemplo de um grafo de uma rede de Petri.

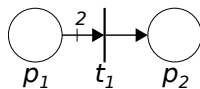


Figura 2.2: Exemplo de um grafo de uma rede de Petri com um arco de peso maior que 1.

### 2.1.2 Redes de Petri com marcação

Ao se modelar um sistema com redes de Petri, é necessário representar os estados do sistema antes e depois de mudanças que ocorrem após cada evento e verificar quais transições estão habilitadas. Para isso, fichas são colocadas nos lugares, possibilitando diferenciar os estados do grafo pelo número de fichas em cada lugar e permitindo verificar se as condições para a ativação das transições são satisfeitas. Uma transição  $t_j \in T$  está ativa quando o número de fichas de cada um dos seus lugares de entrada for maior ou igual do que o peso do seu respectivo arco que liga até a transição.

O estado atual da rede é representado por um vetor de marcação de lugares  $\underline{m} = [m(p_1), m(p_2), \dots, m(p_{n_P})]^T$ , cujas componentes  $m(p_i)$  representam a quantidade de fichas em cada lugar  $p_i \in P$ . O vetor de marcação permite diferenciar os estados do sistema e sua evolução, pois cada vez que o sistema muda de estado há um novo vetor de marcação de lugares na rede de Petri correspondente. Isso leva à definição de redes de Petri com marcação.

**Definição 2.2** (*Rede de Petri com marcação*) Uma rede de Petri com marcação é uma *quintupla*:

$$\mathcal{R} = (P, T, Pre, Post, \underline{m}_0),$$

em que:

- $(P, T, Pre, Post)$  é um grafo de rede de Petri;
- $\underline{m}_0 = [m_0(p_1), m_0(p_2), \dots, m_0(p_{n_P})]^T$  é o vetor de marcação inicial da rede.

A figura 2.3 mostra uma rede de Petri com marcação inicial  $m_0 = [2, 1]^T$

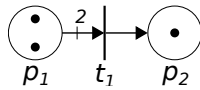


Figura 2.3: Exemplo de uma rede de Petri com marcação.

### 2.1.3 Dinâmicas das redes de Petri

Uma transição  $t_j$  é dita habilitada quando ela está ativa, isto é, quando o número de fichas de cada um dos seus lugares de entrada for maior ou igual do que os pesos dos seus respectivos arcos que ligam esses lugares à transição  $t_j$ . A definição de transição habilitada é dada a seguir.

**Definição 2.3** (*Transição habilitada*) Uma transição  $t_j$  está habilitada se,  $\forall p_i \in I(t_j)$ ,

$$p_i \geq Pre(p_i, t_j). \quad (2.1)$$

Se uma transição  $t_j \in T$  estiver habilitada, ela pode disparar. Quando ela dispara, as fichas se movimentam de acordo com as funções  $Pre(p_i, t_j)$  e  $Post(t_j, p_i)$ , que indicam, respectivamente, as quantidades de fichas que são consumidas e acrescentadas a cada lugar  $p_i \in P$  após o disparo de  $t_j$ . É importante ressaltar que o disparo de uma transição habilitada não é obrigatório.

Na marcação da figura 2.3, a transição  $t_1$  está habilitada pois  $m(p_1) = 2$  e  $Pre(p_1, t_1) = 2$ , enquanto que a figura 2.4, mostra uma rede de Petri na qual transição  $t_1$  não está habilitada, uma vez que  $m(p_1) = 0 \leq Pre(p_1, t_1) = 1$ , o que impossibilita o disparo da transição  $t_1$ .

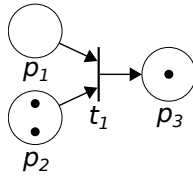


Figura 2.4: Exemplo de uma rede de Petri com transição não habilitada.

O disparo de uma transição habilitada pode causar mudanças no estado do sistema, pois quando uma transição dispara, seus lugares de entrada perdem fichas e seus lugares de saída ganham fichas de acordo com os pesos de seus respectivos arcos. Essa movimentação pode fazer com que a rede de Petri alcance uma nova marcação que modela o estado atual do sistema correspondente.

O disparo de uma transição habilitada  $t_j$  resulta em uma nova marcação  $\underline{m}'$  cujos elementos são obtidos de acordo com a seguinte equação:

$$m'(p_i) = m(p_i) - Pre(p_i, t_j) + Post(t_j, p_i), \forall p_i \in P. \quad (2.2)$$

A figura 2.5 mostra a marcação  $\underline{m}' = [0, 2]$  da rede de Petri da figura 2.3 resultante do disparo da transição  $t_1$ , que está habilitada para a marcação inicial  $\underline{m}_0 = [2, 1]$ . Quando a transição  $t_1$  da figura 2.3 dispara, o lugar de entrada  $p_1$  perde duas fichas, equivalente ao peso do arco que o liga a  $t_1$ , e lugar de saída  $p_2$  ganha uma ficha, de acordo com o peso do arco que liga  $t_1$  a  $p_2$ .

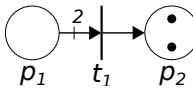


Figura 2.5: Rede de Petri da figura 2.3 após o disparo da transição  $t_1$ .

Para uma rede de Petri  $R$  com  $n_P$  lugares e  $n_T$  transições, a matriz de incidência  $A = [a_{ij}]$  é uma matriz de números inteiros não negativos de dimensões  $n_P \times n_T$  dada por  $a_{ij} = a_{ij}^+ - a_{ij}^-$ , em que  $a_{ij}^+ = Post(t_j, p_i)$  e  $a_{ij}^- = Pre(p_i, t_j)$ [6]. A equação fundamental que descreve a dinâmica é dada por:

$$\underline{m}' = \underline{m} + A\underline{\sigma}_k, \quad (2.3)$$

sendo  $\underline{\sigma}_k$  o vetor característico de tamanho  $n_T$  em que a sua  $j$ -ésima componente corresponde ao número de disparos da transição  $t_j$  para que a marcação  $\underline{m}$  alcance a nova marcação  $\underline{m}'$ .

### 2.1.4 Conflito em redes de Petri

Redes de Petri, como ilustrado na figura 2.6, podem possuir lugares com mais de uma transição de saída. Essa ocorrência é chamada de conflito estrutural [5], sendo definido pela dupla  $C(p, \{t_1, t_2, \dots, t_k\})$  formada pelo lugar  $p$  e pelo conjunto  $\{t_1, t_2, \dots, t_k\}$ , das transições que formam o conflito.

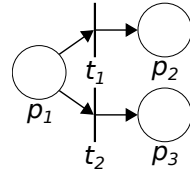


Figura 2.6: Conflito estrutural entre as transições  $t_1$  e  $t_2$  em relação ao lugar  $p_1$ .

Um conflito estrutural  $C(p, \{t_1, t_2, \dots, t_k\})$  pode impedir o disparo simultâneo de todas as transições  $\{t_1, t_2, \dots, t_k\}$  quando o número de fichas de  $p$  for menor do que a soma dos pesos dos arcos que ligam  $p$  a  $t_1, t_2, \dots, t_k$ . Essa ocorrência, chamada de conflito efetivo, afeta a dinâmica das redes por exigir alguma forma de serem escolhidos, dentre as transições envolvidas no conflito efetivo, quais devem disparar.

A figura 2.7(a) ilustra a ocorrência de um conflito entre as transições  $t_1$  e  $t_2$ , pois é impossível disparar as duas simultaneamente, enquanto a figura 2.7(b) mostra a mesma rede de Petri, porém com uma marcação que não possui um conflito efetivo pois, nessa situação, ambas as transições podem disparar simultaneamente.

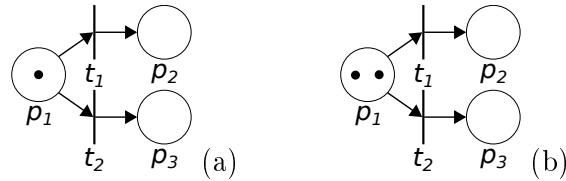


Figura 2.7: Rede de Petri com conflito efetivo entre as transição  $t_1$  e  $t_2$  (a) e sem conflito efetivo (b).

## 2.2 Rede de Petri com arcos inibidores

Com o objetivo de resolver problemas como conflitos, os arcos inibidores foram criados. Eles são representados graficamente como um arco com um determinado

peso e uma circunferência na extremidade final. O arco inibidor é definido pela função  $Inhib : (P \times T \rightarrow \mathbb{Z}_+^*)$  e sempre tem início em um lugar e fim em uma transição. Todo lugar ligado a um arco inibidor é chamado de lugar inibidor. Uma rede de Petri com arcos inibidores é definida formalmente como:

**Definição 2.4** (*Rede de Petri com arcos inibidores*) Uma rede de Petri com arcos inibidores é uma sêxtupla:

$$\mathcal{RIN} = (P, T, Pre, Post, Inhib, \underline{m}_0)$$

em que:

- $(P, T, Pre, Post, \underline{m}_0)$  é uma rede de Petri com marcação de acordo com a definição 2.2 e
- $Inhib : (P \times T) \rightarrow \mathbb{Z}_+^*$  é a função de arcos inibidores que conectam lugares a transições.

Quando uma transição possui arcos inibidores, ela só pode ser habilitada se, além de todos os seus lugares de entrada possuírem números de fichas iguais ou maiores do que os pesos dos respectivos arcos que ligam esses lugares à transição e todos os seus lugares inibidores contiverem números menores de fichas que os pesos dos seus respectivos arcos inibidores. Se essas duas condições forem satisfeitas, a transição estará habilitada e poderá disparar. A definição de uma transição habilitada para uma rede de Petri com arcos inibidores é dada a seguir.

**Definição 2.5** (*Transição habilitada em uma rede de Petri com arcos inibidores*) Seja  $I(t_j)$  o conjunto de todos os lugares de entrada e  $IH(t_j)$  o conjunto de todos os lugares inibidores da transição  $t_j \in T$ . Então  $t_j$  está habilitada se,  $\forall p_i \in I(t_j)$ , a equação (2.1) for verificada e,  $\forall p_k \in IH(t_j)$ ,  $m(p_k) < Inhib(p_k, t_j)$ .

A figura 2.8 exemplifica o funcionamento de um arco inibidor na habilitação de uma transição. Considere, inicialmente, a figura 2.8(a). Note que a transição  $t_1$  pode disparar, pois está habilitada de acordo com as regras de disparo das transições dadas na definição 2.5. Entretanto, na figura 2.8(b), a ficha no lugar  $p_2$  impede a habilitação da transição  $t_1$ .

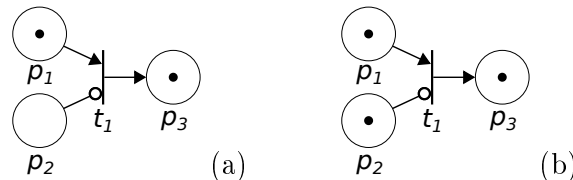


Figura 2.8: Rede de Petri com transição  $t_1$  habilitada (a) e com a mesma transição não habilitada (b).



## 2.3 Rede de Petri temporizada-T

A definição 2.2 de redes de Petri permite descrever principalmente a estrutura lógica e dinâmica de sistemas. Ela, porém, não possui algo que represente a passagem de tempo, a qual é fundamental para tornar a descrição do sistema mais realista. Por isso, surgiram as redes de Petri temporizadas.

As duas principais formas de se adicionar o conceito de tempo a uma rede de Petri é utilizando transições temporizadas ou lugares temporizados para representar intervalos de tempo associados a durações de operações ou tempo de espera entre tarefas. Neste trabalho, apenas transições temporizadas são utilizadas, que são representadas por caixas retangulares vazias, como mostrado na figura 2.9.

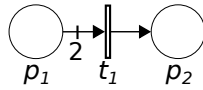


Figura 2.9: Exemplo de uma transição temporizada  $t_1$ .

Nesse contexto, uma rede de Petri temporizada-T pode ser definida da seguinte forma. Para tanto, torna-se necessário particionar o conjunto  $T$  das transições em  $T = T_i \dot{\cup} T_d$ , isto é,  $T = T_i \cup T_d$  e  $T_i \cap T_d = \emptyset$ , em que  $T_d$  denota o subconjunto de transições temporizadas, enquanto  $T_i$  expressa o subconjunto de transições imediatas.

**Definição 2.6** (*Rede de Petri temporizada-T*) Uma rede de Petri temporizada-T é uma *sêxtupla*:

$$\mathcal{RT} = (P, T, Pre, Post, Tempo, \underline{m}_0)$$

em que:

- $(P, T, Pre, Post, \underline{m}_0)$  é uma rede de Petri com marcação de acordo com a definição 2.2 e
- *Tempo*:  $(T_d) \rightarrow \mathbb{Q}_+$  é uma função que associa a cada transição  $t_j \in T_d$  um intervalo de tempo  $d_j$ .

## 2.4 Redes de Petri estocásticas generalizadas

Com o objetivo de adicionar características probabilísticas aos disparos das transições temporizadas e imediatas, torna-se necessário utilizar as chamadas redes de Petri estocásticas generalizadas (RPEG), que permitem atribuir tempos gerados estocasticamente em transições temporizadas e resolver conflitos entre transições imediatas por meio de probabilidades. Nas RPEG, as transições temporizadas têm

retardos associados diferentes de zero, enquanto as transições imediatas têm retardos nulos.

**Observação 2.1** *Normalmente, redes de Petri estocásticas generalizadas resolvem conflitos efetivos ao atribuir prioridades de disparo a cada transição envolvida no conflito [7]. Essa abordagem, entretanto, não é compatível com as redes de Petri apresentadas neste trabalho, pois os conflitos são resolvidos para cada ficha individual pertencente à marcação do lugar envolvido no conflito, isto é, para cada ficha, é escolhida, por meio de probabilidade, qual transição envolvida no conflito irá disparar. É possível, ainda, que conflitos possam ser resolvidos por meio de um agente externo, quando não são atribuídas probabilidades de disparo às transições envolvidas no conflito. Esse problema será abordado mais à frente neste trabalho.*

Os conflitos da rede são resolvidos utilizando um mecanismo de arbitragem chamado chaveamento aleatório, que consiste em atribuir a cada transição do conflito uma probabilidade de disparo. O chaveamento aleatório é formado pelo conjunto de transições imediatas habilitadas e pelas probabilidades definidas sobre elas, conhecida como distribuição de chaveamento [7]. Para tanto, torna-se necessário definir duas funções,  $\Lambda_d(\cdot)$  e  $\Lambda_i(\cdot)$  que mapeiam, respectivamente, as transições temporizadas e imediatas de uma RPEG.

Para cada transição  $t_j \in T_d$  temporizada é definida por uma função densidade de probabilidade  $fdp_{t_j}(x)$ , que gera, a partir de uma variável aleatório  $x$ , um número real não negativo entre 0 e 1. Esse número gerado é então atribuído a  $\Lambda_d(t_j)$ , que representa o tempo que  $t_j$  leva para disparar após sua habilitação.

Quando transições temporizadas estiverem envolvidas em um conflito estrutural  $C(p, T_c)$ , em que  $T_c \subseteq T_d$  e em relação a um lugar  $p$ , e se, nesse conflito, todas as transições forem temporizadas, os valores  $\Lambda_d(t_j)$  são usados para resolver o conflito. Para tanto, os correspondentes tempos de atraso de disparo das transições pertencentes a  $T_c$  são sorteados a partir de suas respectivas  $fdp_{t_j}(\cdot)$  e diferentes variáveis aleatórias. É escolhida, então, a transição  $t_j$ , que obtiver o menor valor dentre os sorteados, sendo o disparo realizado após a passagem do tempo calculado. Dessa forma, define-se  $\Lambda_d(t)$  da seguinte forma:

$$\Lambda_d(t) = \begin{cases} fdp_t(x), & \text{se } t \in T_d; \\ \text{não definido, caso contrário,} & \end{cases} \quad (2.4)$$

em que  $x$  é uma variável aleatória tal que a probabilidade  $p(x \leq d_t) = \int_0^{d_t} fdp_{t_j}(x)dx$ , sendo  $d_t$  o tempo de retardo de disparo da transição  $t_j$ .

Caso haja transições imediatas e habilitadas no conflito, a transição temporizada não dispara, pois independente do valor sorteado para o tempo de atraso de disparo, pelo menos uma transição imediata dispara primeiro.

Caso houver conflitos envolvendo transições imediatas, deverá ser escolhido, por meio de um sorteio, qual delas irá disparar instantaneamente. As probabilidades envolvidas nesse sorteio são definidas a partir de funções  $\Lambda_i(\cdot)$ , que são definidas da seguinte forma:

$$\Lambda_i(t, p) = \begin{cases} PRC(t|p), \forall t \in C(p, \{t_1, t_2, \dots, t_k\}), \text{ se } \{t_1, t_2, \dots, t_k\} \subseteq T_i, \\ \text{tal que } PRC(t|p) \in \mathbb{Q}_+, 0 \leq PRC(t|p) \leq 1 \text{ e} \\ \sum_{j=1}^k PRC(t_j|p) = 1, \text{ ou;} \\ \text{n\~{a}o definido, caso contr\~{a}rio.} \end{cases} \quad (2.5)$$

em que  $PRC(t_j|p)$  representa, para cada transiç~{a}o  $t_j \in T_i$  envolvida no conflito dado  $p$ , a probabilidade de cada transiç~{a}o ser escolhida para disparar. Para certificar-se de que uma transiç~{a}o seja escolhida para disparar no sorteio de um conflito  $C(p, \{t_1, t_2, \dots, t_k\})$  entre as transiç~{a}es imediatas, a soma de todas as probabilidades de todas as transiç~{a}es imediatas  $t_j$  do conflito em rela~{c}~{a}o a um lugar  $p$  deve ser igual a um.

Baseado nessas consideraç~{o}es, uma RPEG pode ser definida da seguinte forma.

**Definiç~{a}o 2.7** (*Rede de Petri estoc\~{a}stica generalizada*) *Uma RPEG \u00e9 uma s\u00e9tupla:*

$$\mathcal{RPGE} = (P, T, Pre, Post, \Lambda_d, \Lambda_i, \underline{m}_0)$$

em que:

- $(P, T, Pre, Post, \underline{m}_0)$  \u00e9 uma rede de Petri com marcaç~{a}o de acordo com a definiç~{a}o 2.2;
- As funç~{o}es  $\Lambda_d(T_d)$  e  $\Lambda_i(T_i, P)$  s~{a}o definidas, respectivamente, de acordo com as equa~{c}~{o}es (2.4) e (2.5).

Na figura 2.10, \u00e9 mostrada uma rede de Petri com diferentes disposiç~{o}es de transiç~{o}es imediatas e temporizadas envolvidas em conflitos. Essa rede pode ser definida como  $RPGE = (P, T, Pre, Post, \Lambda_d, \Lambda_i, \underline{m}_0)$  em que  $P, T, Pre, Post, \underline{m}_0$  s~{a}o definidos na figura 2.10. As funç~{o}es  $\Lambda_d$  s~{a}o dadas por:

$\{\Lambda_d(t_1), \Lambda_d(t_2) = Uniforme([1\ 3]), \Lambda_d(t_3) = Poisson(4), \Lambda_d(t_4) = Poisson(5)\}$ , sendo

$$\Lambda_d(t_1) = \begin{cases} 1, \text{ se } x = 5; \\ 0, \text{ caso contr\~{a}rio.} \end{cases}$$

Neste caso, significando que o atraso de disparo de  $t_1$  é constante e igual a 5.

As funções  $\Lambda_i$  da rede são definidas como:

$$\{\Lambda_i(t_5, p_4) = 0, 1; \Lambda_i(t_7, p_4) = 0, 9\}$$

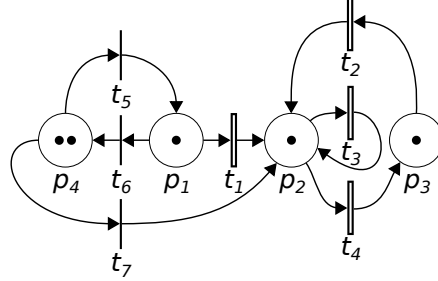


Figura 2.10: Exemplo de uma rede de Petri estocástica generalizada.

Primeiramente, nota-se que foi possível definir o tempo de atraso de disparo da transição temporizada  $t_1$  como determinístico, pois o sorteio realizado por meio da função  $\Lambda_d(t_1)$  sempre resulta no valor cinco. Portanto, apesar da rede ser uma RPEG, é possível determinar transições temporizadas equivalentes às transições temporizadas definidas em 2.6.

No lugar  $p_1$ , existe o conflito  $C(p_1, \{t_1, t_6\})$  entre a transição imediata  $t_6$  e a transição temporizada  $t_1$ . Nessa situação, a transição temporizada  $t_1$  nunca irá disparar, pois todas as fichas de  $p_1$  são usadas para disparar a transição imediata  $t_6$ .

Diferentemente de  $p_1$ , o lugar  $p_4$  está relacionado ao conflito  $C(p_4, \{t_5, t_7\})$  entre duas transições imediatas. Nesse conflito, uma das transições imediatas é escolhida, através do sorteio de um número aleatório entre 0 e 1, para disparar de acordo com as probabilidades definidas em  $\Lambda_i(t_5, p_4)$  e  $\Lambda_i(t_7, p_4)$ . Uma forma de verificar o vencedor desse sorteio é considerar que, se o número sorteado for menor do que 0,1,  $t_5$  dispara, caso contrário,  $t_7$  dispara.

Note que  $p_4$  possui duas fichas que podem realizar dois disparos diferentes. Como as colisões deste trabalho são tratadas individualmente, então, conforme mencionado na observação 2.1, são realizados dois sorteios, um para cada ficha.

O lugar  $p_2$  possui um conflito entre as transições temporizadas  $t_3$  e  $t_4$ , isto é,  $C(p_2, \{t_3, t_4\})$ , sendo ambas as transições definidas por funções densidade de probabilidade de *Poisson*. Comparando ambas e considerando que a transição temporizada do conflito que dispara é a que obtiver o menor tempo de atraso de disparo gerada pelas funções  $\Lambda_d(t_3)$  e  $\Lambda_d(t_4)$ , nota-se que  $t_3$  tem maiores chances de obter um tempo menor e, conseqüentemente, ser escolhida para disparar após a passagem de tempo sorteado. Independente das maiores chances mencionadas, ainda existe a possibilidade de  $t_4$  disparar, dependendo dos sorteios.

Apesar do lugar  $p_3$  não possuir nenhum conflito entre transições, sua transição de saída,  $t_2$  é temporizada e estocástica. Assim, o tempo que  $t_2$  leva para disparar é definido através do sorteio aleatório uniforme de um valor entre 1 e 3, de acordo com  $\Lambda_d(t_2)$ .

## 2.5 Redes de Petri estocásticas generalizadas com controles externos

Existem modelos em redes de Petri, como o que será apresentado posteriormente, que necessitam da atuação de um agente externo na rede, capaz de realizar operações complexas para determinar o disparo de algumas transições dos modelos. Para tal, a definição de RPEG deve ser estendida para permitir que um agente externo também possa resolver conflitos estruturais entre transições imediatas  $C(p, \{t_1, t_2, \dots, t_k\})$ . Essa nova rede de Petri é denominada rede de Petri estocástica generalizada com controle externo (RPCE) e é definida a seguir.

**Definição 2.8** (*Rede de Petri estocástica generalizada com controle externo*) Uma RPCE é uma óctupla:

$$\mathcal{RPCE} = (P, T, Pre, Post, \Lambda_d, \Lambda_i, Esc, \underline{m}_0)$$

em que:

- $(P, T, Pre, Post, \Lambda_d, \Lambda_i, \underline{m}_0)$  é uma RPEG de acordo com a definição 2.7 e
- $Esc$  é definida, para o conflito  $C(p, T_c)$  e  $T_c \subseteq T_i$ , da seguinte forma:

$$Esc(t_j, p) = \begin{cases} 1, & \text{se o agente externo decidir que a transição } t_j \in T_c \\ & \text{da função é escolhida para disparar;} \\ 0, & \text{caso contrário.} \end{cases}$$

Semelhante à função  $\Lambda_i$ , quando  $Esc(t_c, p) = 1$ , ocorre a escolha, dentre as transições imediatas de um conflito  $C(p, \{t_1, t_2, \dots, t_k\})$ , de uma transição  $t_c$  para disparar. Porém, essa decisão não necessariamente precisa ser probabilística, pois o agente externo pode decidir qual transição irá disparar por diferentes formas; por exemplo, a partir da solução de um problema de otimização.

**Observação 2.2** *Existem casos degenerados de conflitos, isto é, um lugar  $p_i$  conectado a apenas uma transição  $t_j$  pode ser considerado um conflito  $C(p_i, \{t_j\})$ . Esse caso não é importante para a solução de conflitos da RPEG, porém, em RPCEs, os conflitos degenerados tornam o agente externo capaz de controlar os disparos de transições que não necessariamente estejam em conflito com outras transições,*

*permitindo, por exemplo, que o agente obrigue uma transição a só disparar em momentos ou condições específicas.*

## 2.6 Algoritmos de busca por força bruta

Força bruta é uma das estratégias mais simples para resolver problemas computacionais [8]. Essa simplicidade é justificada pela forma como a estratégia analisa todas as possíveis respostas de um problema para escolher, de acordo com um conjunto de regras, uma delas para ser a melhor resposta do problema.

Considere, por exemplo, um problema em que seja necessário encontrar o menor número entre o conjunto de números desordenados {10 4 19 11 14 18 5 13 15 8}. Uma solução simples para esse problema é considerar, inicialmente, o primeiro elemento 10 como resposta e, em seguida, compará-lo com o segundo elemento 4. Como 4 é menor que 10, o elemento considerado como resposta passa a ser 4. O mesmo processo é feito com o elemento seguinte, 19, que é maior que 4; logo 4 continua sendo a resposta. O processo é então repetido com todos os elementos do conjunto e, após nove comparações, é possível concluir que o menor número do conjunto é 4.

Algoritmos, como o apresentado no exemplo acima, são chamados de busca por força bruta. Eles verificam todas as possibilidades de respostas para definir a resposta definitiva. Apesar de simples, o algoritmo não é eficiente, pois alguns problemas podem apresentar uma enorme quantidade de possibilidades de respostas que são verificadas uma a uma pelo algoritmo. Se os números do conjunto do exemplo fossem as distâncias dos possíveis caminhos que um veículo poderia percorrer para ir da cidade do Rio de Janeiro para São Paulo, por exemplo, haveria uma grande variedade de opções, o que tornaria o cálculo do trajeto com a menor distância muito demorado.

Apesar da complexidade computacional de um algoritmo de busca por força bruta variar de acordo com o problema, ela é definida pela quantidade de comparações que devem ser feitas entre todas as possíveis soluções dos problemas. Considere, por exemplo, o problema do caixeiro viajante, cujo objetivo é encontrar o menor caminho que precisa ser feito para se visitar  $n$  localizações diferentes. Se o número de localizações for 5, existirão 12 possíveis trajetos. Generalizando para  $n$  localizações, o número de possíveis trajetos será  $(n-1)!/2$ ; portanto, a complexidade do algoritmo é  $O(n!)$ , ou seja, a complexidade da solução cresce de forma não polinomial, pois para  $n = 15$ , por exemplo, a complexidade aumentaria para aproximadamente 43 trilhões de comparações. Portanto, o algoritmo é incapaz de encontrar respostas de problemas de grande porte cuja quantidade de soluções aumenta de forma não polinomial.

## Capítulo 3

# Sistema público de compartilhamento de bicicletas

Os SPCBs possuem como objetivo disponibilizar a um indivíduo a capacidade de se transportar entre lugares de sistemas utilizando uma bicicleta retirada do lugar de partida e devolvida no destino, trazendo várias melhorias de mobilidade urbana. Apesar do objetivo ser simples, a falta de preocupação com frequências e horários de uso dos locais, das rotas que os usuários trafegam, das quantidades de bicicletas nas estações, entre outros fatores, podem diminuir os benefícios esperados. Por isso, a modelagem, análise e avaliação do desempenho de SPCBs são fundamentais, tanto para propiciar uma implementação eficiente de um SPCB como para avaliar a necessidade de novas medidas o seu aprimoramento.

Neste capítulo, inicialmente, na seção 3.1, será descrito com mais detalhes as características de um SPCB. Em seguida, será apresentado, na seção 3.2, o modelo em redes de Petri proposto por Antunes et al. [1, 2]. Finalmente, será feita, na seção 3.3, uma análise crítica sobre o modelo proposto em [1, 2], abordando os problemas que motivaram a realização deste projeto.

### 3.1 Descrição do sistema

Geralmente, os SPCBs são implementados visando o aumento do uso urbano da bicicleta, redução do congestionamento, melhoria da qualidade do ar e o aumento da oferta de opções de transporte não motorizados. Além disso, os SPCBs adicionam uma nova alternativa de transporte público à região.

Cada sistema é composto por estações distribuídas geograficamente em uma região, cada uma com um número fixo de vagas de bicicletas que são preenchidas e desocupadas durante sua operação. Além das estações, um SPCB, como um todo, possui um número finito de bicicletas que inicialmente são distribuídas entre as

estações. Durante o funcionamento do sistema, os usuários retiram bicicletas das estações de partidas que possuam pelo menos uma bicicleta disponível, e as depositam nas suas estações de destino que devem possuir pelo menos uma vaga livre para ser ocupada pela bicicleta. Ambas as ações alteram o número de bicicletas armazenadas em cada estação durante o dia, fato que pode ser preocupante dependendo da quantidade de alterações.

Devido ao seu funcionamento, um SPCB permite que cada usuário trafegue entre locais de uma região sem precisar se responsabilizar pela bicicleta antes e depois da viagem ou arcar com os custos de se ter uma bicicleta. Como o sistema deve atender a diferentes tipos de trajetos entre as estações, o usuário tem a liberdade de retirar uma bicicleta de qualquer estação e viajar para outra estação à sua escolha.

Para que um SPCB possa atender às necessidades de mobilidade dos seus usuários em uma dada região com maior êxito, seus parâmetros, tais como quantidades, a localização e o número de vagas de bicicletas de cada estação, devem ser bem planejados. Para isso, são necessários estudos dos principais deslocamentos feitos na região e de outros fatores que influenciam na implementação do SPCB.

Além dos estudos preliminares, é possível obter informações do sistema durante o seu funcionamento. Através de comunicações entre as estações e uma central, é possível consultar e registrar a quantidade de bicicletas em cada estação e quais usuários retiraram e devolveram bicicletas às estações em tempo real. Esse tipo de informação torna possível o planejamento de uma manutenção da estrutura do sistema para demandas mais atuais.

Apesar do bom planejamento e manutenção da estrutura do sistema ajudarem no seu funcionamento, o fluxo de usuários entre as estações não é algo uniforme ao longo do dia. Por isso, dois problemas podem surgir: muitos usuários podem desejar sair de uma mesma estação simultaneamente, retirando bicicletas até que ela fique sem bicicletas para outros clientes; ou muitos usuários podem viajar para o mesmo destino, lotando a estação e impedindo que outros ciclistas depositem nela as suas bicicletas.

Para evitar que os números de bicicletas nas estações sejam alterados apenas pelos usuários, alguns sistemas adotam a utilização de veículos, motorizados ou não, para realizar a redistribuição de bicicletas entre as estações. Um veículo de redistribuição é capaz de armazenar uma certa quantidade de bicicletas e de se locomover entre as estações e um centro de redistribuição, onde ele pode estacionar. Sua principal função é, através de rondas entre as estações executadas durante o dia, retirar bicicletas das estações que estão cheias de bicicletas e repor bicicletas nos locais que estão vazios. A figura 3.1 ilustra o fluxo de bicicletas realizado pelos usuários e pelo veículo de redistribuição em um sistema de três estações.



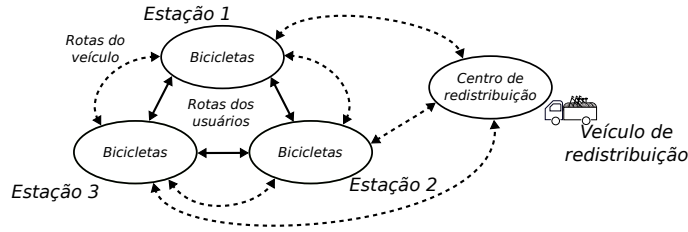


Figura 3.1: Fluxo de bicicletas em um SPCB de três estações com um veículo de redistribuição.

Um balanceamento de um SPCB bem sucedido é essencial para a viabilidade de um sistema com alta demanda, pois isso evita que usuários encontrem estações cheias ou vazias, o que gera insatisfação com o sistema e possíveis perdas de clientes. Por esse motivo, cada balanceamento deve ocorrer baseando-se em alguma operação planejada para otimizar as quantidades de bicicletas nas estações ao longo do funcionamento do sistema. Essa operação pode se basear em vários fatores, como, por exemplo, as rotas entre estações mais comuns feitas pelos usuários em momentos diferentes de cada dia da semana, o que permite fazer uma operação de balanceamento antes da ocorrência de uma alta demanda que cause um grande desbalanceamento, ou também a quantidade atual de bicicletas em cada estação, priorizando estações mais desbalanceadas durante os balanceamentos.

Como a operação de balanceamento depende de vários fatores complexos, diversos estudos de algoritmos de balanceamento foram propostos [9–11]. Na seção seguinte, será considerada a estratégia de balanceamento e o modelo propostos em [1, 2], tendo em vista que este trabalho visa continuar aquele apresentado em [1, 2].

### 3.2 Modelo de Antunes et al. implementando uma estratégia de balanceamento cíclica

O modelo proposto em [1, 2] permite simular e avaliar, através de uma rede de Petri estocástica generalizada com arcos inibidores, o funcionamento da operação de um SPCB com  $N$  estações, agrupadas em um conjunto  $S = \{S_1, S_2, \dots, S_N\}$ , e um veículo de redistribuição, que é capaz de remanejar um certo número de bicicletas em cada estação ao visitá-las em uma ordem cíclica. Cada estação  $S_i$  é capaz de armazenar uma quantidade  $C_i$  de bicicletas, enquanto o veículo de redistribuição possui a capacidade  $C_v$ . A figura 3.2 mostra a rede de Petri do modelo da estação  $S_1$  de um SPCB de três estações. Os correspondentes lugares, transições e parâmetros são definidos, respectivamente, pelas tabelas 3.2, 3.3, 3.1 [1, 2].

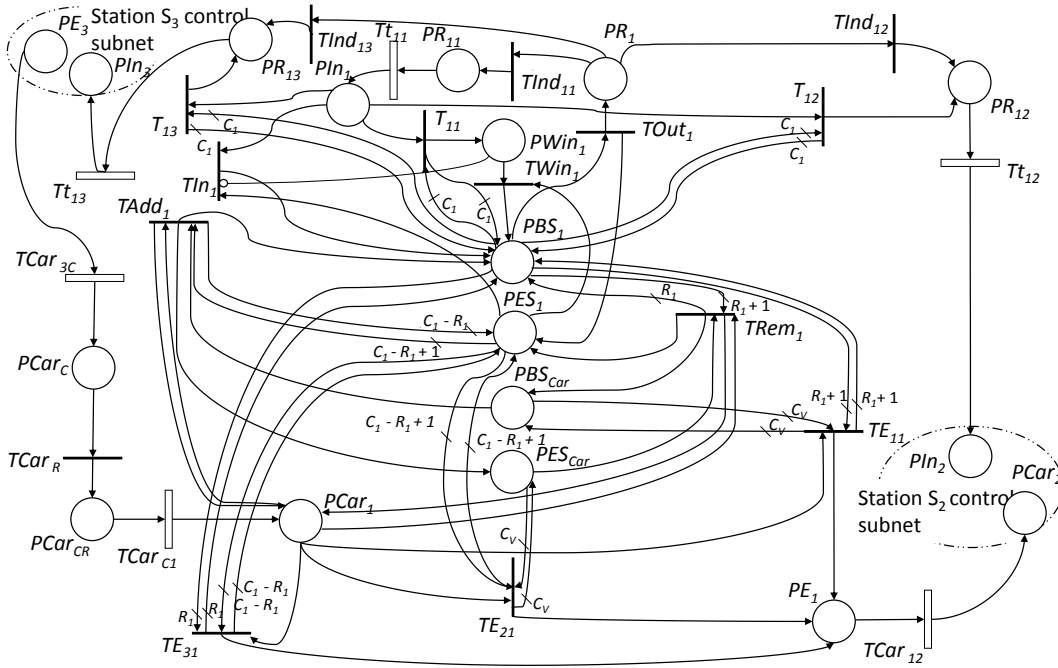


Figura 3.2: Rede de Petri estocástica generalizada com arcos inibidores da estação  $S_1$  de um SPCB de três estações proposto em [1, 2]

Tabela 3.1: Interpretações dos parâmetros da rede de Petri proposta em [1, 2].

Parâmetro	Interpretação
$R_i$	Número de reordenação da estação $S_i$ . Parâmetro no qual o veículo de redistribuição se baseia para tomar a decisão de adicionar ou remover uma bicicleta de acordo com o número de bicicletas disponíveis em $S_i$ .
$C_i$	Capacidade de cada estação $S_i$ . $C_i$ corresponde ao número máximo de suportes para bicicletas em $S_i$ .
$C_v$	Capacidade do veículo de redistribuição $S_i$ . $C_v$ corresponde ao número máximo de suportes para bicicletas no veículo de redistribuição.

Tabela 3.2: Interpretações dos lugares da rede de Petri proposta em [1, 2].

<b>Lugar</b>	<b>Interpretação</b>
$PR_i$	Representa as bicicletas que saíram da estação $S_i$ , ou seja, que estão fora das estações.
$PR_{ij}$	Representa as bicicletas que saíram da estação $S_i$ com destino à estação $S_j$ .
$PIn_i$	Representa as bicicletas dos usuários que desejam estacionar na estação $S_i$ .
$PWin_i$	Indica as bicicletas dos usuários que estão aguardando a próxima vaga disponível em $S_i$ , quando $S_i$ estiver lotada.
$PBS_i$	Representa os suportes ocupados por bicicletas da estação $S_i$ . A sua marcação $m(PBS_i)$ corresponde ao número de bicicletas disponíveis na estação $S_i$ .
$PES_i$	Representa os suportes vazios da estação $S_i$ . A sua marcação $m(PES_i)$ corresponde ao número de vagas disponíveis na estação $S_i$ .
$PBSCar$	Representa os suportes ocupados por bicicletas no veículo de redistribuição. A sua marcação $m(PBSCar)$ corresponde ao número de bicicletas no veículo.
$PESCar$	Representa os suportes vazios no veículo de redistribuição. A marcação $m(PESCar)$ corresponde ao número de lugares disponíveis no veículo.
$PCar_i$	Informa se o veículo de redistribuição está na estação $S_i$ para realizar o balanceamento, que é indicado quando $m(PCar_i) = 1$ .
$PE_i$	Indica o término do balanceamento na estação $S_i$ pelo veículo de redistribuição. $m(PE_i) = 1$ significa que o controle da estação foi realizado e que o veículo está liberado para partir para a próxima estação.
$PCar_C$	Informa se o veículo de redistribuição está no centro de redistribuição esperando para realizar o balanceamento, correspondente a $m(PCar_C) = 1$ .
$PCar_{CR}$	Informa se o veículo de redistribuição está no centro de redistribuição e foi liberado para dar início à ronda.

Tabela 3.3: Interpretações das transições da rede de Petri proposta em [1, 2].

<b>Transição</b>	<b>Interpretação</b>
$TOut_i$	Uma bicicleta é retirada da estação $S_i$ pelo usuário se houver em $S_i$ pelo menos uma bicicleta disponível.
$TInd_{ij}$	Uma bicicleta sai da estação $S_i$ e escolhe a estação $S_j$ como destino.
$Tt_{ij}$	Transição temporizada que representa o deslocamento de uma bicicleta da estação $S_i$ para a estação $S_j$ .
$T_{ij}$	Modela quando um usuário chega à estação $S_i$ e a encontra lotada, decidindo ir para a estação $S_j$ ou esperar nela mesma até a próxima vaga ser liberada (se $i = j$ ).
$TIn_i$	Uma bicicleta é adicionada à estação $S_i$ pelo usuário se houver pelo menos um suporte vazio em $S_i$ e não tiver ninguém esperando na fila por uma vaga ( $m(PWin_i) = 0$ ).
$TWin_i$	Uma bicicleta é adicionada à estação $S_i$ pelo usuário após ele aguardar por um suporte vazio em $S_i$ .
$TRem_i$	O veículo remove uma bicicleta da estação $S_i$ se o número de bicicletas disponíveis na estação for maior que o número de reordenação $R_i$ ( $m(PBS_i) > R_i$ ) e houver pelo menos uma vaga de bicicleta disponível no veículo ( $m(PESCar) > 0$ ).
$TAdd_i$	O veículo adiciona uma bicicleta na estação $S_i$ se o número de bicicletas disponíveis na estação for menor que o número de reordenação $R_i$ ( $m(PES_i) > C_i - R_i$ ) e houver pelo menos uma bicicleta disponível no veículo ( $m(PBSCar) > 0$ ).
$TE_{1i}$	Encerra o balanceamento se o número de bicicletas disponíveis na estação $S_i$ é maior que o número de reordenação $R_i$ , mas não há suportes livres no veículo de redistribuição.
$TE_{2i}$	Encerra o balanceamento se o número de bicicletas disponíveis na estação $S_i$ é menor que o número de reordenação $R_i$ , mas não há bicicletas disponíveis no veículo de redistribuição.
$TE_{3i}$	Encerra o balanceamento se o número de bicicletas disponíveis na estação $S_i$ é igual ao número de reordenação $R_i$ .
$TCar_R$	Dá início à ronda do veículo de redistribuição.
$TCar_{C1}$	Transição temporizada. Modela o deslocamento do veículo de redistribuição do centro de redistribuição para a estação $S_1$ .
$TCar_{i(i+1)}$	Transição temporizada. Modela o deslocamento do veículo de redistribuição da estação $S_i$ para a próxima estação $S_{i+1}$ ( $i \in \{1, 2, \dots, N - 1\}$ ).
$TCar_{NC}$	Transição temporizada. Modela o deslocamento do veículo de redistribuição da última estação $S_i$ para o centro de redistribuição.

Sendo  $N$  o número total de estações do SPCB, é possível calcular o número de lugares ( $n_P$ ) e transições ( $n_T$ ) da rede, em função de  $N$ , qual seja:

$$n_P = N^2 + 7N + 4 \text{ lugares } e \ n_T = 3N^2 + 9N + 2 \text{ transições}$$

Como a rede simboliza diferentes funcionalidades interligadas de um SPCB com um veículo de redistribuição, ela pode ser dividida em três sub-redes:

1. Sub-rede do fluxo de bicicletas, que representa a movimentação das bicicletas entre as estações do sistema causadas pelos usuários que estão alugando as bicicletas. O usuário pode retirar uma bicicleta de uma estação que não esteja vazia e devolvê-la a qualquer estação que não esteja cheia, inclusive à estação de onde a bicicleta foi retirada.
2. Sub-rede do controle de estações, que representa os componentes do modelo envolvidos na reposição/retirada de bicicletas em cada estação  $S_i$  realizada pelo veículo de redistribuição.
3. Sub-rede do circuito de redistribuição, onde é representado o circuito realizado pelo veículo de redistribuição partindo do centro de redistribuição e visitando todas as estações do sistema para realizar o balanceamento.

### 3.2.1 Sub-rede do fluxo de bicicletas

A sub-rede do fluxo de bicicletas modela a atuação dos usuários no sistema, permitindo diferentes implementações de como os usuários transitam entre as estações, tanto na escolha das estações de destino quanto os tempos que levam para transitarem entre as estações através de funções estocásticas. Para simplificar a explicação da sub-rede, ela é dividida para cada estação. Assim, uma sub-rede da estação  $S_i$  utiliza os lugares  $PBS_i$ ,  $PES_i$ ,  $PR_i$ ,  $PR_{ij}$ ,  $Pin_i$ ,  $PWin_i$  e as transições  $TOut_i$ ,  $TInd_{ij}$ ,  $Tt_{ij}$ ,  $TIn_i$ ,  $T_{ij}$ ,  $TWin_i$ , em que  $j \in \{1, 2, \dots, N\}$ .

Em um SPCB, cada estação  $S_i$  funciona como um estacionamento de bicicletas, contendo um número fixo de vagas que são ocupadas e desocupadas por bicicletas ao longo do funcionamento do sistema. Para descrever esse comportamento dinâmico, o modelo utiliza os lugares  $PBS_i$  e  $PES_i$  para representarem, respectivamente, o número de bicicletas que ocupam a estação e o número de vagas disponíveis na estação. A soma dos dois lugares resulta na capacidade da estação de armazenar bicicletas, representada pelo parâmetro  $C_i$ . Assim, para todas as marcações do modelo,  $m(PBS_i) + m(PES_i) = C_i$ .

Se um usuário deseja alugar uma bicicleta da estação  $S_i$  e a estação possui pelo menos uma bicicleta disponível, o que é verificável através da desigualdade

$m(PBS_i) \geq 1$ , ele retira uma bicicleta da estação. Essa ação é representada pelo disparo da transição  $TOut_i$ , que remove uma ficha de  $PBS_i$  e adiciona uma ficha ao lugar  $PES_i$  e outra ficha ao lugar  $PR_i$ , indicando que o usuário deve escolher para qual estação ele deseja ir. Para prever o comportamento dos usuários do sistema, é utilizada uma função probabilística para fazer a escolha de uma estação  $S_j$  como destino, decisão que, na simulação, é feita através do disparo da transição  $TInd_{ij}$ , que move a ficha para o lugar  $PR_{ij}$ .

Após a passagem de um intervalo tempo definido de forma probabilística, que representa o tempo que o usuário leva para se deslocar da estação  $S_i$  para a estação  $S_j$ , a transição  $Tt_{ij}$  dispara, deslocando a ficha que representa esse usuário para o lugar  $PIn_j$ , que é o lugar da estação  $S_j$  que recebe todas as chegadas de bicicletas na estação. No lugar  $PIn_j$  é verificado se existe alguma ficha em  $PES_j$  e se não existem fichas no lugar  $PWin_j$ , verificação esta feita pelo arco inibidor do modelo. Essas verificações representam, respectivamente, se existe alguma vaga para a bicicleta na estação e se não há nenhum outro usuário esperando por vagas na estação. Caso ambas as verificações sejam válidas,  $TIn_j$  dispara, representando que a bicicleta foi depositada na estação  $S_j$ .

Caso a transição  $TIn_j$  não esteja habilitada enquanto  $m(PIn_j \geq 1)$  e  $m(PBS_j) = C_j$ , conclui-se que a estação  $S_j$  está cheia. Nessa situação, o usuário deve escolher entre aguardar uma vaga na estação, ação representada pelo disparo da transição  $T_{jj}$ , ou escolher uma outra estação  $S_k$  como destino, representada pelo disparo da transição  $T_{jk}$ . Para que não haja conflito entre essas transições e a transição  $TIn_j$ , elas verificam se a estação está cheia utilizando  $PBS_j$  como lugar de entrada e saída com pesos iguais a  $C_j$ , garantindo que elas só estejam habilitadas enquanto a estação estiver cheia. A escolha entre as transições  $T_{jk}$  também é feita durante a simulação utilizando uma função probabilística, semelhante à escolha feita para  $TInd_{ij}$ , porém com diferentes probabilidades.

Caso  $T_{jk}$  tenha disparado, a ficha se move de  $PIn_j$  para  $PR_{jk}$ , lugar que representa que o usuário escolheu a estação  $S_k$  como novo destino. Caso  $T_{jj}$  tenha disparado, a ficha se move de  $PIn_j$  para  $PWin_j$ , lugar que representa que o usuário está aguardando uma vaga de bicicleta na estação. Quando uma bicicleta é retirada da estação  $S_j$ ,  $PES_j$  ganha uma ficha, o que permite o disparo de  $TWin_j$ , que retira uma ficha de  $PWin_j$  e uma ficha de  $PES_j$  a adiciona uma ficha em  $PBS_j$ , simulando que o usuário que estava esperando por uma vaga na estação conseguiu devolver a bicicleta.

### 3.2.2 Sub-rede do controle de estações

Na sub-rede do controle de estações, é descrito o funcionamento do balanceamento de bicicletas das estações pelo veículo de redistribuição. Para descrever esse balanceamento, a sub-rede possui lugares relativos ao veículo, que são  $PBSCar$  e  $PESCar$ , e lugares e transições de cada estação, que estabelecem uma relação entre cada uma delas e o veículo para que possa acontecer a troca de bicicletas. Os lugares da sub-rede relativos a cada estação  $S_i$  são  $PBS_i$ ,  $PES_i$ ,  $PCar_i$  e  $PE_i$ , e as transições são  $TAdd_i$ ,  $TRem_i$ ,  $TE_{1i}$ ,  $TE_{2i}$  e  $TE_{2i}$ .

Assim como as estações, o veículo possui um número fixo de vagas de bicicletas que podem ser ocupadas e desocupadas durante o seu funcionamento. Como o comportamento de armazenamento de bicicletas é semelhante entre as estações e o veículo de redistribuição, o modelo utiliza os lugares  $PBSCar$  e  $PESCar$  para representarem, respectivamente, a quantidade de bicicletas que ocupam o veículo de redistribuição e as vagas disponíveis no veículo. Além disso, a soma das marcações desses dois lugares deve ser igual à capacidade total de armazenamento de bicicletas do veículo  $C_v$ . Por isso, em todas as marcações do modelo,  $m(PBSCar) + m(PESCar) = C_v$ . Apesar da semelhança com as estações, deve-se lembrar que o modelo é feito considerando-se apenas um veículo de redistribuição; portanto, independentemente da quantidade de estações, o modelo possui apenas um lugar  $PBSCar$  e um lugar  $PESCar$ .

Como o veículo é capaz de visitar diferentes estações, é necessária alguma forma de se verificar em qual estação o veículo está efetuando o balanceamento. O lugar utilizado para fazer esse tipo de verificação é o  $PCar_i$ , que indica que o veículo está realizando o balanceamento da estação  $S_i$  quando  $m(PCar_i) = 1$ . Quando o veículo está executando o balanceamento da estação  $S_i$ , ele é capaz de executar dois tipos de operações:

1. Operação de adição de bicicletas na estação, que é realizada pela transição  $TAdd_i$  quando há a necessidade de se adicionar bicicletas à estação. A habilitação e disparo da transição ocorrem quando as seguintes condições forem satisfeitas: (i) o veículo está na estação  $S_i$ , ou seja,  $m(PCar_i) = 1$ ; (ii) a quantidade de bicicletas na estação é menor que o seu número de reordenação  $R_i$ , o que pode ser verificado pela inequação  $m(PES_i) \geq C_i - R_i + 1$ ; (iii) se o veículo de redistribuição possuir pelo menos uma bicicleta, ou seja,  $m(PBSCar) \geq 1$ . No disparo da transição  $TAdd_i$ , a estação recebe uma bicicleta ao mesmo tempo em que o veículo perde uma bicicleta.
2. Operação de remoção de bicicletas na estação, que é realizada pela transição  $TRem_i$  quando há a necessidade de se remover bicicletas da estação  $S_i$  por ela estar com excesso de bicicletas, o que pode ser verificado pela condição

$m(PBS_i) \geq R_i + 1$ . De forma semelhante à operação de adição, é verificado pela transição se o veículo está na estação  $S_i$ , porém, ao contrario do que é feito na outra operação, é verificado se o veículo possui pelo menos uma vaga de bicicleta disponível, ou seja, se  $m(PESCar) \geq 1$ .

As operações mencionadas anteriormente são feitas com o objetivo de igualar a quantidades de bicicletas das estações com seus respectivos números de reordenação, ou seja, para cada estação  $S_i$ , tornar  $m(PBS_i) = R_i$ . Dependendo da quantidade de bicicletas no veículo, porém, o veículo pode não ser capaz de balancear completamente uma estação que está tentando balancear por não possuir bicicletas para depositar na estação ou por estar totalmente cheio de bicicletas.

Dada essa adversidade, o modelo considera o balanceamento da estação  $S_i$  pelo veículo de redistribuição como finalizado na ocorrência de uma das seguintes condições:

1. Quando o veículo de redistribuição está cheio, o que pode ser representado por  $m(PBSCar) = C_v$ , e a quantidade de bicicletas na estação  $S_i$  é maior que o número de reordenação, ou seja,  $m(PBS_i) \geq R_i + 1$ . Isso significa que o veículo não é capaz de reduzir a quantidade de bicicletas na estação e, por isso, o balanceamento da estação precisa ser finalizado, o que é modelado pelo disparo da transição  $TE_{1i}$ , que desloca a ficha de  $PCar_i$  para  $PE_i$ , o que indica que o veículo terminou o balanceamento e está pronto para sair da estação atual;
2. Ao contrario da forma de finalizar o balanceamento apresentada anteriormente, é possível também que o veículo de redistribuição esteja vazio, o que é verificado pela equação  $m(PESCar) = C_v$ , e que a quantidade de bicicletas na estação  $S_i$  seja menor do que o número de reordenação, representado pela inequação  $m(PES_i) \geq C_i - R_i + 1$ . Nessa situação, a transição  $TE_{2i}$  é disparada, deslocando a ficha de  $PCar_i$  para  $PE_i$ ;
3. Finalmente, é verificado se o veículo foi capaz de balancear a estação. Para fazer isso, é verificado se o número de bicicletas na estação é igual ao número de reordenação através das equações  $m(PBS_i) = R_i$  e  $m(PES_i) = C_i - R_i$ . Caso essa verificação seja positiva, a transição  $TE_{3i}$  é disparada, fazendo o mesmo deslocamento de ficha que os itens anteriores.

### 3.2.3 Sub-rede do circuitos de redistribuição

Para o veículo balancear as diferentes estações de um SPCB, ele realiza rondas em determinado momentos do funcionamento do sistema em que o veículo sai do centro



de redistribuição para visitar as estações do sistema. No modelo proposto em [1, 2], em cada ronda, o veículo sai do centro de redistribuição e visita todas as estações do sistema através de um circuito predeterminado no modelo. A sub-rede de circuitos de redistribuição tem a função de descrever esse circuito, representando como o veículo se move entre as estações e o centro de redistribuição e simulando o tempo que ele leva para se deslocar de um local para outro. Para fazer essa descrição, a sub-rede utiliza os lugares  $PCar_N$ ,  $PE_N$ ,  $PCar_C$ ,  $PCar_{CR}$  e as transições  $TE_{1N}$ ,  $TE_{2N}$ ,  $TE_{3N}$ ,  $TCar_{NC}$ ,  $TCar_R$ ,  $TCar_{C1}$ , relativos ao centro de redistribuição, a primeira estação  $S_1$  e última estação  $S_N$  do circuito da ronda; os lugares  $PCar_i$ ,  $PE_i$  e as transições  $TE_{1i}$ ,  $TE_{2i}$ ,  $TE_{3i}$ ,  $TCar_{i(i+1)}$  para as estações  $S_i$ , em que  $i \in \{1, 2, \dots, N-1\}$ .

Antes da ronda, quando a marcação  $m(PCar_C) = 1$ , o veículo de redistribuição está estacionado no centro de redistribuição aguardando pela próxima ronda, cuja ativação é representada pelo disparo da transição  $TCar_R$ . O disparo da transição  $TCar_R$  adiciona uma ficha à  $PCar_{CR}$ , representado que o veículo está se deslocando para a primeira estação do sistema a ser visitada (estação  $S_1$ ). A chegada do veículo à estação ocorre com o disparo da transição temporizada e estocástica  $TCar_{C1}$ , que é disparada após a passagem de um intervalo de tempo definido estocasticamente.

A chegada do veículo à primeira estação do circuito é representada pela marcação  $m(PCar_1) = 1$ , que foi feita pelo disparo de  $TCar_{C1}$ . Durante a marcação, é efetuado o balanceamento da estação  $S_1$  através da sub-rede do controle de estações. Quando o balanceamento termina, uma das transições  $TE_{11}$ ,  $TE_{21}$  ou  $TE_{31}$  disparam, adicionando uma ficha ao lugar  $PE_1$ , indicando que o balanceamento na estação terminou e que o veículo deve se mover para a próxima estação  $S_2$ .

O movimento do veículo da estação  $S_1$  até  $S_2$  também é feita por uma transição temporizada que utiliza uma função estocástica para determinar seu intervalo de tempo. Os disparos das transições e trocas de marcações da sub-rede de circuitos de redistribuição das estações  $S_2$ ,  $S_3$ , ...  $S_{N-1}$  são iguais a da estação  $S_1$ .

Quando o veículo chega na estação  $S_N$ , são realizados os mesmos procedimentos até a adição de uma ficha ao lugar  $PE_N$ . Porém, como o veículo deve retornar ao centro de redistribuição, é disparada a transição temporizada  $TCar_{NC}$ , que representa o deslocamento do veículo da estação  $S_N$  para o centro de redistribuição, quando é novamente adicionada uma ficha ao lugar  $PCar_C$ , indicando o fim da ronda de balanceamento.

### 3.3 Análise crítica sobre o modelo da Antunes et al.

O modelo proposto em [1, 2] apresenta uma forma de implementar as principais características de um SPCB com um veículo de redistribuição em redes de Petri generalizadas estocásticas com arco inibidores. Apesar do modelo ter sido simulado

com sucesso no trabalho, é possível melhorá-lo.

Para diminuir a complexidade da rede de Petri do modelo, deve-se remover os arcos inibidores, pois eles tornam o modelo mais complexo, exigindo um processo extra na verificação da habilitação da transição  $TIn_i$  para uma estação  $S_i$ . Essa remoção é possível pois a rede de Petri é limitada, o que significa que nenhuma marcação de cada lugar lugar excede um número finito que, no caso do modelo, é o número total de bicicletas no sistema, denotado por  $N_B$ .

Outra parte da rede que pode ser melhorada é a estratégia de balanceamento. No modelo, foi utilizado uma estratégia de um balanceamento em que o veículo realiza um simples trajeto cíclico e predeterminado durante suas rondas. Essa estratégia, portanto, não se adapta às condições atuais do SPCB e obriga o veículo a visitar todas as estações, inclusive estações que não precisam de balanceamento no momento da visita do veículo.

# Capítulo 4

## Modelo utilizando rede de Petri estocástica generalizada com controle externo

Uma forma de tornar possível a melhoria da estratégia de balanceamento do modelo apresentado na seção anterior é alterar a rede de tal forma que seja possível implementar diferentes estratégias de balanceamento através de algoritmos externos. Para isso, será utilizada a definição de redes de Petri estocásticas generalizadas com controles externos para a criação de um novo modelo a partir daquele proposto em [1, 2].

Neste capítulo, será primeiramente apresentado na seção 4.1 o novo modelo proposto neste trabalho, suas sub-redes e suas principais diferenças em relação ao modelo apresentado na seção 3.2. Em seguida, na seção 4.2 será obtida a matriz de incidência do modelo proposto para um SPCB com  $N$  estações. Finalmente, a matriz de incidência obtida será utilizada para calcular a ordem da rede de Petri do modelo na seção 4.3.

### 4.1 Modelo proposto

Assim como o modelo apresentado na seção 3.2, o modelo aqui proposto pode ser utilizado para modelar um SPCB com  $N$  estações. Os parâmetros do modelo, os lugares e suas transições são descritos, respectivamente, nas tabelas 4.1, 4.2 e 4.3. A figura 4.1 mostra a rede de Petri associada à estação  $S_1$  de um SPCB com 3 estações.

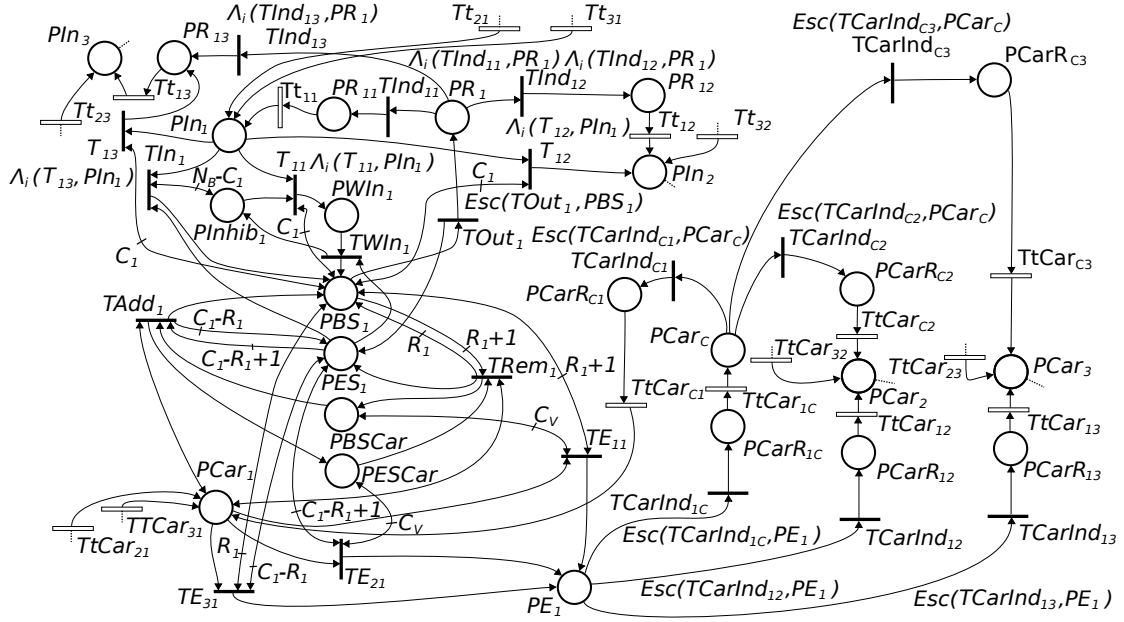


Figura 4.1: Rede de Petri estocástica generalizada com controle externo da estação  $S_1$  de um SPCB com três estações.

Tabela 4.1: Interpretações dos parâmetros da rede de Petri proposta.

Parâmetro	Interpretação
$R_i$	Número de reordenação de cada estação $S_i$ . Parâmetro no qual o veículo de redistribuição se baseia para tomar a decisão de adicionar ou remover uma bicicleta.
$C_i$	Capacidade da estação $S_i$ . Corresponde ao número máximo de suportes para bicicletas em $S_i$ .
$C_v$	Capacidade do veículo de redistribuição $S_i$ . Corresponde ao número máximo de suportes para bicicletas no veículo de redistribuição.
$N_B$	Número total de bicicletas no SPCB.

Tabela 4.2: Interpretações dos lugares da rede de Petri proposta.

<b>Lugar</b>	<b>Interpretação</b>
$PIn_i$	Representa as bicicletas dos usuários que desejam estacionar na estação $S_i$ .
$PInhib_i$	Lugar utilizado para indicar que o usuário que chegou à estação $S_i$ com uma bicicleta e precisa aguardar o depósito de bicicletas de outros usuários que estavam aguardando por vagas antes da sua chegada. Se $m(PInhib_i) = N_B - C_i$ para uma estação $S_i$ , então o usuário deposita sua bicicleta sem esperar, caso contrário, ele deve aguardar.
$PE_i$	Indica o término do balanceamento na estação $S_i$ pelo veículo de redistribuição. $m(PE_i) = 1$ significa que o balanceamento da estação foi realizado e que o sistema pode definir a próxima estação que o veículo visitará.
$PES_i$	Representa os suportes vazios da estação $S_i$ . A sua marcação $m(PES_i)$ corresponde ao número de vagas disponíveis na estação $S_i$ .
$PBS_i$	Representa os suportes ocupados da estação $S_i$ , sendo $m(PBS_i)$ o número de bicicletas disponíveis na estação $S_i$ .
$PR_i$	Representa as bicicletas que saíram da estação $S_i$ , ou seja, que estão fora das estações.
$PWIn_i$	Indica quantas bicicletas estão aguardando a próxima vaga de bicicletas disponível em $S_i$ , quando $S_i$ estiver lotada.
$PCar_i$	Informa se o veículo de redistribuição está na estação $S_i$ para realizar o balanceamento, que é indicado por $m(PCar_i) = 1$ .
$PR_{ij}$	Representa as bicicletas que saíram da estação $S_i$ com destino à estação $S_j$ .
$PCarR_{iC}$	Informa que o sistema definiu o próximo destino do veículo de redistribuição como o centro de redistribuição. Portanto ele está liberado para partir da estação $S_i$ para o centro de redistribuição caso $m(PCarR_{iC}) = 1$ .
$PCarR_{ij}$	Informa que o sistema definiu o próximo destino do veículo de redistribuição como sendo a estação $S_j$ , estando liberado para partir da estação $S_i$ para a estação $S_j$ quando $m(PCarR_{ij}) = 1$ .
$PCar_C$	Informa se o veículo de redistribuição está no centro de redistribuição aguardando a próxima ronda, o que é indicado quando $m(PCar_C) = 1$ .
$PCarR_{Cj}$	Informa que o sistema definiu o próximo destino do veículo de redistribuição como a estação $S_j$ , estando o veículo liberado para partir do centro de redistribuição para a estação $S_j$ quando $m(PCarR_{Cj}) = 1$ .
$PBSCar$	Representa os suportes ocupados por bicicletas no veículo de redistribuição. A sua marcação $m(PBSCar)$ corresponde ao número de bicicletas disponíveis no veículo.
$PESCar$	Representa os suportes vazios no veículo de redistribuição. A marcação $m(PESCar)$ corresponde ao número de lugares disponíveis no veículo.

Tabela 4.3: Interpretações das transições da rede de Petri proposta.

Transição	Interpretação
$TOut_i$	Modela a retirada de uma bicicleta da estação $S_i$ pelo usuário se houver, em $S_i$ , pelo menos uma bicicleta disponível.
$TIn_i$	Adiciona uma bicicleta à estação $S_i$ pelo usuário, se houver pelo menos um suporte vazio em $S_i$ e não houver ninguém esperando na fila por uma vaga ( $m(PInhib_i) = N_B - C_i$ ).
$TWIn_i$	Adiciona uma bicicleta à estação $S_i$ pelo usuário após ele aguardar por um suporte vazio em $S_i$ .
$TAdd_i$	Modela a ação do veículo adicionar uma bicicleta à estação $S_i$ se o número de bicicletas disponíveis for menor que o número de reordenação $R_i$ ( $m(PES_i) > C_i - R_i$ ) e se houver pelo menos uma bicicleta disponível no veículo ( $m(PBSCar) > 0$ ).
$TRem_i$	Modela a remoção de uma bicicleta da estação $S_i$ pelo veículo se o número de bicicletas disponíveis for maior que o número de reordenação $R_i$ ( $m(PBS_i) > R_i$ ) e se houver pelo menos uma vaga disponível no veículo ( $m(PESCar) > 0$ ).
$TE_{1i}$	Dispara se o número de bicicletas disponíveis na estação $S_i$ for maior que o número de reordenação $R_i$ , mas não há suportes livres no veículo.
$TE_{2i}$	Dispara se o número de bicicletas disponíveis na estação $S_i$ for menor que o número de reordenação $R_i$ , mas não há bicicletas disponíveis no veículo.
$TE_{3i}$	Dispara se o número de bicicletas disponíveis na estação $S_i$ for igual ao número de reordenação $R_i$ .
$T_{ij}$	Essa transição é habilitada quando o usuário chega à estação $S_i$ e a encontra lotada, decidindo ir para a estação $S_j$ (se $i \neq j$ ) ou esperar nela mesma até a próxima vaga ser liberada (se $i = j$ ).
$TInd_{ij}$	Modela a saída de uma bicicleta da estação $S_i$ com destino a estação $S_j$ .
$Tt_{ij}$	Transição temporizada que representa o deslocamento de uma bicicleta da na estação $S_i$ para a estação $S_j$ .
$TCarInd_{iC}$	Dispara quando o veículo de redistribuição sai da estação $S_i$ e escolhe o centro de redistribuição como destino.
$TCarInd_{ij}$	Dispara quando o veículo de redistribuição se move da estação $S_i$ para a estação $S_j$ .
$TCarInd_{Cj}$	Dispara quando o veículo de redistribuição sai do centro de redistribuição e escolhe a estação $S_j$ como destino.
$TtCar_{iC}$	Transição temporizada que representa o deslocamento do veículo de redistribuição da estação $S_i$ para o centro de redistribuição.
$TtCar_{ij}$	Transição temporizada que representa o deslocamento do veículo de redistribuição da estação $S_i$ para a estação $S_j$ .
$TtCar_{Cj}$	Transição temporizada que representa o deslocamento do veículo de redistribuição do centro de redistribuição para a estação $S_j$ .

Note que na figura 4.1 não estão representadas as funções  $\Lambda_d(\cdot)$ , que foram apenas omitidas por todas as transições temporizadas da rede serem estocásticas. Isso ocorre pois todas elas representam movimentações entre diferentes locais do sistema, que são feitas por usuários com diferentes velocidades e pelo veículo de redistribuição que está sujeito a adversidades, como o trânsito.

Apesar do modelo ter mantido os três tipos de sub-redes do modelo proposto em [1, 2], algumas características dos mesmos foram alteradas. As mudanças na sub-rede do fluxo de bicicletas, na sub-rede do controle de estações e na sub-rede do circuito de redistribuição são apresentadas a seguir.

### 4.1.1 Sub-rede do fluxo de bicicletas

Na sub-rede do fluxo de bicicletas, esboçada na figura 4.2, o usuário continua sendo capaz de retirar uma bicicleta de uma estação  $S_i$  qualquer, que não esteja vazia, a partir do disparo da transição  $TOut_i$ . Esse disparo, porém, depende do controle externo  $Esc(TOut_i, PBS_i)$  que representa um conflito degenerado  $C(PBS_i, \{TOut_i\})$ . Essa adição foi feita para que um programa externo defina de forma controlada a entrada dos usuários no sistema.

Após a entrada de um usuário, ele escolhe, assim como em [1, 2], uma estação de destino  $S_j$  qualquer através do sorteio realizado utilizando os valores  $\Lambda_i(TInd_{ij}, PR_i)$ , que escolhe uma das transições  $TInd_{ij}$  para disparar.

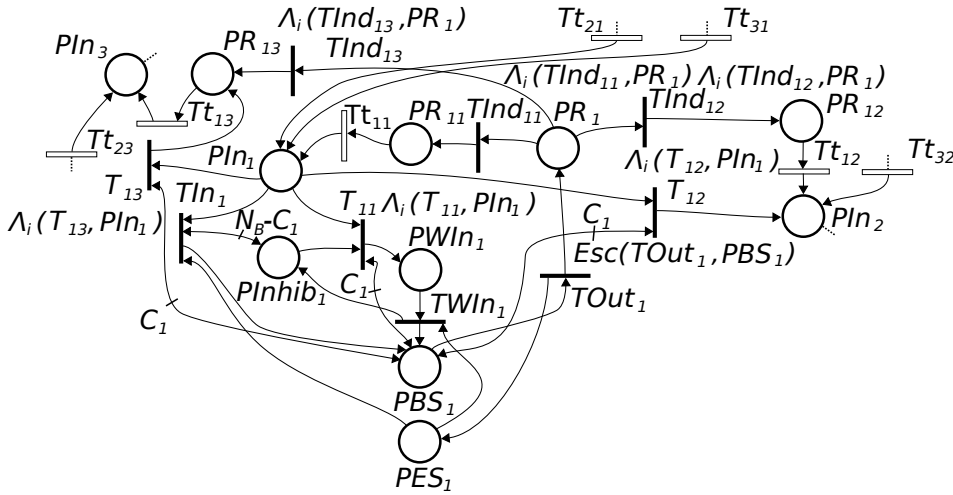


Figura 4.2: Sub-rede do fluxo de bicicletas da estação  $S_1$ .

Quando o usuário chega à estação  $S_j$ , deve ser verificado se há alguma vaga disponível na estação. É verificado também se alguém estava esperando anteriormente por uma vaga na estação através do lugar  $PInhib_j$ , que substituiu o arco inibidor entre o lugar  $PWin_j$  e  $TIn_j$ , para evitar que o usuário ocupasse a vaga antes de quem estava esperando por uma vaga antes dele. Caso a estação tenha

vagas disponíveis, isto é,  $m(PES_j) \geq 1$ , e não tenham pessoas aguardando por vaga ( $m(PInhib_j) = N_b - C_j$ ), a transição  $TIn_j$  dispara, indicando que o usuário conseguiu depositar a bicicleta na estação. Caso a estação esteja cheia, o usuário decide, através dos valores de  $\Lambda_i(T_{jk}, PIn_j)$ , se ele irá se deslocar para uma outra estação  $S_k$  através do disparo de  $T_{jk}$ , ou se ele irá aguardar por uma vaga, caso em que a transição  $T_{jj}$  irá disparar, removendo uma ficha do lugar  $PInhib_j$ .

A substituição do arco inibidor para o lugar  $PInhib_j$  é possível pois a rede de Petri é limitada, o que permite definir um valor máximo de marcação que substitui o arco inibidor, que, no caso do lugar  $PWin_j$ , vale  $N_B - C_j$ .

Para executar a função do arco inibidor, inicialmente, a marcação do novo lugar é  $m(PInhib_j) = N_B - C_j$ . Conforme  $T_{jj}$ , dispara, que representa que um usuário decidiu esperar por uma vaga na estação, uma ficha é retirada de  $PInhib_j$ , o que impede o futuro disparo de  $TIn_j$  por precisar da marcação inicial de  $PInhib_j$  para ficar habilitada. Quando o usuário que estava esperando uma vaga consegue depositar uma bicicleta,  $TWin_j$  dispara, adicionando uma ficha ao  $PInhib_j$  que, caso sua marcação tenha voltado ao seu valor inicial, permite futuros disparos de  $TIn_j$  novamente.

#### 4.1.2 Sub-rede do controle de estações

O funcionamento da sub-rede de controle de estações do novo modelo, ilustrado na figura 4.3 é igual ao do modelo proposto em [1, 2], explicado na seção 3.2.2, realizando trocas de bicicletas entre a estação e o veículo através das transições  $TAdd_i$  e  $TRem_i$  quando o veículo está na estação  $S_i$ , o que representado pela marcação  $m(PCar_i) = 1$ . Além disso, o modelo continua verificando o fim do balanceamento através das transições  $TE_{1i}$ ,  $TE_{2i}$  e  $TE_{3i}$ , cujos critérios foram explicados anteriormente.

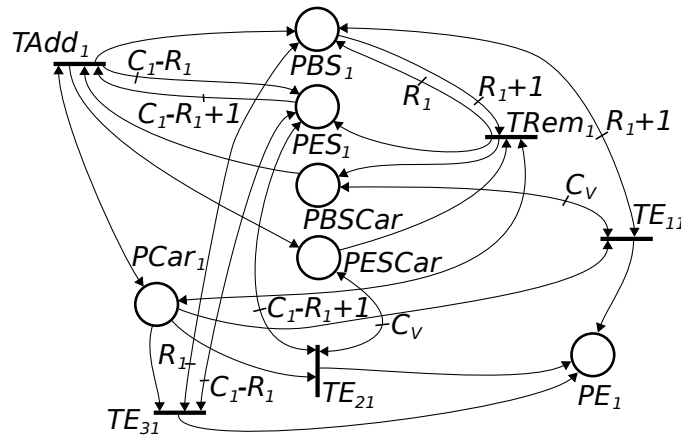


Figura 4.3: Sub-rede do controle de estações da estação  $S_1$ .



### 4.1.3 Sub-rede do fluxo do veículo de redistribuição

A sub-rede do fluxo do veículo de redistribuição é responsável por gerenciar as decisões e transportes entre as estações e o centro de redistribuição efetuados pelo veículo durante o funcionamento do sistema. No modelo, o veículo é capaz de se locomover de uma estação ou centro de redistribuição para qualquer outro lugar do sistema. Para decidir qual é o próximo lugar de destino, o modelo utiliza uma função implementada externamente.

A sub-rede utiliza os lugares  $PCar_C$ ,  $PCarR_{Ci}$ ,  $PCar_i$ ,  $PE_i$ ,  $PCarR_{ij}$ ,  $PCarR_{iC}$  e as transições  $TCarInd_{Ci}$ ,  $TtCar_{Ci}$ ,  $TE_{1i}$ ,  $TE_{2i}$ ,  $TE_{3i}$ ,  $TCarInd_{ij}$ ,  $TtCar_{ij}$ ,  $TCarInd_{iC}$ ,  $TtCar_{iC}$ , em que  $i \in \{1, 2, \dots, N\}$  e  $j \in \{1, 2, \dots, i-1, i+1, \dots, N-1, N\}$ . A figura 4.4 demonstra a sub-rede do fluxo do veículo de redistribuição para a estação  $S_1$ .

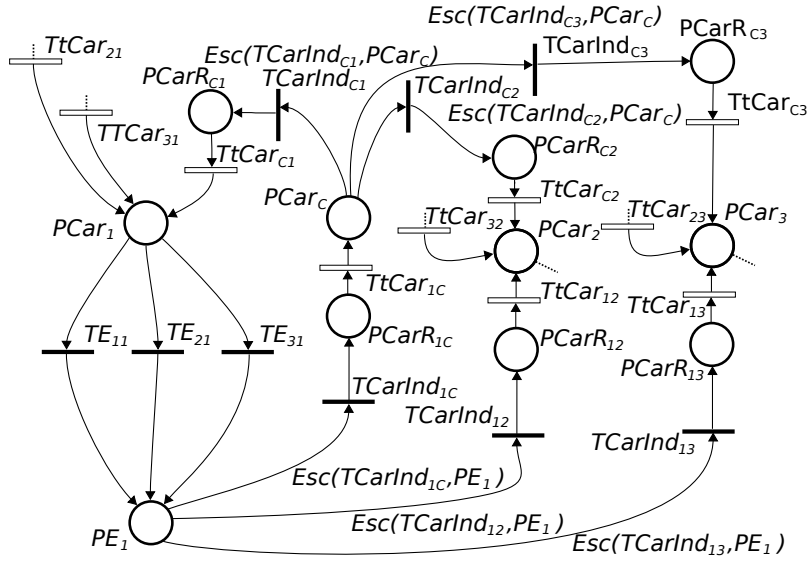


Figura 4.4: Sub-rede do fluxo do veículo de redistribuição da estação  $S_1$  e do centro de redistribuição.

No início, o veículo está estacionado no centro de redistribuição, o que é representado pela marcação  $m(PCar_c) = 1$ , cuja ficha é deslocada entre os lugares da sub-rede para representar o estado do veículo do balanceamento. Como só há um veículo de redistribuição na estação, a sub-rede do fluxo do veículo de redistribuição só possui uma ficha que é deslocada de um lugar para o outro através do disparo das transições.

Para decidir quando o veículo deve sair do centro e para qual estação ele deve ir para fazer seu primeiro balanceamento, é utilizada um controle externo para escolher, através dos  $Esc(TCarInd_{Ci}, PCar_C)$ , qual transição  $TCarInd_{Ci}$  deve disparar, começando o movimento do veículo para a estação escolhida  $S_i$  com a marcação  $(PCarR_{Ci}) = 1$ . Depois de passar um intervalo de tempo definido por uma função

estocástica, a transição temporizada  $TtCar_{C_i}$  dispara, transmitido a ficha para o lugar  $PCar_i$ , representando que o veículo chegou a estação e pode começar a realizar o balanceamento da estação através do sub-rede do controle de estações.

Quando o balanceamento na estação é finalizado, a sub-rede do controle de estações dispara uma das transições  $TE_{1i}$ ,  $TE_{2i}$  ou  $TE_{3i}$ , movendo a ficha para  $PE_i$ , representando que deve ser escolhido, através de uma função externa, um dos  $Esc(TCarInd_{ij}, PE_i)$  para decidir o próximo destino do veículo. Caso o destino escolhido seja outra estação  $S_j$  diferente de  $S_i$ , a transição  $TCarInd_{ij}$  dispara, caso o destino seja o centro de redistribuição,  $TCarInd_{iC}$  é escolhido para disparar. Ambas escolhas resultam em um movimento de ficha semelhante ao ocorrido do lugar  $PCar_C$  até  $PCar_i$  descrito anteriormente, transportando a ficha para o lugar  $PCar_j$ , continuando o balanceamento do veículo na estação  $S_j$ , ou deslocando a ficha para o lugar  $PCar_C$ , transportando o veículo para o centro de redistribuição e finalizando a ronda do veículo.

## 4.2 Matrizes de incidência do modelo proposto para um SPCB com $N$ estações

Como o tipo da rede de Petri do modelo proposto para um SPCB com  $N$  estações é uma rede de Petri estocástica generalizada com controle externo, é possível descrevê-lo através das matrizes  $Post$ ,  $Pre$  e  $A$ , em que  $A = Post - Pre$ . A matriz  $Pre$  permite verificar a habilitação de cada transição do sistema, enquanto  $A$  é usado para compor a seguinte equação que descreve a dinâmica do sistema:

$$m_{k+1}(\underline{p}) = m_k(\underline{p}) + A\sigma(\underline{t}), \quad (4.2)$$

em que  $\underline{p}$  e  $\underline{t}$  são, respectivamente, os vetores dos lugares e das transições da rede de Petri.

Como é inviável a visualização do modelo para um SPCB com um número grande de estações através de uma figura, é necessário descrever o modelo de um sistema de  $N$  estações através das matrizes de incidência  $Pre$  e  $Post$ . Ambas possuem a dimensão  $n_P \times n_T$ , em que  $n_P$  e  $n_T$  são, respectivamente, o número total de lugares e o número total de transições na rede.

Os números  $n_P$  e  $n_T$  crescem polinomialmente conforme  $N$  aumenta, fato que é demonstrado na próxima seção. Por isso, é vantajoso dividir as matrizes  $Pre$  e  $Post$  em submatrizes para facilitar o entendimento das mesmas e a definição dos elementos dos blocos a partir dos arcos da rede. Para auxiliar a separação, os lugares e transições também são divididos em vetores, uma vez que eles definem, respectivamente, as linhas e colunas das matrizes.

Os lugares são agrupados nos seguintes vetores:

$$\underline{pb}_{S_i} = [PIn_i PInhib_i PE_i PES_i PBS_i PR_i PWin_i PCar_i]^T,$$

$$\underline{pr}_{S_i} = [PR_{i1} PR_{i2} \dots PR_{iN}]^T,$$

$$\underline{pcar}_{S_i} = [PCarR_{i1} PCarR_{i2} \dots PCarR_{i(i-1)} PCarR_{iC} PCarR_{i(i+1)} \dots PCarR_{iN}]^T,$$

em que  $i \in \{1, 2, \dots, N\}$ ,

$$\underline{pcarC} = [PCarC PCarR_{C1} PCarR_{C2} \dots PCarR_{CN}]^T,$$

$$\underline{pcar} = [PBSCar PESCar]^T.$$

As transições são agrupadas nos vetores:

$$\underline{tb}_{S_i} = [TOut_i TIn_i TWin_i TAdd_i TRem_i TE_{1i} TE_{2i} TE_{3i}]^T,$$

$$\underline{tr}_{S_i} = [T_{i1} T_{i2} \dots T_{iN}]^T,$$

$$\underline{ti}_{S_i} = [TInd_{i1} TInd_{i2} \dots TInd_{iN}]^T,$$

$$\underline{tt}_{S_i} = [Tt_{i1} Tt_{i2} \dots Tt_{iN}]^T,$$

$$\underline{tcar}_{S_i} = [TCarInd_{i1} TCarInd_{i2} \dots TCarInd_{i(i-1)} TCarInd_{iC} TCarInd_{i(i+1)} \dots TCarInd_{iN}]^T,$$

$$\underline{ttcar}_{S_i} = [TtCar_{i1} TtCar_{i2} \dots TtCar_{i(i-1)} TtCar_{iC} TtCar_{i(i+1)} \dots TtCar_{iN}]^T,$$

em que  $i \in \{1, 2, \dots, N\}$ ,

$$\underline{tcarC} = [TCar_{C1} TCar_{C2} \dots TCar_{CN}]^T,$$

$$\underline{ttcarC} = [TtCar_{C1} TtCar_{C2} \dots TtCar_{CN}]^T,$$

É possível também agrupar os lugares e transições em vetores que representam cada estação  $S_i$ :

$$\underline{p}_{S_i} = [\underline{pb}_{S_i}^T \underline{pr}_{S_i}^T \underline{pcar}_{S_i}^T]^T,$$

$$\underline{t}_{S_i} = [\underline{t}b_{S_i}^T \underline{t}r_{S_i}^T \underline{t}i_{S_i}^T \underline{t}t_{S_i}^T \underline{t}car_{S_i}^T \underline{t}tcar_{S_i}^T]^T.$$

Os grupos de vetores definidos são organizados em dois vetores que representam, respectivamente, todos os lugares e transições da rede do sistema,  $\underline{p}$  e  $\underline{t}$ , que são definidos como:

$$\underline{p} = [\underline{p}_{S_1}^T \dots \underline{p}_{S_N}^T \underline{p}carC^T \underline{p}car^T]^T$$

$$\underline{t} = [\underline{t}_{S_1}^T \dots \underline{t}_{S_N}^T \underline{t}carC^T \underline{t}tcarC^T]$$

Com os vetores definidos, é possível descrever as matrizes *Pre* e *Post* em função de blocos de matrizes. As matrizes *Pre* e *Post* podem ser definidas das seguintes formas:

$$Pre = \begin{array}{c} \underline{p}_{S_1}, \dots, \underline{p}_{S_N} \\ \underline{p}carC \\ \underline{p}car \end{array} \left( \begin{array}{c|c} \underline{t}_{S_1}, \dots, \underline{t}_{S_N} & \underline{t}carC, \underline{t}tcarC \\ \hline PreS & 0 \\ \hline 0 & PreCC \\ \hline PreC & 0 \end{array} \right)$$

$$Post = \begin{array}{c} \underline{p}_{S_1}, \dots, \underline{p}_{S_N} \\ \underline{p}carC, \underline{p}car \end{array} \left( \begin{array}{c|c} \underline{t}_{S_1}, \dots, \underline{t}_{S_N} & \underline{t}carC, \underline{t}tcarC \\ \hline PostS & PostCC \\ \hline PostC & PostCCC \end{array} \right)$$

A seguir, são descritos cada um dos blocos e sub-blocos pertencentes às matrizes *Pre* e *Pos*.

- *PreS*: Matriz diagonal por blocos de dimensões  $(2N^2 + 8N) \times (5N^2 + 8N)$  que relaciona os lugares e transições dos vetores  $\underline{p}_{S_1}, \dots, \underline{p}_{S_N}$  e  $\underline{t}_{S_1}, \dots, \underline{t}_{S_N}$  através de blocos. Assim, pode-se escrever *PreS* como:

$$PreS = \text{blockdiag} \{ PreS_1 \ PreS_2 \ \dots \ PreS_N \}$$

- *PreS<sub>i</sub>*: Matriz de dimensões  $(2N+8) \times (5N+8)$  de uma estação  $S_i$  que relaciona os lugares e transições dos vetores  $\underline{p}_{S_i}$  e  $\underline{t}_{S_i}$ . Ela é definida como:

$$PreS_i = \begin{matrix} \underline{pb}_{S_i} \\ \underline{pr}_{S_i} \\ \underline{pcar}_{S_i} \end{matrix} \begin{pmatrix} \underline{tb}_{S_i} & \underline{tr}_{S_i} & \underline{ti}_{S_i} & \underline{tt}_{S_i} & \underline{tcari}_{S_i} & \underline{tcar}_{S_i} \\ \hline PreSB_i & PreST_i & PreSI_i & 0_{8 \times N} & PreSCI & 0_{8 \times N} \\ 0_{N \times 8} & 0_{N \times N} & 0_{N \times N} & I_{N \times N} & 0_{N \times N} & 0_{N \times N} \\ \hline 0_{N \times 8} & 0_{N \times N} & 0_{N \times N} & 0_{N \times N} & 0_{N \times N} & I_{N \times N} \end{pmatrix}$$

•  $PreSB_i$ : Matriz de dimensões  $8 \times 8$  de uma estação  $S_i$  que relaciona os lugares e transições dos vetores  $\underline{pb}_{S_i}$  e  $\underline{tb}_{S_i}$ . Ela é definida como:

$$PreSB_i = \begin{matrix} PIn_i \\ PInhib_i \\ PE_i \\ PES_i \\ PBS_i \\ PR_i \\ PWin_i \\ PCar_i \end{matrix} \begin{pmatrix} TOut_i & TIn_i & TWin_i & TAdd_i & TRem_i & TE_{1i} & TE_{2i} & TE_{3i} \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & N_B - C_i & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & C_i - R_i + 1 & 0 & 0 & C_i - R_i + 1 & C_i - R_i \\ 1 & 0 & 0 & 0 & R_i + 1 & R_i + 1 & 0 & R_i \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

•  $PreST_i$ : Matriz de dimensões  $8 \times N$  de uma estação  $S_i$  que relaciona os lugares e transições dos vetores  $\underline{pb}_{S_i}$  e  $\underline{tr}_{S_i}$ . Nela, todas as  $N$  colunas  $j$  são definidas como:

$$[1 \ 0 \ 0 \ 0 \ C_i \ 0 \ 0 \ 0]^T \text{ se } i \neq j; [1 \ 1 \ 0 \ 0 \ C_i \ 0 \ 0 \ 0]^T \text{ se } i = j$$

•  $PreSI_i$ : Matriz de dimensões  $8 \times N$  de uma estação  $S_i$  que relaciona os lugares e transições dos vetores  $\underline{pb}_{S_i}$  e  $\underline{ti}_{S_i}$ . Nela, todos os elementos são nulos, exceto a sexta linha, cujos elementos são uns.

•  $PreSCI_i$ : Matriz de dimensões  $8 \times N$  de uma estação  $S_i$  que relaciona os lugares e transições dos vetores  $\underline{pb}_{S_i}$  e  $\underline{tcari}_{S_i}$ . Nela, todos os elementos são nulos, exceto a terceira linha, cujos elementos são uns.

•  $PreC$ : Matriz de dimensões  $2 \times (5N^2 + 8N)$  que relaciona os lugares e transições dos vetores  $\underline{pcar}$  e  $\underline{t}_{S_1}, \dots, \underline{t}_{S_N}$ . É possível dividi-la entre sub-blocos de colunas que representam cada estação. Assim, a matriz é definida como:

$$PreC = \begin{matrix} \underline{t}_{S_1} & \underline{t}_{S_2} & \dots & \underline{t}_{S_N} \\ \hline PreCS_1 & PreCS_2 & \dots & PreCS_N \end{matrix}$$

•  $PreCS_i$ : Matriz de dimensões  $2 \times (5N + 8)$  de uma estação  $S_i$  que relaciona os lugares e transições dos vetores  $\underline{pcar}$  e  $\underline{t}_{S_i}$ . Ela é definida como:

$$PreCS_i = \begin{matrix} PBSCar \\ PESCar \end{matrix} \begin{pmatrix} TOut_i & TIn_i & TWin_i & TAdd_i & TRem_i & TE_{1i} & TE_{2i} & TE_{3i} & \underline{tr}_{S_i}, \underline{ti}_{S_i}, \underline{tt}_{S_i}, \underline{tcari}_{S_i}, \underline{tcar}_{S_i} \\ \hline 0 & 0 & 0 & 1 & 0 & C_v & 0 & 0 & 0_{1 \times 5N} \\ 0 & 0 & 0 & 0 & 1 & 0 & C_v & 0 & 0_{1 \times 5N} \end{pmatrix}$$

- *PreCC*: Matriz de dimensões  $(N + 1) \times (2N)$  que relaciona os lugares e transições dos vetores  $\underline{pcarC}$  e  $\underline{tcarC}, \underline{ttcarC}$ . A matriz é formada por dois sub-blocos de colunas:

$$PreCC = \begin{bmatrix} \underline{tcarC} & \underline{ttcarC} \\ PreCCC & PreCCT \end{bmatrix}$$

- *PreCCC*: Matriz de dimensões  $(N + 1) \times N$  que relaciona os lugares e transições dos vetores  $\underline{pcarC}$  e  $\underline{tcarC}$ . Nela, todos os elementos são nulos, exceto a primeira linha, cujos elementos são uns.

- *PreCCT*: Matriz de dimensões  $(N + 1) \times (N)$  de uma estação  $S_i$  que relaciona os lugares e transições dos vetores  $\underline{pcarC}$  e  $\underline{ttcarC}$ . Ela é definida como:

$$PreCCT = \begin{pmatrix} 0_{1 \times N} \\ I_{N \times N} \end{pmatrix}$$

- *PostS*: Matriz de dimensões  $(2N^2 + 8N) \times (5N^2 + 8N)$  que relaciona os lugares e transições dos vetores  $\underline{p}_{S_1}, \dots, \underline{p}_{S_N}$  e  $\underline{t}_{S_1}, \dots, \underline{t}_{S_N}$  através de blocos organizados em colunas  $PostSc_i$  e blocos organizados em diagonais  $PostS_i$ , em que  $i \in \{1, 2, \dots, N\}$ . A descrição de *PostS*, portanto, é a soma:

$$PostS = blockdiag \{ PostS_1 \ PostS_2 \ \dots \ PostS_N \} + [ PostSc_1 \ PostSc_2 \ \dots \ PostSc_N ]$$

- *PostS<sub>i</sub>*: Matriz de dimensões  $(2N + 8) \times (5N + 8)$  de uma estação  $S_i$  que relaciona os lugares e transições dos vetores  $\underline{p}_{S_i}$  e  $\underline{t}_{S_i}$ . Ela é definida como:

$$PostS_i = \begin{matrix} & \underline{tb}_{S_i} & \underline{tr}_{S_i} & \underline{ti}_{S_i} & \underline{tt}_{S_i} & \underline{tcari}_{S_i} & \underline{tcar}_{S_i} \\ \underline{pb}_{S_i} & \left( \begin{array}{c|c|c|c|c|c} PostSB_i & PostST_i & 0_{8 \times N} & 0_{8 \times N} & 0_{8 \times N} & 0_{8 \times N} \\ \hline 0_{N \times 8} & PostSTR_i & I_{N \times N} & 0_{N \times N} & 0_{N \times N} & 0_{N \times N} \\ \hline 0_{N \times 8} & 0_{N \times N} & 0_{N \times N} & 0_{N \times N} & I_{N \times N} & 0_{N \times N} \end{array} \right) \\ \underline{pr}_{S_i} & & & & & & \\ \underline{pcar}_{S_i} & & & & & & \end{matrix}$$

- *PostSc<sub>i</sub>*: Matriz de dimensões  $(2N^2 + 8N) \times (5N + 8)$  de uma estação  $S_i$  que, junto com *PostS<sub>i</sub>*, relaciona os lugares e transições dos vetores  $\underline{p}_{S_1}, \dots, \underline{p}_{S_N}$  e  $\underline{t}_{S_i}$ . É possível dividi-la entre sub-blocos de colunas. Assim, a matriz é definida como:

$$PostSc_i = \begin{bmatrix} \underline{tb}_{S_i}, \underline{tr}_{S_i}, \underline{ti}_{S_i} & \underline{tt}_{S_i} & \underline{tcari}_{S_i} & \underline{tcar}_{S_i} \\ 0_{(2N^2+8N) \times (2N+8)} & PostTT_i & 0_{(2N^2+8N) \times (N+8)} & PostTTCar_i \end{bmatrix}$$

•  $PostSB_i$ : Matriz de dimensões  $8 \times 8$  de uma estação  $S_i$  que relaciona os lugares e transições dos vetores  $\underline{pb}_{S_i}$  e  $\underline{tb}_{S_i}$ . Ela é definida como:

$$PostSB_i = \begin{matrix} & TOut_i & TIn_i & TWIN_i & TAdd_i & TRem_i & TE_{1i} & TE_{2i} & TE_{3i} \\ \begin{matrix} PIn_i \\ PInhib_i \\ PE_i \\ PES_i \\ PBS_i \\ PR_i \\ PWIN_i \\ PCar_i \end{matrix} & \left( \begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & N_B - C_i & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & C_i - R_i & 1 & 0 & C_i - R_i + 1 & C_i - R_i & \\ 0 & 1 & 1 & 1 & R_i & R_i + 1 & 0 & R_i & \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{array} \right) \end{matrix}$$

•  $PostST_i$ : Matriz de dimensões  $8 \times N$  de uma estação  $S_i$  que relaciona os lugares e transições dos vetores  $\underline{pb}_{S_i}$  e  $\underline{tr}_{S_i}$ . Nela, todas as  $N$  colunas  $j$  são definidas como:

$$[0 \ 0 \ 0 \ 0 \ C_i \ 0 \ 0 \ 0]^T \text{ se } i \neq j; [0 \ 0 \ 0 \ 0 \ C_i \ 0 \ 1 \ 0]^T \text{ se } i = j$$

•  $PostSTR_i$ : Matriz de dimensões  $N \times N$  de uma estação  $S_i$  que relaciona os lugares e transições dos vetores  $\underline{pr}_{S_i}$  e  $\underline{tr}_{S_i}$ . Ela é uma matriz diagonal em que todos os elementos da diagonal, exceto o elemento da  $i$ -ésima linha e  $i$ -ésima coluna, são unitários, os outros elementos da matriz são nulos.

•  $PostTT_i$ : Matriz de dimensões  $(2N^2 + 8N) \times N$  de uma estação  $S_i$  que relaciona os lugares e transições dos vetores  $\underline{p}_{S_1}, \dots, \underline{p}_{S_N}$  e  $\underline{tt}_{S_i}$ . É possível dividi-la entre sub-blocos de linhas que representam cada estação. Assim, a matriz é definida como:

$$PostTT_i = \begin{matrix} \underline{p}_{S_1} & \underline{p}_{S_2} & \dots & \underline{p}_{S_N} \\ [PostTT_{i1} & PostTT_{i2} & \dots & PostTT_{iN}]^T \end{matrix}$$

•  $PostTT_{ij}$ : Matriz de dimensões  $(2N + 8) \times N$  das estações  $S_i$  e  $S_j$  que relaciona os lugares e transições dos vetores  $\underline{p}_{S_j}$  e  $\underline{tt}_{S_i}$ . Nela, todos elementos são nulos exceto o elemento da primeira linha e  $j$ -ésima coluna, cujo valor é um.

•  $PostTTCar_i$ : Matriz de dimensões  $(2N^2 + 8N) \times N$  da estação  $S_i$  que relaciona os lugares e transições dos vetores  $\underline{p}_{S_1}, \dots, \underline{p}_{S_N}$  e  $\underline{tcar}_{S_i}$ . É possível dividi-la entre sub-blocos de linhas que representam cada estação. Assim, a matriz é definida como:

$$PostTTCar_i = \begin{matrix} \underline{p}_{S_1} & \underline{p}_{S_2} & \dots & \underline{p}_{S_N} \\ [PostTTCar_{i1} & PostTTCar_{i2} & \dots & PostTTCar_{iN}]^T \end{matrix}$$

•  $PostTTCar_{ij}$ : Matriz de dimensões  $(2N + 8) \times N$  das estações  $S_i$  e  $S_j$  que relaciona os lugares e transições dos vetores  $\underline{p}_{S_j}$  e  $\underline{tcar}_{S_i}$ . Nela, se  $i = j$ , todos os

elementos são nulos. Caso contrário, todos os elementos são nulos exceto o elemento da oitava linha e  $j$ -ésima coluna, cujo valor é um.

- *PostC*: Matriz de dimensões  $(N + 3) \times (5N^2 + 8N)$  que relaciona os lugares e transições dos vetores  $\underline{pcarC}$ ,  $\underline{pcar}$  e  $\underline{t}_{S_1}, \dots, \underline{t}_{S_N}$ . É possível dividi-la entre sub-blocos de colunas que representam cada estação. Assim, a matriz é definida como:

$$PostC = \begin{bmatrix} \underline{t}_{S_1} & \underline{t}_{S_2} & \dots & \underline{t}_{S_N} \\ PostCS_1 & PostCS_2 & \dots & PostCS_N \end{bmatrix}$$

- *PostCS<sub>i</sub>*: Matriz de dimensões  $(N + 3) \times (5N + 8)$  de uma estação  $S_i$  que relaciona os lugares e transições dos vetores  $\underline{pcarC}$ ,  $\underline{pcar}$  e  $\underline{t}_{S_i}$ . Ela é definida como:

$$PostCS_i = \begin{matrix} & TOut_i & TIn_i & TWIn_i & TAdd_i & TRem_i & TE1_i & TE2_i & TE3_i & \underline{tr}_{S_i}, \underline{ti}_{S_i}, \underline{tt}_{S_i}, \underline{tcari}_{S_i} & \underline{tcar}_{S_i} \\ \underline{pcarC} & 0_{(N+1) \times 1} & 0_{(N+1) \times 1} & 0_{(N+1) \times 1} & 0_{(N+1) \times 1} & 0_{(N+1) \times 1} & 0_{(N+1) \times 1} & 0_{(N+1) \times 1} & 0_{(N+1) \times 1} & 0_{(N+1) \times 4N} & PostCTS_i \\ \underline{PBSCar} & 0 & 0 & 0 & 0 & 1 & C_v & 0 & 0 & 0_{1 \times 4N} & 0_{1 \times N} \\ \underline{PESCar} & 0 & 0 & 0 & 1 & 0 & 0 & C_v & 0 & 0_{1 \times 4N} & 0_{1 \times N} \end{matrix}$$

- *PostCTS<sub>i</sub>*: Matriz de dimensões  $(N + 1) \times N$  da estação  $S_i$  que relaciona os lugares e transições dos vetores  $\underline{pcarC}$  e  $\underline{tcar}_{S_i}$ . Nela, todos os elementos são nulos exceto o elemento da primeira linha e  $i$ -ésima coluna, cujo valor é um.

- *PostCC*: Matriz de dimensões  $(2N^2 + 8N) \times 2N$  que relaciona os lugares e transições dos vetores  $\underline{p}_{S_1}, \dots, \underline{p}_{S_N}$  e  $\underline{tcarC}$ ,  $\underline{ttcarC}$ . É possível dividi-la entre sub-blocos de linhas que representam cada estação. Assim, a matriz é definida como:

$$PostCC = \begin{bmatrix} \underline{p}_{S_1} & \underline{p}_{S_2} & \dots & \underline{p}_{S_N} \\ PostCC_1 & PostCC_2 & \dots & PostCC_N \end{bmatrix}^T$$

- *PostCC<sub>i</sub>*: Matriz de dimensões  $(8 + 2N) \times 2N$  da estação  $S_i$  que relaciona os lugares e transições dos vetores  $\underline{p}_{S_i}$  e  $\underline{tcarC}$ ,  $\underline{ttcarC}$ . Nela, todos os elementos são nulos, exceto o elemento da oitava linha e da coluna  $(N + i)$ , cujo valor é um.

- *PostCCC*: Matriz de dimensões  $(N + 3) \times 2N$  que relaciona os lugares e transições dos vetores  $\underline{pcarC}$ ,  $\underline{pcar}$  e  $\underline{tcarC}$ ,  $\underline{ttcarC}$ . Ela é definida como:

$$PostCCC = \begin{matrix} & \underline{tcarC} & \underline{ttcarC} \\ PCarC & 0_{1 \times N} & 0_{1 \times N} \\ PCarR_{C_1}, \dots, PCarR_{C_N} & I_{N \times N} & 0_{N \times N} \\ \underline{pcar} & 0_{2 \times N} & 0_{2 \times N} \end{matrix}$$

Com as matrizes *Pre* e *Post* descritas em função de  $N$ , é possível encontrar a matriz de incidência  $A$ , o que permite a implementação de um software que simule o comportamento de SPCBs com diferentes características e quantidades de estações.



### 4.3 Ordem do modelo proposto

A ordem do modelo é dada pela quantidade de lugares e transições que a rede de Petri possui, por isso, como a matriz de incidência  $A$  foi definida em função de  $N$  e vetores de lugares e transições na seção 4.2, é possível calcular o número total de nós da rede verificando quantos lugares e transições cada um desses vetores possuem.

Os lugares que representam as estações são agrupados em três principais vetores em função de uma estação  $S_i$ :  $\underline{pb}_{S_i}$ ,  $\underline{pr}_{S_i}$  e  $\underline{pcar}_{S_i}$ .  $\underline{pb}_{S_i}$  possui os oito lugares:  $PIn_i, PInhib_i, PE_i, PES_i, PBS_i, PR_i, PWin_i$  e  $PCar_i$ .  $\underline{pr}_{S_i}$  e  $\underline{pcar}_{S_i}$  possuem, separadamente,  $N$  lugares, por isso, os dois totalizam  $2N$  lugares. Assim, cada estação possui  $2N + 8$  lugares, totalizando  $2N^2 + 8N$  para todas as estações.

Depois das estações, os lugares do vetor  $\underline{pcar}_C$  representam os lugares do centro de redistribuição, este que possui um lugar próprio  $PCar_C$ , e outros  $N$  lugares para cada estação  $S_i$ :  $PCarR_{C1}, PCarR_{C2}, \dots, PCarR_{CN}$ . Assim, o vetor possui  $N + 1$  lugares.

Finalmente, a rede possui mais dois lugares,  $PBSCar$  e  $PESCar$ , que representam, respectivamente, a quantidade de bicicletas e de vagas no veículo de redistribuição.

Somando todos os lugares mencionados, é possível concluir que a rede possui um total de  $2N^2 + 9N + 3$  lugares.

As transições que representam as estações, diferente dos lugares, são divididas em seis vetores, em que o primeiro,  $\underline{tb}_{S_i}$ , possui as 8 transições:  $TOut_i, TIn_i, TWin_i, TAdd_i, TRem_i, TE_{1i}, TE_{2i}$  e  $TE_{3i}$ . Cada um dos outros cinco vetores  $\underline{tr}_{S_i}^T, \underline{ti}_{S_i}^T, \underline{tl}_{S_i}^T, \underline{tcar}_i^T$  e  $\underline{ttcar}_{S_i}$  possuem  $N$  transições, totalizando  $5N$  transições. Assim, o total de transições de cada estação é  $5N + 8$ , totalizando  $5N^2 + 8N$  transições para todas as estações.

Além das transições das estações, existem as transições do centro de redistribuição, representadas pelos vetores  $\underline{tcar}_C$  e  $\underline{ttcar}_C$ . Cada vetor possui  $N$  transições. Por isso, o total de transições dos dois vetores é  $2N$ .

Somando os totais explicados anteriormente, o número de transições da rede é  $5N^2 + 10N$ .

Com as quantidades de lugares e transições da rede definidas, é possível descrever a ordem do modelo proposto como:

$$n_P = 2N^2 + 9N + 3$$

$$n_T = 5N^2 + 10N$$

## Capítulo 5

# Estudo do sistema de bicicletas compartilhadas Integra UFRJ para a implementação de um modelo para simulações

Para testar e validar o modelo proposto neste trabalho, é vantajoso utilizar um SPCB planejado para auxiliar no transporte de uma região real, pois além de ser possível adquirir informações reais da demanda da região, futuramente será possível testar os algoritmos de balanceamento propostos no trabalho em um sistema real.

O SPCB escolhido para ser modelado é o sistema de bicicletas compartilhadas da Universidade Federal do Rio de Janeiro (UFRJ), o projeto Integra UFRJ, que foi implementado sem uma estratégia de balanceamento explícita.

Neste capítulo, inicialmente, será descrito, na seção 5.1, o projeto Integra UFRJ e a região onde ele foi implementado. Em seguida, na seção 5.2, serão expostos algumas informações sobre a estrutura do projeto. Após a exposição da estrutura, serão analisados os comportamentos dos usuários do sistema na seção 5.3. Finalmente, será calculado, na seção 5.4, o tempo que leva para os elementos do sistema se locomoverem entre as estações.

### 5.1 Descrição do projeto Integra UFRJ

Criado pelo Fundo Verde de Desenvolvimento e Energia para a Cidade Universitária da Universidade Federal do Rio de Janeiro [12], o sistema de bicicletas compartilhadas da Cidade Universitária da UFRJ foi inaugurado no dia 12 de setembro de 2017. O sistema possui 8 estações posicionadas de acordo com as maiores demandas por deslocamentos internos na Cidade Universitária, com um total de 60 bicicle-

tas disponíveis, número que permite uma rotina de manutenção entre as bicicletas pertencentes ao projeto [13].

A Cidade Universitária, também conhecida como Ilha do Função, é um bairro situado na cidade do Rio de Janeiro que contém a maioria das unidades da Universidade Federal do Rio de Janeiro. O bairro ocupa uma área superior a cinco milhões de metros quadrados.

Além das unidades da UFRJ, o bairro possui centros de pesquisa de diversas empresas. Por ser uma região que envolve diversos tipos de unidades, a necessidade por mobilidade urbana na região é alta, o que motivou a implementação de oito estações de bicicletas nos lugares com maiores demandas, como mostra o mapa da figura 5.1.

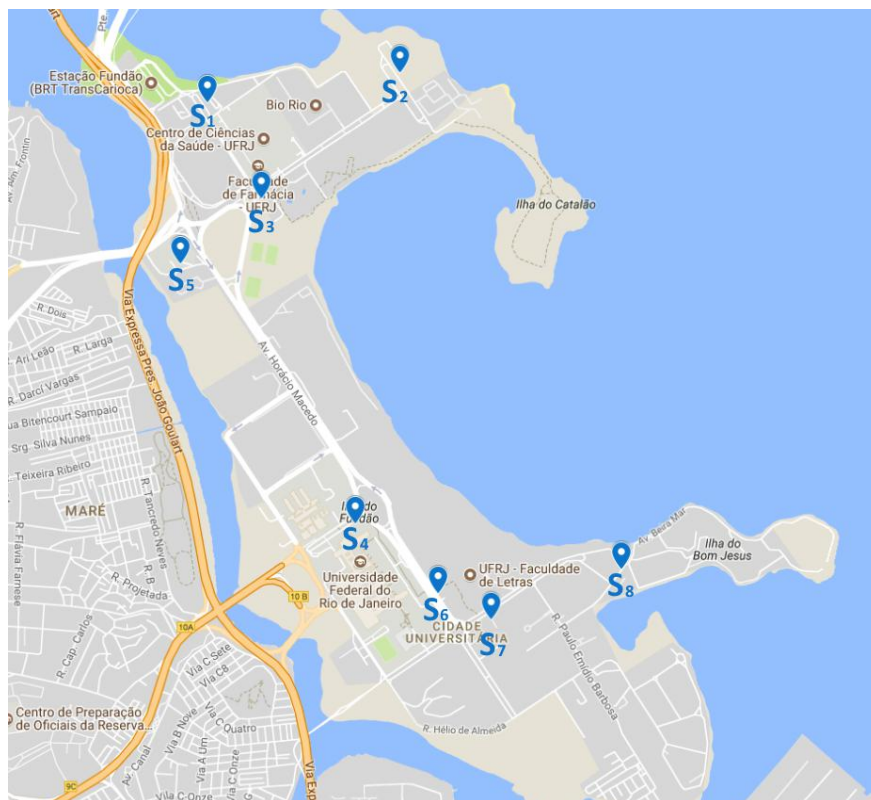


Figura 5.1: Mapa com as oito estações de bicicletas do Integra UFRJ na Cidade Universitária.

## 5.2 Estrutura e uso do Integra UFRJ

Para facilitar o tratamento dos dados das estações, elas foram numeradas de um a oito, como mostrado na figura 5.1. Além disso, o sistema possui uma plataforma web [14] capaz de mostrar a quantidade de bicicletas e de vagas em cada estação, o que permite encontrar a capacidade total  $C_i$  de bicicletas de cada estação  $S_i$ . Com

essas informações, foi gerada a tabela 5.1 que relaciona números de identificação das estações com o lugar físico da Cidade Universitária e a capacidade de cada estação.

Tabela 5.1: Número e local físico de cada estação.

Número da estação	Local físico da estação	Capacidade da estação ( $C_i$ )
$S_1$	Terminal UFRJ	21
$S_2$	Alojamento Estudantil	14
$S_3$	Restaurante Universitário	14
$S_4$	CT/CCMN	14
$S_5$	CT2	7
$S_6$	Letras	12
$S_7$	Reitoria	14
$S_8$	COPPEAD	8

O Grupo Verde disponibilizou um relatório do rendimento do sistema durante o ano de 2017 [13], este que é utilizado neste trabalho para estimar algumas características do sistema. Verificando o uso do sistema durante os meses entre setembro e dezembro, foi possível notar que o mês mais estável foi o mês de novembro. Por isso, são consideradas as informações de novembro desse documento para as análises.

No relatório, foi mencionado que foram feitas 5143 viagens no mês de novembro. A partir desse número, é possível obter uma média de quantas viagens foram efetuadas por dia. Como praticamente não houveram viagens nos fins de semana, só são considerados os 22 dias úteis do mês de novembro. o que resulta em  $233,7$  viagens/dia.

Outra maneira de se chegar ao número de viagens por dia é considerando o número de viagens por bicicletas por dia, que vale 3,896 para 60 bicicletas. Como, dessas 60 bicicletas, somente 40 são utilizadas por dia, deve-se converter essa taxa multiplicando-a por 40 e dividindo-a por 60. O resultado dessa conta é  $5,85$  viagens/bicicletas/dia. Multiplicando o resultado pelo número efetivo de bicicletas na estação  $N_B = 40$ , é possível encontrar o valor  $234$  viagens/dia, o que valida o valor encontrado no parágrafo anterior.

Mesmo que tenha sido possível encontrar os valores da quantidade de viagens por dia que geralmente ocorrem, as informações não foram suficientes para estimar quantas viagens ocorreriam caso o sistema possuísse uma estratégia de balanceamento proposto neste trabalho, que, conseqüentemente, poderia aumentar o número de viagens. Para estimar essas viagens extras, são adicionadas 100 viagens ao valor encontrado, totalizando  $334$  viagens/dia.

Outro relatório utilizado para a descrição do sistema é o que foi feito especificamente para o mês de novembro de 2017 [15]. Nele, foi possível verificar que o sistema funciona das 6:00h até 22:00h. Portanto, em cada dia, o sistema funciona por 16:00h.

### 5.3 Comportamento dos usuários

Para estimar a quantidade de usuários que retiram bicicletas de cada estação, é utilizada a tabela 5.2, que possui a porcentagem de viagens que começaram em cada estação em relação ao total de viagens.

Tabela 5.2: Porcentagem de retirada de bicicletas de cada estação.

<b>Estação de origem</b>	<b>Porcentagem em relação à todas as viagens</b>
$S_1$ - Terminal UFRJ	16,71%
$S_2$ - Alojamento Estudantil	6,79%
$S_3$ - Restaurante Universitário	20,58%
$S_4$ - CT/CCMN	22,97%
$S_5$ - CT2	3,86%
$S_6$ - Letras	15,86%
$S_7$ - Reitoria	9,24%
$S_8$ - COPPEAD	4%

Com a porcentagem de viagens que originaram de cada estação e o número total de viagens por dia 334, é composto um vetor  $N_{pe}$  que representa quantos usuários retiram bicicletas de cada estação durante o dia, que conseqüentemente indica a quantidade de vezes que cada transição  $TOut_i$  de cada estação  $S_i$  dispara. Os valores dos elementos do vetor  $N_{pe}$  são:

$$N_{pe} = [53 \ 23 \ 69 \ 77 \ 13 \ 53 \ 31 \ 13]^T$$

Outra informação importante sobre a retirada de bicicletas pelos usuários é sobre os picos de utilização das estações. Situações, como, por exemplo, a movimentação de um grupo de usuários por causa de almoço, podem gerar picos de uso em certas estações. Para verificar esses picos, é utilizada a tabela 5.3 que contém a porcentagem de retiradas de bicicletas em relação aos horários de funcionamento do sistema.

Tabela 5.3: Porcentagem de retirada de bicicletas ao longo do dia.

Horário	Porcentagem de retiradas
6h-8h	5,09%
8h-10h	8,69%
10h-12h	19%
12h-14h	17,51%
14h-16h	15,05%
16h-18h	17,69%
18h-20h	11,65%
20h-22h	2,13%

Baseado na tabela e nos lugares próximos às estações, estão explícitos dois picos:

- Durante o horário de almoço, às 12:00h: Nesse horário, as estações do CT/CCMN e Letras, que são algumas das estações mais utilizadas, recebem uma maior demanda por bicicletas por estarem envolvidas na movimentação dos usuários para irem até o restaurante para almoçarem.

- Depois do almoço, às 13:00h: Nesse horário, a estação do Restaurante Universitário recebe uma maior demanda por bicicletas, pois os usuários que almoçaram devem voltar para os seus respectivos campus.

- Durante a saída dos alunos do campus, às 17:00h: Nesse horário, as estações do CT/CCMN e Letras recebem uma maior demanda por bicicletas por estarem envolvidas na movimentação dos usuários para irem à estação do Terminal UFRJ para voltarem para suas casas.

Além dos dois picos notados pela tabela, foi considerado um pico de usuário às 8:00h na estação do Terminal UFRJ, pois, nesse horário, os estudantes chegam ao terminal para depois se dirigirem às suas unidades de destino.

Quando o usuário sai de uma estação, ele deve escolher uma das oito estações como destino. No modelo proposto em redes de Petri, esse comportamento é representado pelos conflitos, existentes em cada estação  $S_i$ , entre as transições  $TInd_{ij}$ , em que  $j$  representa outra estação  $S_j$ . Para solucionar esses conflitos, a rede utiliza uma função estocástica para cada estação, essa que escolhe uma das transições  $TInd_{ij}$  para disparar de acordo com uma probabilidade definida para cada uma.

Para definir a probabilidade de cada  $TInd_{ij}$  disparar, é necessário dividir, para cada estação  $S_i$  do sistema, oito porcentagens  $\lambda_{ij}$  que representam as chances do usuário escolher cada uma das estações  $S_j$  como destino. Para limitar essa escolha entre as oito estações, a seguinte equação deve ser respeitada:

$$\forall i \in \{1, 2, \dots, 8\}, \sum_{j=1}^8 \lambda_{ij} = 1. \quad (5.1)$$

Uma forma de estimar as porcentagens  $\lambda_{ij}$  do Integra UFRJ é estudando, para um intervalo de tempo, as quantidades de vezes que cada possível trajeto de todo o sistema foi utilizado. Tais quantidades podem ser atribuídas às porcentagens  $\lambda_{ij}$  por indicar o comportamento dos usuários em cada estação.

Apesar dos relatórios mostrarem várias informações, nenhum dos dois apresentou, de forma explícita, a porcentagem de viagens que foram feitas de uma estação para a outra para todas as possibilidades de trajetos. Por isso, são utilizadas as informações das porcentagens de vezes que foram devolvidas bicicletas à cada estação para estimar a porcentagem do uso de cada trajeto de todo o sistema. As porcentagens são obtidas no relatório de outubro [16] e são mostradas na tabela 5.4.

Tabela 5.4: Porcentagem de devoluções de bicicletas à cada estação.

<b>Estação de destino</b>	<b>Porcentagem em relação à todas as viagens</b>
$S_1$ - Terminal UFRJ	19,24%
$S_2$ - Alojamento Estudantil	6,77%
$S_3$ - Restaurante Universitário	21,35%
$S_4$ - CT/CCMN	22,15%
$S_5$ - CT2	2,73%
$S_6$ - Letras	15,54%
$S_7$ - Reitoria	8,29%
$S_8$ - COPPEAD	3,94%

Apesar do documento utilizado em análises anteriores ser de novembro, ambas as tabelas 5.2 e 5.4 são do mês de outubro, pois ao analisar as tabelas de porcentagem de retiradas de bicicletas de cada estação do mês de novembro, foi possível concluir que havia um erro nela, uma vez que a soma de todas as porcentagens de todas as estações não totalizava 100%.

Para facilitar o uso dos dados apresentados, as porcentagens da tabela 5.4 são agrupadas no vetor: *viagOut*, em que os elementos estão ordenados de acordo com seus números de estação.

Outra hipótese considerada é a de que, como o espaço da Cidade Universitária não é de lazer, ou seja, o sistema não é utilizado para recreação, é considerado que os usuários não devolvem bicicletas às estações de onde eles retiraram as mesmas.

Para definir cada porcentagem  $\lambda_{ij}$  de cada estação  $S_i$  e  $S_j$ , são utilizados os

valores de  $\underline{viagOut}$  para atribuir diferentes pesos para cada estação de destino representada por  $j$ . Como a porcentagem de  $\lambda_{ii}$  foi considerada, por hipótese, nula, o valor do elemento  $\underline{viagOut}(i)$  deve ser subtraído do cálculo desse peso para que a soma de todos os  $\lambda_{ij}$  da estação  $S_i$  resulte em um, respeitando a igualdade (5.1). Assim, as porcentagens são definidas através da equação:

$$\lambda_{ij} = \begin{cases} \frac{\underline{viagOut}(j)}{100\% - \underline{viagOut}(i)}, & \text{se } i \neq j; \\ 0, & \text{se } i = j. \end{cases} \quad (5.2)$$

Com a equação de  $\lambda_{ij}$  definida, é possível escrever a matriz  $DIda$ , com dimensão  $8 \times 8$ , que representa as probabilidades do usuário que retira uma bicicleta da estação da linha  $i$  escolher como destino uma estação da coluna  $j$ .

$$DIda = \begin{matrix} & S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 \\ \begin{matrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \end{matrix} & \left( \begin{array}{cccccccc} 0,00\% & 8,38\% & 26,43\% & 27,42\% & 3,37\% & 19,23\% & 10,26\% & 4,87\% \\ 20,63\% & 0,00\% & 22,89\% & 23,75\% & 2,92\% & 16,66\% & 8,89\% & 4,22\% \\ 24,45\% & 8,60\% & 0,00\% & 28,15\% & 3,47\% & 19,75\% & 10,53\% & 5,00\% \\ 24,71\% & 8,69\% & 27,42\% & 0,00\% & 3,50\% & 19,95\% & 10,64\% & 5,06\% \\ 19,77\% & 6,95\% & 21,94\% & 22,76\% & 0,00\% & 15,97\% & 8,52\% & 4,05\% \\ 22,77\% & 8,01\% & 25,27\% & 26,22\% & 3,23\% & 0,00\% & 9,81\% & 4,66\% \\ 20,97\% & 7,38\% & 23,27\% & 24,14\% & 2,97\% & 16,94\% & 0,00\% & 4,29\% \\ 20,02\% & 7,04\% & 22,22\% & 23,05\% & 2,84\% & 16,17\% & 8,62\% & 0,00\% \end{array} \right) \end{matrix}$$

Além do comportamento explicitado em  $DIda$ , é necessário também determinar como o usuário reage quando ele chega à uma estação  $S_i$  que está cheia. Essa ocasião é representada na rede de Petri através do conflito entre as transições  $T_{ij}$ , que, quando disparadas, representam o redirecionamento do trajeto do usuário para a estação  $S_j$ . Assim como os conflitos entre as transições  $TInd_{ij}$ , esses conflitos também são solucionados através de funções estocásticas.

Para definir as probabilidades de cada transição  $T_{ij}$  disparar, é suposto que os usuários, nessa situação, tem grandes chances de aguardar por uma vaga na estação. Porém, caso ele decida ir para outra estação, o novo destino escolhido é uma estação próxima da estação que está lotada. Baseando-se nesses fatores e no mapa da figura 5.1, foi feita a matriz  $DReIda$ , que define a probabilidade do usuário, ao encontrar a estação da linha  $i$  cheia, escolher aguardar por uma vaga de bicicleta, representado pela coluna  $i$ , ou viajar para uma outra estação de alguma coluna  $j$ .



$$DReIda = \begin{matrix} & S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 \\ \begin{matrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \end{matrix} & \begin{pmatrix} 90\% & 0\% & 10\% & 0\% & 0\% & 0\% & 0\% & 0\% & 0\% \\ 0\% & 100\% & 0\% & 0\% & 0\% & 0\% & 0\% & 0\% & 0\% \\ 10\% & 0\% & 80\% & 0\% & 10\% & 0\% & 0\% & 0\% & 0\% \\ 0\% & 0\% & 0\% & 80\% & 0\% & 20\% & 0\% & 0\% & 0\% \\ 0\% & 0\% & 10\% & 0\% & 90\% & 0\% & 0\% & 0\% & 0\% \\ 0\% & 0\% & 0\% & 5\% & 0\% & 80\% & 15\% & 0\% & 0\% \\ 0\% & 0\% & 0\% & 0\% & 0\% & 20\% & 80\% & 0\% & 0\% \\ 0\% & 0\% & 0\% & 0\% & 0\% & 0\% & 0\% & 0\% & 100\% \end{pmatrix} \end{matrix}$$

## 5.4 Tempos gastos pelo veículo de redistribuição e pelas bicicletas para transitarem pelo sistema

Para calcular o tempo médio que leva para as bicicletas e o veículo de redistribuição transitarem de um local para a outro no sistema, incluindo o centro de redistribuição  $C$  para o veículo que é posicionado na estação  $S_4$ , são encontradas, primeiramente, as distancias que devem ser percorridas para ir de um local ao outro. Tais distâncias foram encontradas utilizando a plataforma web *Google Maps* e agrupadas nas matrizes  $D_c$  e  $D_b$ , que representam as distâncias, em metros, percorridas das estações e centro de redistribuição das linhas  $i$  até as estações e centro de redistribuição das colunas  $j$ , respectivamente, pelo veículo de redistribuição e pelas bicicletas.

$$D_c = \begin{matrix} & S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 & C \\ \begin{matrix} C \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \end{matrix} & \begin{pmatrix} 2900 & 3300 & 2000 & 0 & 2500 & 1200 & 1100 & 1900 & 0 \\ 0 & 3200 & 2100 & 4300 & 2400 & 4800 & 4700 & 5400 & 4300 \\ 1500 & 0 & 1300 & 3500 & 1700 & 4000 & 3900 & 4700 & 3500 \\ 850 & 1200 & 0 & 2600 & 750 & 3100 & 3000 & 3800 & 2600 \\ 2900 & 3300 & 2000 & 0 & 2500 & 1200 & 1100 & 1900 & 0 \\ 1500 & 1900 & 650 & 2000 & 0 & 2600 & 2500 & 3300 & 2000 \\ 3000 & 3400 & 2200 & 1300 & 2600 & 0 & 1200 & 2100 & 1300 \\ 4400 & 4800 & 3600 & 2700 & 4000 & 1400 & 0 & 800 & 2700 \\ 4100 & 4500 & 3300 & 2400 & 3700 & 1100 & 1300 & 0 & 2400 \end{pmatrix} \end{matrix}$$

$$D_b = \begin{matrix} & S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 \\ \begin{matrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \end{matrix} & \left( \begin{array}{cccccccc} 1000 & 1400 & 750 & 2500 & 1300 & 3100 & 3500 & 4400 \\ 1500 & 1000 & 1300 & 3000 & 1700 & 3500 & 3900 & 4700 \\ 750 & 1000 & 1000 & 2200 & 750 & 2600 & 3000 & 3800 \\ 2800 & 3000 & 2000 & 1000 & 2500 & 650 & 1100 & 1900 \\ 1400 & 1600 & 650 & 1600 & 1000 & 2100 & 2500 & 3300 \\ 2900 & 3100 & 2200 & 700 & 2600 & 1000 & 800 & 1700 \\ 3300 & 3500 & 2600 & 1000 & 3000 & 350 & 1000 & 800 \\ 4000 & 4200 & 3300 & 1800 & 3700 & 1100 & 800 & 1000 \end{array} \right) \end{matrix}$$

Pode-se notar que, em alguns casos, a distância percorrida de uma estação  $S_i$  para outra  $S_j$  é diferente da percorrida de  $S_j$  para  $S_i$ . Tal fato ocorre devido ao trânsito da Cidade Universitária, que, nesses casos, não permite a mesma rota de transporte em diferentes direções.

Para calcular o tempo que leva para percorrer os trajetos, é considerado que o veículo percorre o bairro a uma velocidade de  $40km/h$ , enquanto as bicicletas trafegam com uma velocidade de  $20km/h$ . Com as velocidades definidas, é possível calcular o tempo que leva para as bicicletas e o veículo trafegarem entre as construções do sistema dividindo as distâncias pelas velocidades definidas. Ambos os tempos resultantes são demonstrados, em segundos, nas matrizes  $Tt_c$  e  $Tt_b$ , que representam, respectivamente, os tempos que o veículo e as bicicletas levam para trafegar da construção da linha  $i$  para a construção da coluna  $j$ .

$$Tt_c = \begin{matrix} & S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 & C \\ \begin{matrix} C \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \end{matrix} & \left( \begin{array}{cccccccc} 261 & 297 & 180 & 0 & 225 & 108 & 99 & 171 & 0 \\ 0 & 288 & 189 & 387 & 216 & 432 & 423 & 486 & 387 \\ 135 & 0 & 117 & 315 & 153 & 360 & 351 & 423 & 315 \\ 76,5 & 108 & 0 & 234 & 67,5 & 279 & 270 & 342 & 234 \\ 261 & 297 & 180 & 0 & 225 & 108 & 99 & 171 & 0 \\ 135 & 171 & 58,5 & 180 & 0 & 234 & 225 & 297 & 180 \\ 270 & 306 & 198 & 117 & 234 & 0 & 108 & 189 & 117 \\ 396 & 432 & 324 & 243 & 360 & 126 & 0 & 72 & 243 \\ 369 & 405 & 297 & 216 & 333 & 99 & 117 & 0 & 216 \end{array} \right) \end{matrix}$$

$$T^t_b = \begin{matrix} & S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 \\ \begin{matrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \end{matrix} & \left( \begin{array}{cccccccc} 180 & 252 & 135 & 450 & 234 & 558 & 630 & 792 \\ 270 & 180 & 234 & 540 & 306 & 630 & 702 & 846 \\ 135 & 180 & 180 & 396 & 135 & 468 & 540 & 684 \\ 504 & 540 & 360 & 180 & 450 & 117 & 198 & 342 \\ 252 & 288 & 117 & 288 & 180 & 378 & 450 & 594 \\ 522 & 558 & 396 & 126 & 468 & 180 & 144 & 306 \\ 594 & 630 & 468 & 180 & 540 & 63 & 180 & 144 \\ 720 & 756 & 594 & 324 & 666 & 198 & 144 & 180 \end{array} \right) \end{matrix}$$

# Capítulo 6

## Formulações de algoritmos para serem utilizados no balanceamento do sistema do modelo proposto

O uso de um veículo de redistribuição em um SPCB é importante para o seu funcionamento por diminuir o tempo que as estações de bicicletas ficam vazias ou cheias, melhorando as experiências dos usuários com o sistema.

Porém, a ronda, que é realizada pelo veículo entre as estações em intervalos de tempo, não deve ser feita sem nenhum tipo de planejamento. Enquanto alguns SPCBs utilizam estratégias simples, que de fato melhoram o sistema, outros sistemas utilizam estratégias complexas, que melhoram ainda mais a eficiência do balanceamento pelo veículo.

Como o modelo em redes de Petri proposto neste trabalho permite a implementação de diferentes estratégias, neste capítulo serão apresentados as três estratégias: na seção 6.1, será apresentada a estratégia de balanceamento cíclica para o trajeto que o veículo realiza durante as rondas. Em seguida, na seção 6.2, será apresentado um algoritmo adaptativo que também decide a rota do veículo durante o balanceamento, porém ele se baseia no algoritmo de busca por força bruta. Finalmente, na seção 6.3, será apresentada uma estratégia de balanceamento preditiva, que altera os números de reordenação de acordo com situações futuras previstas no sistema.

### 6.1 Estratégia simples que realiza um balanceamento cíclico

Para garantir que o veículo de redistribuição visite todas as estações durante cada ronda, a estratégia de balanceamento cíclico consiste em traçar uma rota constante em que o veículo visita cada estação sequencialmente e retorna para o centro de

redistribuição após visitar a última estação.

O algoritmo mencionado foi implementado e avaliado na rede de Petri do trabalho de Antunes et al. [1, 2]. No trabalho, pôde-se concluir que, apesar da estratégia ser simples, ela diminui consideravelmente os tempos que as estações permanecem cheias ou vazias durante o funcionamento do sistema. Assim, esse sucesso motivou a implementação da mesma estratégia no modelo deste trabalho.

No modelo proposto, quando o veículo de redistribuição está no centro de redistribuição, sua próxima estação de destino é definida através da escolha de uma das transições  $T_{CarInd_{C_j}}$  para disparar, indicando que a estação  $S_j$  foi escolhida como destino. Caso ele esteja em alguma estação  $S_i$ , é escolhido uma das estações  $S_j$  como destino, através de  $T_{CarInd_{i_j}}$ , ou o centro de redistribuição, pelo disparo de  $T_{CarInd_{i_C}}$ , o que finaliza a ronda. Na estratégia cíclica, as ações dos agentes externos de iniciar a ronda, quando  $m(PCar_C) = 1$ , e de escolher a estação seguinte para balancear, quando  $m(PE_i) = 1$ , são feitas atribuindo os valores às funções externas:

$$Esc(T_{CarInd_{C_1}}, PCar_C) = 1; \quad (6.1)$$

$$Esc(T_{CarInd_{(i)(i+1)}}, PE_i) = 1, \text{ para } i \in \{1, 2, \dots, 7\}; \quad (6.2)$$

$$Esc(T_{CarInd_{8C}}, PE_8) = 1. \quad (6.3)$$

Com as funções (6.1), (6.2) e (6.3), pode-se concluir que, no começo da ronda, o veículo se move para a estação  $S_1$ , depois visita a estação com a numeração seguinte, até quando o veículo chega à estação  $S_8$ , em que o veículo então possui como destino final o centro de redistribuição. A movimentação completa do veículo pode ser visualizada na sub-rede do modelo da figura 6.1, em que são apresentados os lugares e as transições que disparam com o uso do algoritmo.

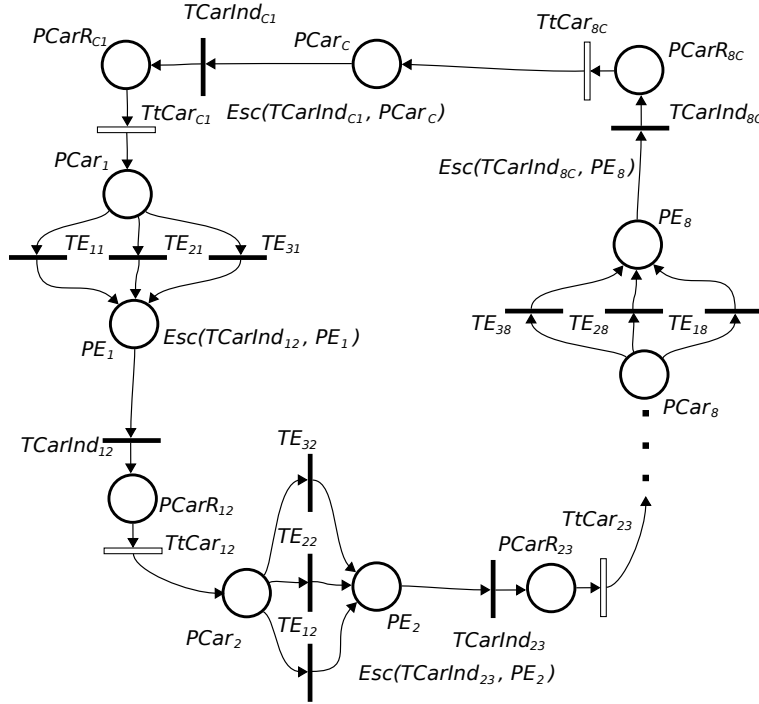


Figura 6.1: Sub-rede demonstrando o trajeto do veículo na estratégia cíclica.

Apesar do algoritmo garantir que o veículo visite todas as estações em cada ronda, caso ele visite uma estação com o número de bicicletas abaixo do número de reordenação enquanto está vazio ou visite uma estação com o número de bicicletas acima do número de reordenação enquanto está cheio, o veículo não realizará nenhum balanceamento na estação. Isto acontece pois, no primeiro caso, de acordo com a sub-rede de controle de estações explicada na seção 3.2.2, no momento em que o veículo chega à uma estação  $S_i$ , a transição  $TE_{2i}$  dispara imediatamente, indicando que não houve balanceamento na estação. O mesmo acontece no segundo caso, em que  $TE_{1i}$  dispara imediatamente.

Além disso, durante a estratégia cíclica, o veículo pode visitar estações que não precisem de balanceamento por possuírem números razoáveis de bicicletas. Como o trajeto é feito sem considerar o estado atual das estações, em alguns momentos do balanceamento o veículo pode dirigir-se à uma estação que precise de pouca ou nenhuma movimentação de bicicletas para que o seu número de bicicletas se torne igual ao número de reordenação, desperdiçando a viagem que realizou para chegar à estação.

## 6.2 Estratégia adaptativa baseado no algoritmo de busca por força bruta

Com o intuito de evitar que o veículo não visite estações que não precisem ser balanceadas ou que não podem trocar bicicletas com o veículo, é necessário formular uma estratégia que aja de acordo com os estados das estações, ou seja, o número de bicicletas e reordenação  $R_i$  de cada estação  $S_i$ , no momento da escolha do próximo destino do veículo.

A partir desse pensamento, inicialmente, é proposto que, no momento da decisão da próxima estação que o veículo deve visitar, é avaliado qual escolha é a mais benéfica para o sistema. A análise do quão benéfico uma decisão é para o sistema é feita utilizando o indicador  $DB$ , que representa o total de movimentos de bicicletas que devem ser realizados para estabilizar completamente a quantidade de bicicletas em cada estação com seus respectivos números de reordenação. O indicador  $DB$  é descrito como:

$$DB = \sum_{i=1}^8 |m(PBS_i) - R_i| \quad (6.4)$$

O valor de  $DB$  representa o quanto o sistema está instável, ou seja, o quanto as quantidades de bicicletas das estações estão distantes dos seus respectivos números de reordenação. Portanto, a rota de balanceamento realizada pelo veículo deve ter como prioridade reduzir o valor de  $DB$ .

Para prever essa prioridade, uma primeira possibilidade de um algoritmo de estratégia adaptativa é fazer uma busca por força bruta, apresentada na seção 2.6, para verificar a próxima estação que o veículo deve visitar para estabilizar o máximo possível o sistema. Ele faz isso simulando, dentro de um algoritmo externo, o que aconteceria com o número de bicicletas das estações se o veículo visitasse cada uma delas. Com cada possibilidade simulada, é calculado o valor de  $DB$  para cada situação, para, com esses resultados, escolher a estação que causou o menor valor de  $DB$  para ser o novo destino do veículo.

Ainda que esse primeiro algoritmo proposto execute o melhor balanceamento da primeira estação que o veículo visita, as rondas realizadas consistem de várias viagens. Portanto, é necessário que o algoritmo considere as estações que o veículo visitará após outras visitas no momento em que escolhe o próximo destino. Isso pode ser feito tornando a busca por força bruta mais robusta ao verificar todas as possibilidades de rotas que envolvam mais de uma viagem entre as estações.

Aumentar a robustez da busca, entretanto, acrescenta consideravelmente o número de possibilidades que são estudadas. Para um SPCB com  $N$  estações, os números  $PT(N, j)$  de possíveis trajetos entre estações que o veículo pode realizar,

do centro de redistribuição, em  $j \in \{1, 2, \dots, J\}$  viagens, em que  $J$  é o número máximo de viagens que o veículo ainda pode realizar na ronda antes de voltar para o centro de redistribuição, é:

$$PT(N, j) = N(N - 1)^{j-1}$$

No caso do sistema do Integra UFRJ, que possui 8 estações, a busca por força bruta utilizando  $j = 8$  viagens resultaria em  $PT(8, 8) = 6588344$  possibilidades. Encontrar, dentro desse grande número de possibilidades, a que realiza o melhor balanceamento da estação consumiria tempo demais, atrasando a execução do balanceamento.

Outro problema causado por realizar a busca para várias viagens de uma ronda é que, durante as viagens realizada pelo veículo entre as estações, a quantidade de bicicletas em cada estação pode alterar consideravelmente, pois os usuários estão constantemente removendo e adicionando bicicletas às estações.

Para que o algoritmo evite os problemas mencionados, a avaliação dos possíveis trajetos, por meio de uma busca por força bruta, é feita considerando apenas duas viagens. Essa escolha resulta em um número relativamente pequeno de possibilidades de trajetos  $PT(8, 2) = 56$ , que aumenta polinomialmente conforme  $N$  aumenta, pois  $PT(N, 2) = N^2 - N$ . Como a equação é polinomial, a busca por força bruta não seria demorada para  $N$  maiores. Para um sistema com vinte estações, por exemplo,  $PT(20, 2) = 380$ .

Finalmente, deve ser escolhido, para este algoritmo, quando o veículo deve retornar para o centro de redistribuição. Para facilitar a comparação com a estratégia de balanceamento cíclico apresentada na seção 6.1, o número de viagens realizadas em cada ronda não pode superar o número de estações, no caso do sistema do integra UFRJ, esse número é oito. Além disso, para limitar o consumo máximo de combustível ou outra fonte de energia para mover o veículo por ronda, caso ele percorra uma distância superior a uma estipulada, ele deve retornar para o centro de redistribuição. É considerado, para o integra UFRJ, que o veículo não pode percorrer mais do que  $25km$ .

### 6.3 Estratégia preditiva que varia os números de reordenação de acordo com a demanda

Ambas as estratégias apresentadas anteriormente influenciam nos trajetos realizados pelo veículo e, no caso da segunda, escolhe os trajetos a partir de informações atuais do sistema. A fim de compor uma estratégia que, diferente das outras, use informações de possíveis estados futuros do sistema para alterar diretamente a ação



de balanceamento de cada estação, é proposta uma estratégia preditiva que varia o número de reordenação de cada estação de acordo com a demanda.

Em um sistema em que a demanda por bicicletas durante o dia é constante, poderia-se supor que o número ideal de bicicletas em cada estação seria metade da capacidade total dela, ou seja, para cada estação  $S_i$ ,  $R_i = C_i/2$ . Tal suposição, porém, costuma ser inválida para sistemas reais, cujos usos podem variar em diferentes horários do dia.

Foi visto na seção 5.3 que o sistema do Integra UFRJ possui picos de utilização do sistema em diferentes horários do dia, como, por exemplo, no horário do almoço, em que os usuários se deslocam dos principais campus, CT/CCMN e Letras, para almoçarem. Pode-se concluir que o ideal para essa situação é que as quantidades de bicicletas nas estações mencionadas seja grandes enquanto as quantidades de bicicletas das estações de destino, como o Restaurante Universitário, diminuam. Assim, durante os picos, deve-se aumentar os números de reordenação das estações do pico e reduzir os das estações de destino.

Quando uma estação  $S_i$  não está envolvida em um pico, o seu número de reordenação é a metade da sua capacidade total, porém, no momento de pico na estação, esse número é aumentado para 90% da capacidade total da estação. Se, por exemplo, a estação do Restaurante Universitário  $S_3$ , com capacidade 14, não estiver em uma situação de pico, seu número de reordenação é  $R_3 = 7$ , caso ela esteja em um momento de pico, esse valor muda para  $R_3 = 12,6$ . Como o número de reordenação deve ser um inteiro, o valor aproximado que é considerado durante o balanceamento é  $R_3 = 13$ .

Além de aumentar o número de reordenação da estação com o pico, deve-se reduzir os números de reordenação das suas estações de destino. Essa redução é feita utilizando os dados descritos na matriz  $DIDA$  que, conforme descrito na seção 5.3, representa a probabilidade do usuário que retira uma bicicleta da estação da linha  $i$  escolher a estação da coluna  $j$  como destino. Deve-se lembrar que os elementos dessa matriz podem ser representados por  $\lambda_{ij}$ . Assim, para cada estação  $S_j$  que não possua um pico no momento, mas que seja um possível destino de uma estação  $S_i$  com pico, seu novo número de reordenação  $R'_j$  é reduzido de acordo com a equação:

$$R'_j = R_j - Rpa_i \lambda_{ij},$$

em que  $Rpa_i$  representa o valor que foi adicionado à  $R_i$  devido ao pico. Então, no exemplo da estação do Restaurante Universitário  $S_3$ , o valor seria  $Rpa_3 = 13 - 7 = 6$ . Como o valor do número de reordenação deve ser um inteiro positivo, caso a equação resulte em um valor menor do que zero, é considerado que  $R_j = 0$ .

Finalmente, deve-se definir quando os números de reordenação devem ser modi-

ficados em relação aos horários de picos. Como as rondas começam em intervalos fixos de tempo, é recomendável realizar pelo menos uma ronda com os números de reordenação modificados antes do horário de pico. Por isso, os números de reordenação são modificados com o mesmo intervalo de tempo que as rondas, antes da ocorrência do pico, até o horário em que o pico acontece. Então se, por exemplo, ocorrer um pico às 12:00h em um sistema que realiza as rondas de duas em duas horas, os números de reordenação são alterados, em relação a esse pico, de 10:00h até 12:00h.

## Capítulo 7

# Implementação e avaliação do algoritmo de simulação do modelo de um sistema público de compartilhamento de bicicletas com $N$ estações

Com o modelo de SPCB, as principais características do Integra UFRJ e estratégias de balanceamento definidos, é possível implementar um algoritmo capaz de simular o funcionamento do sistema e dos diferentes tipos de estratégias para avaliá-los.

Apesar de, neste trabalho, só ser simulado o sistema do projeto Integra UFRJ, para facilitar trabalhos futuros, o simulador é feito para simular qualquer SPCB com diferentes quantidades  $N$  de estações.

Para apresentar as características da simulação e seus resultados, este capítulo é estruturado da seguinte forma. Inicialmente, na seção 7.1, será apresentado o funcionamento e principais características do simulador. Em seguida, na seção 7.2, os parâmetros de entrada necessários para a simulação serão apresentados juntos com os valores que são utilizados neste trabalho. Após os parâmetros de entrada, na seção 7.3, serão apresentados os indicadores que são calculados durante a simulação, necessários para, posteriormente, avaliar a eficiência de cada estratégia e compará-las. Finalmente, na seção 7.4, serão apresentados, avaliados e comparados os resultados das simulações de cinco tipos diferentes de combinações de estratégias.

## 7.1 Algoritmo de simulador

Como redes de Petri estocásticas com controles externos, definidos na seção 2.5, foram propostos para atender às necessidades do modelo para SPCB descrito no capítulo 4, não existem simuladores prontos para esse tipo de rede de Petri. Portanto, é preciso implementar um novo algoritmo específico para esse tipo de rede.

Para implementar um algoritmo para simular o modelo, é utilizado um software voltado para cálculos numéricos, o MATLAB. Ele permite, através da composição de diferentes linhas de códigos em grupos, chamados de *scripts*, a simulação e visualização de diferentes tipos de algoritmos computacionais.

Inicialmente, através dos blocos de matrizes descritos na seção 4.2, o simulador compõe as matrizes *Pre* e *Post*, de dimensões  $n_P \times n_T$ , que representa um sistema com  $N$  estações de bicicletas. Em seguida, é feito um vetor  $\underline{m}_{SPCB}$ , com  $n_P$  elementos que representam as quantidades de fichas em cada lugar da rede ordenados conforme  $p$ , que também foi descrito na seção 4.2.

Apesar da maioria das marcações iniciais dos lugares serem nulas, algumas começam com valores diferentes de zero, por isso, devem ser adicionadas fichas a elas antes de simular a dinâmica da rede. Primeiro, são adicionadas as fichas que representam as quantidades de bicicletas em cada estação. Para isso, supondo que tenha sido escolhido o valor inicial de  $NbIni_i$  bicicletas para cada estação  $S_i$ , são adicionadas  $NbIni_i$  fichas à marcação  $m(PBS_i)$  e  $C_i - NbIni_i$  à  $m(PES_i)$ . Além disso, devem ser adicionadas  $N_B - C_i$  fichas ao lugar  $PInhib_i$  de cada estação  $S_i$ . Assim, Para cada estação  $S_i$ , os seguintes valores são inicialmente atribuídos à  $\underline{m}_{SPCB}$ :

$$\begin{aligned}\underline{m}_{SPCB}((i-1)(8+2N)+2) &= N_B - C_i; \\ \underline{m}_{SPCB}((i-1)(8+2N)+4) &= C_i - NbIni_i; \\ \underline{m}_{SPCB}((i-1)(8+2N)+5) &= NbIni_i.\end{aligned}$$

Além das marcações dos lugares das estações, alguns lugares relacionados ao veículo de redistribuição e o centro de redistribuição também possuem marcações diferentes de zero. O lugar  $PCar_C$  deve possuir uma ficha para indicar que o veículo se encontra em sua posição inicial, ou seja, o centro de redistribuição. Assim como as estações, o veículo também possui lugares que representam as quantidades de bicicletas e vagas no veículo, que são, respectivamente  $PBSCar$  e  $PESCar$ . Considerando que  $NbIni_v$  representa o número inicial de bicicletas no veículo, os valores iniciais que também devem ser atribuídos à  $\underline{m}_{SPCB}$  são:

$$\underline{m}_{SPCB}(N(8+2N)+1) = 1;$$

$$\begin{aligned}\underline{m}_{SPCB}(N(8 + 2N) + N + 2) &= NbIni_v; \\ \underline{m}_{SPCB}(N(8 + 2N) + N + 3) &= C_v - NbIni_v.\end{aligned}$$

Após definir a marcação inicial da rede, é descrita a dinâmica da rede. Para representar uma escala de tempo real, o programa simula cada segundo do sistema. Isso significa que cada passo, em que ocorrem os disparos das transições, simula 1s do sistema.

Como a estratégia de balanceamento apresentada na seção 6.3 altera o número de reordenação  $R_i$  de cada estação  $S_i$  durante o funcionamento do sistema, o simulador permite a modificação dos números de reordenação em determinados momentos do sistema, que ocorrem entre intervalos fixos. Quando essa modificação é realizada, as matrizes  $Pre$  e  $Post$  são recalculadas com os novos números.

Com as matrizes definidas, o simulador verifica quais transições devem disparar em um determinado segundo, ou seja, em um passo  $k$  da simulação. Nesse passo, é definido um vetor  $\sigma_k$ , de comprimento  $n_T$ , cujos elementos representam as transições do sistema e são ordenados de acordo com  $\underline{t}$ , apresentado na seção 4.2. Caso o elemento em uma posição  $j$  do vetor seja um, significa que, no passo  $k$ , a transição desse elemento deve disparar uma vez, gerando uma nova marcação  $\underline{m}'_{SPCB}$  através da equação:

$$\underline{m}'_{SPCB} = \underline{m}_{SPCB} + (Post - Pre)\sigma_k$$

Nota-se que, caso o valor de um dos elementos seja maior que um, a transição dispara mais de uma vez.

Apesar do cálculo da dinâmica do sistema ser simples dado  $\sigma_k$ , definir completamente  $\sigma_k$  não é trivial, pois existem conflitos entre transições e transições temporizadas, cujos disparos não ocorrem apenas com suas habilitações. A fim de padronizar o disparo dos casos contidos na rede, as transições são divididas entre os grupos: transições imediatas que não estão envolvidas em conflitos; transições temporizadas estocásticas; transições imediatas que estão envolvidas em conflitos solucionados estocasticamente; transições imediatas que estão envolvidas em conflitos solucionados por meio de um controle externo.

### 7.1.1 Transições imediatas que não estão envolvidas em conflitos

A verificação da habilitação de qualquer transição imediata é simples: verificar se a quantidade de fichas dos lugares de entrada são maiores ou iguais aos pesos dos respectivos arcos. Portanto, para verificar se uma transição qualquer  $t_j$ , que representa a  $j$ -ésima coluna da matriz  $Pre$ , está habilitada, basta subtrair a marcação

atual da  $j$ -ésima coluna de  $\underline{m}_{SPCB}$  pela  $j$ -ésima coluna da matriz  $Pre$  e verificar se o valor de algum elemento da subtração é negativo. Caso o resultado possua algum valor negativo, significa que a marcação não possui fichas suficientes para disparar a transição, portanto, o valor  $\sigma_k(j)$  é mantido como zero, caso contrário, a transição deve disparar, tornando o valor de  $\sigma_k(j)$  igual a um.

As transições que estão contidas neste grupo são, para cada estação  $S_i$ :  $TIn_i$ ,  $TWIn_i$ ,  $TAdd_i$ ,  $TRem_i$ ,  $TE_{1i}$ ,  $TE_{2i}$ ,  $TE_{3i}$ .

### 7.1.2 Transições temporizadas estocásticas

Como todas as transições temporizadas da rede são estocásticas, é atribuída, para cada transição temporizada  $t_j$ , uma função  $fdp_{t_j}$   $\Lambda_d(\cdot)$  para sortear valores, em segundos, do tempo que leva para as transições dispararem após se tornarem habilitadas. Para essa rede, todas as funções das transições temporizadas são uniformes, ou seja, para definir o tempo da transição, é escolhido um valor aleatório dentro de um intervalo fixo. Os pontos médios desses intervalos foram definidos na seção 5.4, em que a matriz  $Tt_c$  representa os tempos médios que levam para as transições  $TtCar_{ij}$ , tal que  $i, j \in \{C, 1, 2, \dots, 8\}$  e  $i \neq j$ , dispararem após suas habilitações, e a matriz  $Tt_b$ , que representa os tempos médios das transições  $Tt_{ij}$ , tal que  $i, j \in \{1, 2, \dots, 8\}$ .

Para encontrar os valores mínimos e máximos de tempo que cada função pode alcançar, valores necessários para defini-las completamente, é considerado que as bicicletas e o veículo podem se mover até  $10Km/h$  mais rapidamente ou mais lentamente do que seus respectivos pontos médios. Ambos valores resultam, respectivamente e na escala de tempo, nas matrizes mínimas  $TtMin_b$ ,  $TtMin_c$  e nas matrizes máximas  $TtMax_b$ ,  $TtMax_c$ , cujos valores são:

$$TtMin_b = \begin{matrix} & S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 \\ \begin{matrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \end{matrix} & \left( \begin{array}{ccccccccc} 120 & 168 & 90 & 300 & 156 & 372 & 420 & 528 \\ 180 & 120 & 156 & 360 & 204 & 420 & 468 & 564 \\ 90 & 120 & 120 & 264 & 90 & 312 & 360 & 456 \\ 336 & 360 & 240 & 120 & 300 & 78 & 132 & 228 \\ 168 & 192 & 78 & 192 & 120 & 252 & 300 & 396 \\ 348 & 372 & 264 & 84 & 312 & 120 & 96 & 204 \\ 396 & 420 & 312 & 120 & 360 & 42 & 120 & 96 \\ 480 & 504 & 396 & 216 & 444 & 132 & 96 & 120 \end{array} \right) \end{matrix}$$

$$TtMin_c = \begin{matrix} & S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 & C \\ C & \left( \begin{array}{c} 209 \\ 0 \\ 108 \\ 61 \\ 209 \\ 108 \\ 216 \\ 317 \\ 295 \end{array} \right. & \left( \begin{array}{c} 238 \\ 230 \\ 0 \\ 86 \\ 238 \\ 137 \\ 245 \\ 346 \\ 324 \end{array} \right. & \left( \begin{array}{c} 144 \\ 151 \\ 94 \\ 0 \\ 144 \\ 47 \\ 158 \\ 259 \\ 238 \end{array} \right. & \left( \begin{array}{c} 0 \\ 310 \\ 252 \\ 187 \\ 0 \\ 144 \\ 94 \\ 194 \\ 173 \end{array} \right. & \left( \begin{array}{c} 180 \\ 173 \\ 122 \\ 54 \\ 180 \\ 0 \\ 187 \\ 288 \\ 266 \end{array} \right. & \left( \begin{array}{c} 86 \\ 346 \\ 288 \\ 223 \\ 86 \\ 187 \\ 0 \\ 101 \\ 79 \end{array} \right. & \left( \begin{array}{c} 79 \\ 338 \\ 281 \\ 216 \\ 79 \\ 180 \\ 86 \\ 0 \\ 94 \end{array} \right. & \left( \begin{array}{c} 137 \\ 389 \\ 338 \\ 274 \\ 137 \\ 238 \\ 151 \\ 58 \\ 0 \end{array} \right. & \left. \begin{array}{c} 0 \\ 310 \\ 252 \\ 187 \\ 0 \\ 144 \\ 94 \\ 194 \\ 173 \end{array} \right)$$

$$TtMax_b = \begin{matrix} & S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 \\ S_1 & \left( \begin{array}{c} 360 \\ 540 \\ 270 \\ 1008 \\ 504 \\ 1044 \\ 1188 \\ 1440 \end{array} \right. & \left( \begin{array}{c} 504 \\ 360 \\ 360 \\ 1080 \\ 576 \\ 1116 \\ 1260 \\ 1512 \end{array} \right. & \left( \begin{array}{c} 270 \\ 468 \\ 360 \\ 720 \\ 234 \\ 792 \\ 936 \\ 1188 \end{array} \right. & \left( \begin{array}{c} 900 \\ 1080 \\ 792 \\ 360 \\ 576 \\ 252 \\ 360 \\ 648 \end{array} \right. & \left( \begin{array}{c} 468 \\ 612 \\ 270 \\ 900 \\ 360 \\ 936 \\ 1080 \\ 1332 \end{array} \right. & \left( \begin{array}{c} 1116 \\ 1260 \\ 936 \\ 234 \\ 756 \\ 360 \\ 126 \\ 396 \end{array} \right. & \left( \begin{array}{c} 1260 \\ 1404 \\ 1080 \\ 396 \\ 900 \\ 288 \\ 360 \\ 288 \end{array} \right. & \left. \begin{array}{c} 1584 \\ 1692 \\ 1368 \\ 684 \\ 1188 \\ 612 \\ 288 \\ 360 \end{array} \right)$$

$$TtMax_c = \begin{matrix} & S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 & C \\ C & \left( \begin{array}{c} 348 \\ 0 \\ 180 \\ 102 \\ 348 \\ 180 \\ 360 \\ 528 \\ 492 \end{array} \right. & \left( \begin{array}{c} 396 \\ 384 \\ 0 \\ 144 \\ 396 \\ 228 \\ 408 \\ 576 \\ 540 \end{array} \right. & \left( \begin{array}{c} 240 \\ 252 \\ 156 \\ 0 \\ 240 \\ 78 \\ 264 \\ 432 \\ 396 \end{array} \right. & \left( \begin{array}{c} 0 \\ 516 \\ 420 \\ 312 \\ 0 \\ 240 \\ 156 \\ 324 \\ 288 \end{array} \right. & \left( \begin{array}{c} 300 \\ 288 \\ 204 \\ 90 \\ 300 \\ 0 \\ 312 \\ 480 \\ 444 \end{array} \right. & \left( \begin{array}{c} 144 \\ 576 \\ 480 \\ 372 \\ 144 \\ 312 \\ 0 \\ 168 \\ 132 \end{array} \right. & \left( \begin{array}{c} 132 \\ 564 \\ 468 \\ 360 \\ 132 \\ 300 \\ 144 \\ 0 \\ 156 \end{array} \right. & \left( \begin{array}{c} 228 \\ 648 \\ 564 \\ 456 \\ 228 \\ 396 \\ 252 \\ 96 \\ 0 \end{array} \right. & \left. \begin{array}{c} 0 \\ 516 \\ 420 \\ 312 \\ 0 \\ 240 \\ 156 \\ 324 \\ 288 \end{array} \right)$$

A partir das funções  $\Lambda_d(\cdot)$  definidas, torna-se possível criar uma lista que, dada a habilitação de uma transição temporizada, adiciona à si um elemento contendo o identificador da transição e o tempo que foi sorteado através da função. A cada passo do simulador, o valor de todos os tempos dos elementos da lista são reduzidos por um segundo. Conforme os elementos dessa lista se tornam menores ou iguais a zero, as transições relativas aos elementos disparam, através da soma do valor do elemento da transição em  $\sigma_k$  por um, e o elemento é removido da lista.

Pode-se notar que, além de ser possível a lista possuir mais de um elemento relativo à mesma transição, esses dois elementos podem disparar simultaneamente. Quando isso acontece, o valor do elemento da transição em  $\sigma_k$  se torna maior do que um, causando mais de um disparo da transição no passo  $k$ .

### 7.1.3 Transições imediatas que estão envolvidas em conflitos solucionados estocasticamente

Assim como definido na seção 2.4, conflitos entre transições imediatas podem ser solucionados através de um sorteio realizado utilizando probabilidades definidas para cada transição através de  $\Lambda_i(\cdot)$ . Na rede do modelo proposto neste trabalho, para cada estação  $S_i$ , dois grupos de conflitos são solucionados estocasticamente: o conflito entre cada transição  $TInd_{ij}$ , que representa a ação do usuário de escolher uma estação  $S_j$  como destino, e o conflito entre  $T_{ij}$ , representando o conflito do usuário de escolher  $S_j$  como próximo destino ao chegar em uma estação que está cheia. Os valores de  $\Lambda_i(TInd_{ij}, PR_i)$  e  $\Lambda_i(T_{ij}, PIn_i)$  são iguais, respectivamente, aos elementos das linhas  $i$  e colunas  $j$  das matrizes  $DIDA$  e  $DREIDA$ , descritas na seção 5.3.

Para realizar o sorteio entre um grupo de transições, como por exemplo a escolha no conflito  $C(PR_1, \{TInd_{12}, TInd_{13}, \dots, TInd_{18}\})$ , primeiro é feita uma soma cumulativa dos valores  $\Lambda_i(TInd_{1j}, PR_1)$ , para  $j \in \{1, 2, \dots, 8\}$ , de tal forma que resulte em um vetor ordenado, que, no exemplo, é igual a  $[0,0004 \ 0,0842 \ 0,3485 \ 0,6227 \ 0,6564 \ 0,8487 \ 0,9513 \ 1]$ . Após isso, é sorteado um número aleatório de 0 a 1, cujo valor indica que a posição do primeiro elemento, da esquerda para direita, que é maior do que o número aleatório representa qual das transições deve disparar. Se o número sorteado fosse 0,65, a posição escolhida no vetor seria a quinta, representando que a transição escolhida para disparar seria  $TInd_{15}$ . O mesmo tipo de sorteio se aplica às transições  $T_{ij}$ .



### 7.1.4 Transições imediatas que estão envolvidas em conflitos solucionados por meio de um controle externo

Por se tratar de uma rede de Petri estocástica com controle externo, alguns dos conflitos entre transições imediatas da rede podem ser resolvidos através de um agente externo. Neste trabalho existem dois tipos de agentes externos:  $Esc(TOut_i, PBS_i)$ , que representa a ação do usuário alugar uma bicicleta em uma estação  $S_i$ ;  $Esc(TCarInd_{ij}, PE_i)$  ou  $Esc(TCarInd_{Ci}, PCar_C)$ , tal que  $i \in \{1, 2, \dots, N\}$  e  $j \in \{C, 1, 2, \dots, N\}$ , representando a decisão de mover o veículo de redistribuição entre os locais do sistema.

Durante a inicialização do primeiro agente externo, que avalia o disparo das transições  $TOut_i$  de cada estação  $S_i$ , é definido quando cada um dos usuários tentam alugar uma bicicleta em cada estação em cada passo da simulação através de uma matriz  $Tim_u$ , de dimensões  $N \times Ts$ , que representa a chegada de um usuário na estação da linha  $i$  no tempo, em segundos, da coluna  $j$ , em que  $Ts$  é a duração de tempo total da simulação, que vale 57600s no sistema da integra UFRJ. Para realizar isso, deve-se dividir o número de pessoas que usam cada estação durante o dia, valores contidos no vetor  $N_{pe}$  e definido na seção 5.3, em números de pessoas que retiram bicicletas das estações durante os horários de pico, representado pelo vetor  $N_p$ , e números de pessoas que alugam bicicletas normalmente ao longo do dia, representado por  $N_u$ .

Para alocar os usuários representados em  $N_u$ , é feito, para cada um, um sorteio uniforme de um número  $j$  aleatório de 1 até  $Ts$ , que representa o momento em que ele tenta retirar uma bicicleta. Enquanto a estação  $S_i$  representa a linha  $i$ , o número sorteado representa a coluna  $j$  do elemento da matriz  $Tim_u$  cujo valor é alterado para um. Caso dois sorteios resultem no mesmo tempo para dois usuários da mesma estação, o segundo é alocado para um tempo  $j - 1$  ou  $j + 1$ , o módulo dessa adição e subtração aumentam até que o usuário seja alocado em um espaço vazio de  $Tim_u$ , elas sejam menores do que um ou maiores do que  $Ts$ .

O passo para alocar os usuários de  $N_p$  é semelhante ao de  $N_u$ , porém, o sorteio é realizado através da distribuição de Poisson ( $fdp_{poiss}(\cdot, \cdot)$ ), em que a probabilidade de cada número  $k$ , entre o intervalo de 1 a  $Ts$ , ser escolhido é, em função da constante  $\lambda$ , que representa os horários de pico:

$$fdp_{poiss}(x, k) = \frac{\exp^{-\lambda} \lambda^k}{k!}$$

Como exemplificado na figura 7.1, a distribuição atribuí à cada inteiro do intervalo uma probabilidade que aumenta conforme se aproxima de  $\lambda$ . Deve-se notar que a soma de todas essas probabilidades resulta em um. Portanto, é possível realizar um sorteio, semelhante ao apresentado na seção 7.1.2, a partir dessas probabilidades.

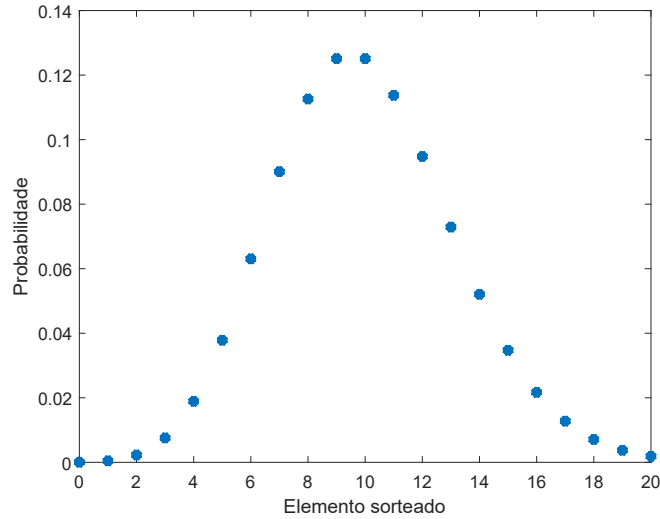


Figura 7.1: Gráfico da distribuição de Poisson para um intervalo de 0 a 20 e  $\lambda = 10$ .

Esse tipo de distribuição é conveniente para distribuir os usuários dos picos, pois é desejado que ocorra o maior uso desses usuários durante os horários de picos, apresentados na seção 5.3, que são igualados ao  $\lambda$  da equação de distribuição. Assim, para cada pico de cada estação, é realizado esse sorteio, alocando usuários na matriz  $Tim_u$  ao alterar o valor do elemento da estação do pico e tempo sorteado para um.

Caso uma estação tenha mais de um pico, o valor de  $N_p$  é dividido igualmente entre os picos para depois serem realizados os sorteio e, semelhante à distribuição de  $N_u$ , se tiverem usuários alocados nos horários que foram sorteados, estes são alocados um segundo antes ou um segundo depois.

Com os horários de entrada no sistema dos usuários definidos, é verificado, em cada passo  $k$  da simulação e para cada estação  $S_i$ , se  $Tim_u(i, k)$  é igual a um. Caso a verificação seja positiva, o simulador tenta disparar  $TOut_i$ . Porém, Caso a transição não esteja habilitada por não ter bicicletas na estação, esse usuário é movido para uma fila, com um tamanho fixo e determinado, onde ele aguarda até que  $TOut_i$  esteja habilitada para disparar, reduzindo a quantidade de usuários na fila por um. Se um usuário chegar na estação enquanto sua fila está cheia, é considerado que um dos usuários desistiu de esperar por uma bicicleta, logo, ele não aluga uma bicicleta.

Outro conflito resolvido por controle externo é o da escolha da rota do veículo entre as estações e o centro de redistribuição, representada pelas transições  $TCarInd_{ij}$ , em que  $i, j \in \{C, 1, 2, \dots, 8\}$  e  $i \neq j$ . Em um momento escolhido do sistema, o controle externo  $Esc(TCarInd_{C_i}, PCar_C)$  realiza o disparo de uma das transições  $TCarInd_{C_i}$ , movendo o veículo para uma estação  $S_j$  de acordo com um dos algoritmos apresentados no capítulo 6 e iniciando uma ronda, em que todas as rotas são definidas pelo algoritmo externo através das funções  $Esc(TCarInd_{ij}, PE_i)$

e realizadas pelos disparos das transições  $T_{CarInd_{ij}}$ . Após o fim da ronda, o veículo aguarda, no centro de redistribuição, a passagem de um intervalo de tempo fixo após o começo da ronda anterior. Depois da espera, uma nova ronda começa, rebalanceando o sistema novamente.

## 7.2 Parâmetros de entrada do simulador

O simulador, que contém os agentes externos que resolvem conflitos, utiliza diferentes parâmetros de entrada para que seja possível implementar diferentes SPCBs em diferentes situações. Para apresentar essas variáveis, elas são nomeadas e descritas a seguir:

- **Ts**: Duração de tempo total da simulação, em segundos. A variável representa também o número de passos simulados. O tempo total simulado para o sistema Integra UFRJ é de 16h, ou seja, 57600s.

- **init<sub>T</sub>**: Tempo inicial do sistema, em horas. Apesar de todos os cálculos realizados no simulador considerarem  $k = 1$  como o tempo inicial, esta variável é utilizada para compor gráficos que ilustram o funcionamento do sistema utilizando horários em que o sistema opera durante o dia. O valor dessa variável para o Integra UFRJ de é 6:00h.

- **N**: Número total de estações do SPCB a ser simulado, cujo valor para o Integra UFRJ é oito.

- **P<sub>Cheg</sub>**: Matriz de dimensões  $N \times (P_{Max} + 1)$  que contém os horários, em segundos, de todos os picos presentes nas estações, representadas pelas linhas da matriz. Como as estações podem possuir diferentes quantidades de picos durante o dia, alguns elementos na matriz não representam um pico da estação. Como esses elementos não representam nada, eles são determinados com o valor  $-1$ , que também é utilizado na última coluna da matriz para representar o fim dela. De acordo com os picos apresentados na seção 5.3, a matriz  $P_{Cheg}$  do Integra UFRJ é definida, em segundos, como:

$$P_{Cheg} = \begin{pmatrix} 7200 & -1 & -1 \\ -1 & -1 & -1 \\ 25200 & -1 & -1 \\ 21600 & 39600 & -1 \\ -1 & -1 & -1 \\ 21600 & 39600 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

- $\mathbf{N}_u$  e  $\mathbf{N}_p$ : Vetores de dimensão  $N$  cujos elementos da posição  $i$  representam, respectivamente, a quantidade de usuários que alugam bicicletas ao longo do dia e usuários que alugam bicicletas durante os picos da estação  $S_i$ . Para obter-se os vetores, os valores dos elementos do vetor  $N_{pe}$  são divididos entre os dois. Como nem todas as estações possuem picos, essa divisão deve ser feita por estação. Para cada estação  $S_i$ , se ela não estiver envolvida em um pico,  $N_u(i) = N_{pe}(i)$  e  $N_p = 0$ . Do contrário, 35% de  $N_{pe}(i)$  é atribuído à  $N_p(i)$ , enquanto o resto é atribuído à  $N_u(i)$ . No caso do Integra UFRJ, portanto, os valores gerados de  $N_u$  e  $N_p$  são:

$$N_u = [36 \ 23 \ 45 \ 50 \ 13 \ 34 \ 31 \ 13]^T$$

$$N_p = [20 \ 0 \ 24 \ 27 \ 0 \ 19 \ 0 \ 0]^T$$

- **DIda**: Matriz de dimensões  $N \times N$  em que cada elemento representa a chance do usuário que retira uma bicicleta da estação da linha  $i$  escolher como destino a estação da coluna  $j$ . A matriz *DIda* do Integra UFRJ foi definida na seção 5.3.

- **DReIda**: Matriz de dimensões  $N \times N$  em que cada elemento representa a chance do usuário, após chegar em uma estação da linha  $i$  e encontrá-la cheia, escolher como destino a estação da coluna  $j$ , se  $i \neq j$ , ou escolher aguardar por uma vaga na estação, se  $i = j$ . A matriz *DReIda* do Integra UFRJ, assim como *DIda*, foi definida na seção 5.3.

- **C**: Vetor de tamanho  $N$  que contém a capacidade de armazenamento de bicicletas em cada estação  $S_i$  em elementos  $C_i$ . Como encontrado na seção 5.2, a capacidade das estações  $S_i$ , tal que  $i \in \{1, 2, \dots, N\}$ , são:

$$C = [21 \ 14 \ 14 \ 14 \ 7 \ 12 \ 14 \ 8]^T$$

- **C<sub>v</sub>**: Número que representa a capacidade de armazenamento de bicicletas do veículo de redistribuição. Nas simulações deste trabalho, é considerado que a capacidade do veículo de redistribuição do Integra UFRJ possui a capacidade de carregar dez bicicletas.

- **NbIni**: Vetor de tamanho  $N$  que contém a quantidade inicial de bicicletas em cada estação  $S_i$ , representada pelo elemento  $NbIni_i$ . Nas simulações do Integra UFRJ, é considerado que cada elemento  $NbIni_i$  é aproximadamente igual a metade do elemento correspondente  $C_i$ .

- **NbIni<sub>v</sub>**: Variável que representa a quantidade inicial de bicicletas no veículo de redistribuição. Nas simulações do Integra UFRJ, é considerado que o valor de  $NbIni_v$  é aproximadamente igual a metade de  $C_v$ .

- **TtMin<sub>b</sub>** e **TtMax<sub>b</sub>**: Matrizes  $N \times N$  que representam, respectivamente, os menores e os maiores tempos que levam para o usuário se mover, com uma bicicleta,

da estação da linha  $i$  até a estação da coluna  $j$ , em que  $i, j \in \{1, 2, \dots, N\}$ . Os valores das matrizes para o Integra UFRJ foram calculados na seção 7.1.2.

- **TtMin<sub>c</sub>** e **TtMax<sub>c</sub>**: Matrizes  $N \times N$  que representam, respectivamente, os menores e os maiores tempos que levam para o veículo se mover da estação ou centro de redistribuição da linha  $i$  até a estação ou centro de redistribuição da coluna  $j$ , em que  $i, j \in \{C, 1, 2, \dots, N\}$ . Os valores das matrizes para o Integra UFRJ foram calculados na seção 7.1.2.

- **D<sub>c</sub>**: Matriz  $N \times N$  que representa a distância percorrida pelo veículo da estação ou centro de redistribuição da linha  $i$  até a estação ou centro de redistribuição da coluna  $j$ , em que  $i, j \in \{C, 1, 2, \dots, N\}$ . Os valores dessa matriz para o Integra UFRJ foram calculados na seção 5.4.

- **Ron<sub>St</sub>**: Variável que representa o tempo, em segundos, que a primeira ronda do veículo de redistribuição é realizada durante o dia. Para o sistema a ser simulado, o veículo realiza sua primeira ronda uma hora depois do começo da simulação, ou seja,  $Ron_{St} = 3600$ .

- **Ron<sub>Car</sub>**: Variável que representa o tempo, em segundos, que o veículo aguarda para realizar uma ronda depois de ter iniciado uma ronda anterior. O valor usado para essa espera, neste trabalho, é de  $2h$ , ou seja,  $Ron_{Car} = 7200$ .

- **Max<sub>Fila</sub>**: Vetor de tamanho  $N$  que representa o número máximo de pessoas que aguardam em cada estação  $S_i$  por bicicletas, caso ela esteja cheia. Na simulação, é considerado que, em cada estação, até duas pessoas aguardam por bicicletas enquanto ela estiver cheia.

- **Max<sub>Km</sub>**: Variável que representa a distância máxima, em quilômetros, que o veículo pode percorrer durante uma ronda planejada pela estratégia adaptativa, apresentada na seção 6.2. Como dito anteriormente, na simulação,  $Max_{Km} = 25$ .

- **Gra<sub>Stp</sub>**: Variável que representa o intervalo, em segundos, aguardado para apresentar o gráfico do andamento da simulação, um dos indicadores apresentados na próxima seção. O valor escolhido dessa variável para a simulação realizada neste trabalho é de  $1h$ , ou seja,  $3600s$ .

### 7.3 Indicadores para a avaliação dos desempenhos dos sistemas simulados

Para avaliar como a eficiência do SPCB varia de acordo com as estratégias, torna-se necessário estabelecer indicadores que, além de permitirem a comparação entre as simulações com diferentes estratégias, tornam possível comparar os resultados das simulações com dados de sistemas reais. Os Indicadores utilizados são:

- **tE<sub>i</sub>**: Indica a porcentagem de tempo que cada estação  $S_i$  permaneceu vazia

durante o funcionamento do sistema. Para calculá-lo, no início de cada passo do simulador, é verificado se a marcação  $m(PES_i) = C_i$ . Caso a verificação seja válida, é incrementado o uma unidade a uma variável  $tESum_i$ . Após a simulação de todos os passos, os indicadores  $tE_i$  das estações são calculados através da divisão  $tESum_i/Ts$ .

- **tF<sub>i</sub>**: Indicador semelhante ao apresentado anteriormente, porém indica o tempo, também em porcentagem, que cada estação permanece cheia. Para calculá-la, a cada passo do simulador, se a marcação  $m(PBS_i) = C_i$ , é incrementado uma unidade a uma variável  $tFSum_i$ , que, após o calculo de todos os passos, permite o calculo de  $tF_i$  através da divisão  $tFSum_i/Ts$ .

- **Usr<sub>Lost</sub>**: Indica a porcentagem de usuários que deixaram de usar o sistema por causa de estações vazias. Quando um usuário tenta retirar uma bicicleta da estação  $S_i$  que está vazia, ou seja,  $m(PES_i) = C_i$ , enquanto há uma quantidade  $Max_{Fila}(i)$  de usuários aguardando por vagas na estação, um usuário desiste de usar o sistema, o que é representado pelo incremento de uma unidade de uma variável  $UsrSum_{Lost}$ . Após a ocorrência de todos os passos, o valor  $UsrSum_{Lost}$  é dividido pela quantidade total de usuários que interagiram com o sistema, que é obtida pela soma de todos os elementos da matriz  $Tim_u$ . Assim, a divisão resulta em  $Usr_{Lost}$ .

- **Gráficos dos estados das estações e do veículo de redistribuição**: Para acompanhar o andamento da quantidade de bicicletas em cada estação  $S_i$  durante a simulação, em intervalos  $Gras_{tp}$ , são esboçados gráficos que mostram a quantidade de bicicletas  $m(PBS_i)$  e o número de reordenação  $R_i$  a cada segundo dos passos que já foram simulados. Além disso, há um gráfico que ilustra a quantidade de bicicletas no veículo de redistribuição, ou seja,  $m(PBSCar)$ , e as localizações do veículo a cada segundo, que são verificadas através do lugar  $PCar_C$  do centro de redistribuição e os lugares  $PCar_i$  das estações  $S_1, S_2, \dots, S_8$ .

## 7.4 Resultados

Nesta seção, a partir dos parâmetros do sistema do Integra UFRJ, são realizadas diversas simulações de cinco tipos de combinações das estratégias de balanceamento definidas no capítulo 6: nenhuma estratégia de balanceamento; balanceamento cíclico com números de reordenação fixos; balanceamento adaptativo com números de reordenação fixos; balanceamento cíclico com números de reordenação preditivos; balanceamento adaptativo com números de reordenação adaptativo.

Como alguns dos disparos das transições ocorrem de acordo com escolhas feitas estocasticamente, duas simulações executadas com os mesmos parâmetros podem gerar diferentes resultados. Portanto, a análise dos indicadores de uma simulação individual para cada estratégia não é confiável. Para solucionar esse problema, cada estratégia é simulada 50 vezes, resultando em 50 grupos de indicadores:  $Usr_{Lost}$  e,

para cada estação  $S_i$ ,  $tE_i$  e  $tF_i$ . Para cada tipo de indicador, é feita uma média dos 50 elementos, somando-os e dividindo por 50. Além dos indicadores mencionados, são apresentados também os gráficos das primeiras simulações dentre as 50.

Nota-se que, desconsiderando as simulações feitas anteriormente, para esta seção são feitas 250 simulações de diferentes tipos de estratégias. A realização desta grande quantidade de simulações é possível pois cada uma leva, em média, 20s para calcular e apresentar os resultados em um computador pessoal. Assim, o tempo total decorrido para realizar todas simulações é de aproximadamente uma 1:20h, aceitável considerando que são simuladas cada segundo de 4000h de funcionamento do sistema.

### 7.4.1 Sistema sem balanceamento

Para avaliar o quanto cada estratégia melhora a eficiência do sistema individualmente, é necessário obter o rendimento do sistema sem a utilização de um veículo de redistribuição. Por isso, nesta seção, são feitas simulações do funcionamento do Integra UFRJ sem balanceamento.

Em uma primeira simulação do sistema, foi gerado o gráfico da figura 7.2 e, após 50 simulações, os indicadores da tabela 7.1. Neles, nota-se que todas as estações possuem problemas com suas quantidades de bicicletas durante o dia, seja esse problema relacionado com a falta de bicicletas ou o excesso de bicicletas. Além disso, aproximadamente 11% dos usuários, o que resulta em aproximadamente 37 pessoas, deixaram de usar o sistema por encontrar estações vazias.

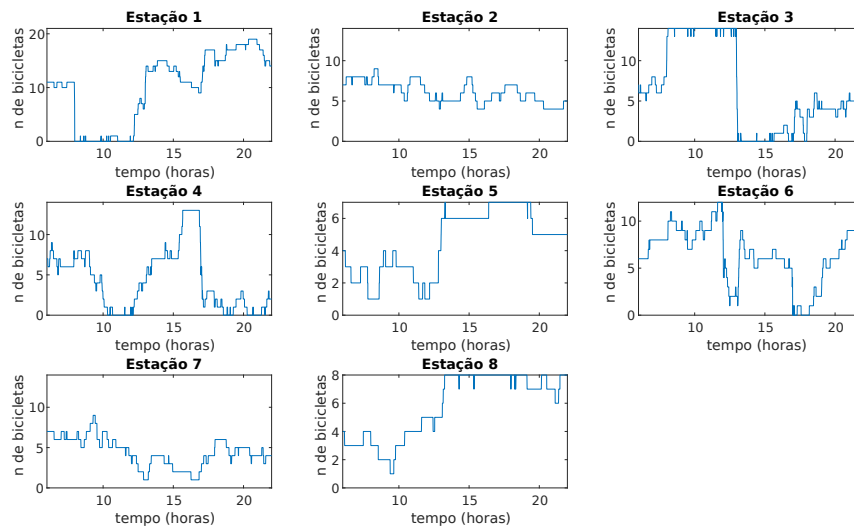


Figura 7.2: Gráfico da quantidade de bicicletas em cada estação durante a simulação do sistema sem balanceamento.

Tabela 7.1: Indicadores médios das 50 simulações do sistema sem balanceamento.

Estação	Porcentagem de tempo cheia	Porcentagem de tempo vazia	Porcentagem de usuários perdidos
$S_1$	19,30 %	13,45 %	
$S_2$	7,42 %	2,51 %	
$S_3$	14,54 %	20,44 %	
$S_4$	5,66 %	24,73 %	
$S_5$	14,24 %	10,85 %	
$S_6$	7,82 %	19,20 %	
$S_7$	0,93 %	12,88 %	
$S_8$	19,43 %	3,94 %	
<i>Todas</i>			10,99 %

Além das informações sobre a eficiência do sistema, a comparação entre o gráfico e a tabela mostrou também a necessidade de simular o sistema mais de uma vez para avaliar resultados. Essa necessidade pode ser notada, por exemplo, através da porcentagem de tempo que a estação  $S_7$  ficou vazia, pois apesar da porcentagem ser nula na figura 7.2, o indicador médio obtido pela média dos indicadores das 50 simulações, cujo valor é 12,88%, segundo a tabela 7.1, mostra que, normalmente, a estação  $S_7$  se encontra vazia por um tempo durante o dia.

#### 7.4.2 Sistema com balanceamento cíclico e números de reordenação fixos

A primeira estratégia de balanceamento simulada é a estratégia cíclica sem alterar os números de reordenação. Nela, o valor  $R_i$  para cada estação  $S_i$  é, durante toda a simulação, o inteiro aproximado da divisão de  $C_i$  por dois. Assim como na simulação apresentada anteriormente, é gerado o gráfico da figura 7.3, relativo à primeira simulação, e uma tabela 7.2 contendo as médias dos indicadores das 50 simulações.

Apesar da porcentagem de usuários perdidos ter reduzido, aproximadamente, apenas 2% com a implementação do balanceamento, os tempos em que as estações permaneceram cheias e vazias reduziram bastante. Nota-se isso pois os valores máximos de tempos em que as estações ficaram cheias e vazias com a estratégia foram, respectivamente, 6,27% e 15,19%, significativamente menores que os valores máximos, 19,43% e 24,73%, das simulações sem estratégia.



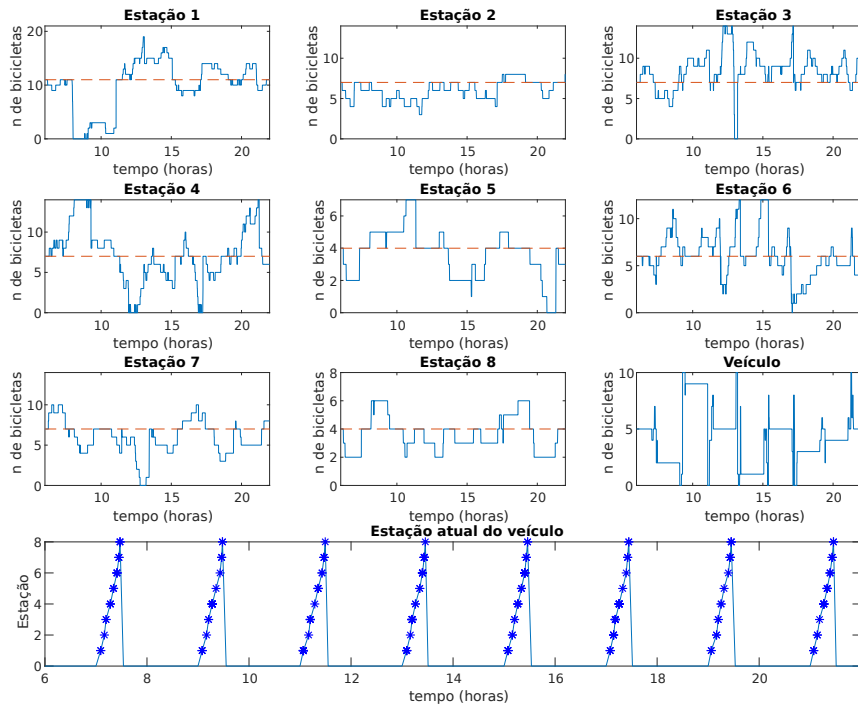


Figura 7.3: Gráfico da quantidade de bicicletas em cada estação (azul) em relação ao número de reordenação (laranja), da quantidade de bicicletas no veículo e das estações que foram visitadas pelo veículo ao longo da simulação utilizando estratégia de balanceamento cíclico com números de reordenação constantes.

Tabela 7.2: Indicadores médios das 50 simulações do sistema com balanceamento cíclico com números de reordenação constantes.

Estação	Porcentagem de tempo cheia	Porcentagem de tempo vazia	Porcentagem de usuários perdidos
$S_1$	2,54 %	7,28 %	
$S_2$	0,36 %	0,12 %	
$S_3$	6,27 %	1,83 %	
$S_4$	2,24 %	9,05 %	
$S_5$	1,76 %	1,04 %	
$S_6$	2,61 %	15,19 %	
$S_7$	0,23 %	2,46 %	
$S_8$	1,08 %	1,57 %	
<i>Todas</i>			9,74 %

### 7.4.3 Sistema com balanceamento adaptativo e números de reordenação fixos

Para aumentar ainda mais a eficiência do balanceamento, é simulada a estratégia adaptativa, otimizando a escolha da rota do veículo de acordo com a necessidade do sistema no momento do balanceamento. Assim como a estratégia anterior, as simulações que resultam no gráfico da figura 7.4 e na tabela 7.3 são feitas utilizando os mesmos números de reordenação  $R_i = C_i/2$  para cada estação  $S_i$ .

Como a estratégia planeja a rota da ronda de acordo com o estado atual do sistema, a média dos indicadores da tabela 7.3 melhoraram vagamente em relação à estratégia anterior, reduzindo, aproximadamente, 1% a perda de usuários e os valores máximos de tempos cheios e vazios para, respectivamente, 2, 40% e 7, 96%.

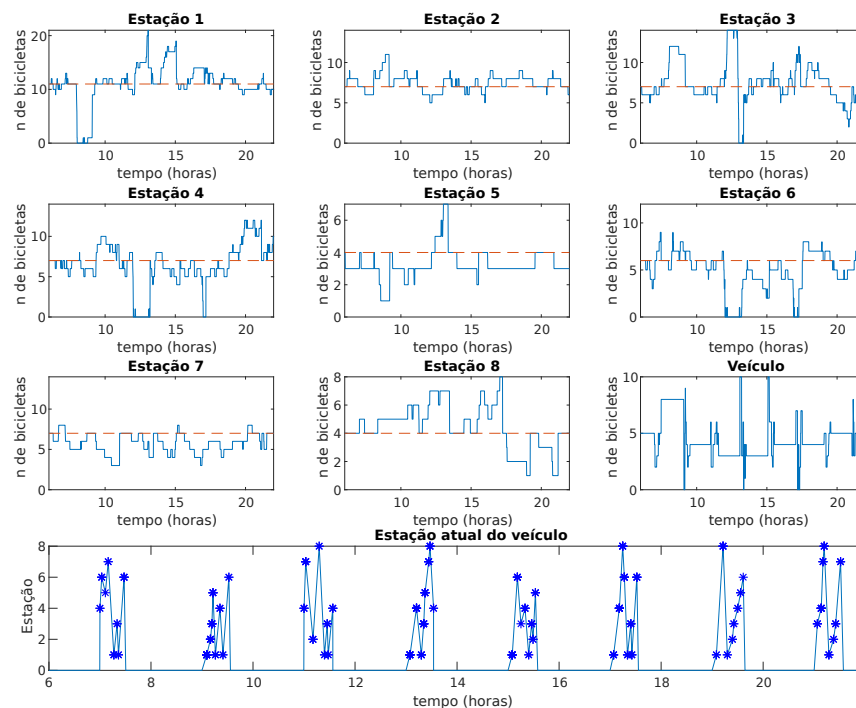


Figura 7.4: Gráfico da quantidade de bicicletas em cada estação (azul) em relação ao número de reordenação (laranja), da quantidade de bicicletas no veículo e das estações que foram visitadas pelo veículo ao longo da simulação utilizando estratégia de balanceamento adaptativo com números de reordenação constantes.

Tabela 7.3: Indicadores médios das 50 simulações do sistema com balanceamento adaptativo com números de reordenação constantes.

Estação	Porcentagem de tempo cheia	Porcentagem de tempo vazia	Porcentagem de usuários perdidos
$S_1$	0,48 %	5,57 %	
$S_2$	0,18 %	0,01 %	
$S_3$	2,40 %	1,41 %	
$S_4$	0,77 %	7,75 %	
$S_5$	0,89 %	0,80 %	
$S_6$	1,27 %	7,96 %	
$S_7$	0,28 %	0,19 %	
$S_8$	1,01 %	0,45 %	
<i>Todas</i>			8,62 %

#### 7.4.4 Sistema com balanceamento cíclico e números de reordenação preditivos

Para que o balanceamento das bicicletas das estações seja feito considerando os picos de utilização estudados, nas próximas simulações, é utilizada a estratégia preditiva apresentada na seção 6.3, tornando os números de reordenação  $R_i$  altos durante os picos que ocorrem em algumas estações  $S_i$ . Como a estratégia só altera o valor do número de reordenação, ela deve ser combinada com uma estratégia responsável pela rota que o veículo utiliza em cada ronda. Nesta seção, a estratégia é combinada com a estratégia de balanceamento cíclico.

Após as simulações, os resultados, ilustrados no gráfico da figura 7.5 e na tabela 7.4, mostram que, apesar dos indicadores de tempo que as estações ficam cheia terem aumentado em relação às simulações na seção 7.4.2, a porcentagem de clientes perdidos por causa de estações vazias reduziu significativamente. Isto ocorreu pois, antes dos picos, as estações se encontravam cheias, aumentando a disponibilidade de bicicletas para a grande quantidade de usuários que chegaram na estação nesses momentos.

Pode-se dizer, através do gráfico, que os aumentos propositais das disponibilidades de bicicletas em certos momentos causaram o aumento dos indicadores de estações cheias. Como o algoritmo aumenta o número de reordenação da estação  $S_i$  que está em um momento de pico para 90% da capacidade da estação, é necessário apenas a chegada de 10% de  $C_i$  usuários na estação para que ela fique lotada de bicicletas.

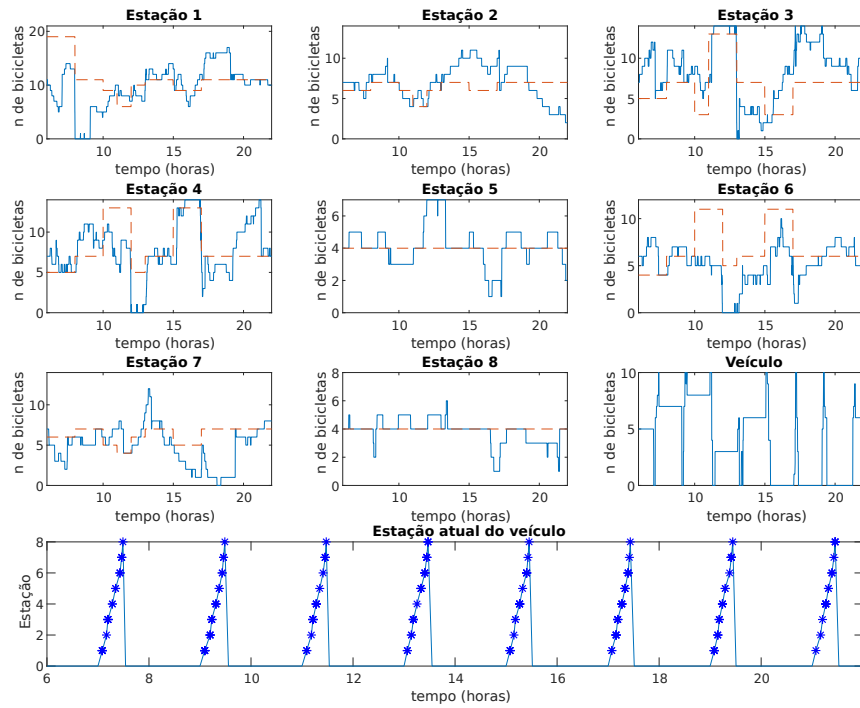


Figura 7.5: Gráfico da quantidade de bicicletas em cada estação (azul) em relação ao número de reordenação (laranja), da quantidade de bicicletas no veículo e das estações que foram visitadas pelo veículo ao longo da simulação utilizando estratégia de balanceamento cíclico com números de reordenação preditivos.

Tabela 7.4: Indicadores médios das 50 simulações do sistema com balanceamento cíclico com números de reordenação preditivos.

Estação	Porcentagem de tempo cheia	Porcentagem de tempo vazia	Porcentagem de usuários perdidos
$S_1$	2,03 %	5,81 %	
$S_2$	0,43 %	0,33 %	
$S_3$	9,96 %	2,23 %	
$S_4$	4,87 %	6,69 %	
$S_5$	2,00 %	1,93 %	
$S_6$	2,95 %	8,5 %	
$S_7$	0 %	2,68 %	
$S_8$	1,62 %	2,08 %	
<i>Todas</i>			4,19 %

### 7.4.5 Sistema com balanceamento adaptativo e números de reordenação preditivos

O último tipo de simulação realizado nesse trabalho consiste da combinação entre a estratégia adaptativa e a estratégia preditiva, tornando a rota do veículo e a resposta à picos mais eficientes. Assim como as outras simulações, são gerados o gráfico da figura 7.6 e a tabela 7.5.

Apesar dessa estratégia aumentar as porcentagens de tempo em que as estações  $S_4$ ,  $S_5$ ,  $S_6$  e  $S_7$  permanecem cheias em relação às simulações da seção 7.4.4, todos os outros indicadores foram reduzidos. Além da porcentagem de usuários perdidos por causa de estações vazias ter reduzido cerca de metade do valor anterior, as quantidades de tempo que algumas estações permanecem vazias melhoraram consideravelmente, como, por exemplo, a estação  $S_7$ , que mudou de 8,58% das simulações da seção 7.4.4 para para 0,99% das simulações desta seção.

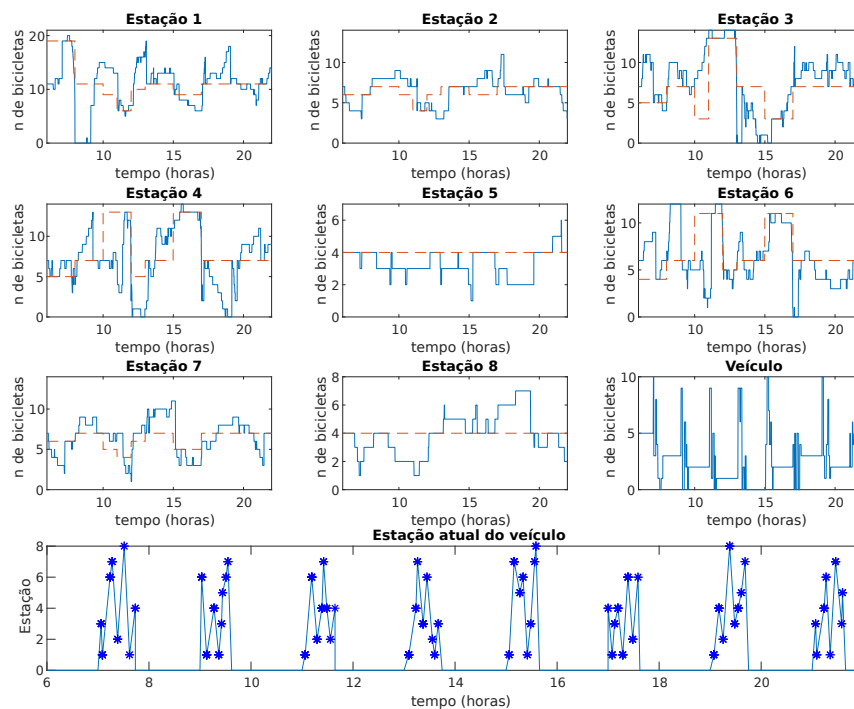


Figura 7.6: Gráfico da quantidade de bicicletas em cada estação (azul) em relação ao número de reordenação (laranja), da quantidade de bicicletas no veículo e das estações que foram visitadas pelo veículo ao longo da simulação utilizando estratégia de balanceamento adaptativo com números de reordenação preditivos.

Tabela 7.5: Indicadores médios das 50 simulações do sistema com balanceamento adaptativo com números de reordenação preditivos.

<b>Estação</b>	<b>Porcentagem de tempo cheia</b>	<b>Porcentagem de tempo vazia</b>	<b>Porcentagem de usuários perdidos</b>
$S_1$	0,33 %	4,85 %	
$S_2$	0,07 %	0,06 %	
$S_3$	6,93 %	2,2 %	
$S_4$	5,33 %	4,21 %	
$S_5$	2,11 %	0,87 %	
$S_6$	6,29 %	0,99 %	
$S_7$	0,04 %	0,64 %	
$S_8$	1,27 %	1,01 %	
<i>Todas</i>			2,27 %

# Capítulo 8

## Conclusão e trabalhos futuros

Neste trabalho, foi proposto o modelo de um novo tipo de rede de Petri, as redes de Petri estocásticas generalizadas com controle externo, para descrever o funcionamento de um sistema público de compartilhamento de bicicletas com um veículo de redistribuição. Além do modelo permitir a análise do rendimento de um sistema qualquer com suas características estocásticas, ele torna possível a implementação de diferentes tipos de estratégias de balanceamento, através de um veículo, utilizando um algoritmo externo.

A descrição da dinâmica do modelo através de blocos matrizes facilitou significativamente a implementação de um simulador capaz de replicar o funcionamento do modelo de diferentes SPCBs com diferentes quantidades  $N$  de estações. Além disso, através das variáveis de entrada do simulador, é possível alterar características do mesmo sistema para avaliar o impacto dessas mudanças no sistema como, por exemplo, remover uma estação que não está sendo utilizada com frequência do sistema.

Como o simulador implementado apresentou bons tempos de simulação para o sistema do Integra UFRJ em computadores pessoais, variando de 20s a 1min, foi possível simular o mesmo sistema varias vezes. Isso permitiu a realização de uma média dos resultados das simulações que, conseqüentemente, são dados mais confiáveis do que os apresentados por apenas uma simulação.

Apesar da formulação do modelo e simulador que representam o funcionamento do sistema, pode-se considerar que a maior contribuição deste trabalho foi a introdução, no modelo e no simulador, da capacidade de ser implementado um controle externo para a definição de rotas do veículo de redistribuição. O controle externo é apropriado pois a mudança do tipo de estratégia não precisa ser realizada diretamente na rede de Petri, dispensando a necessidade de replanejá-la e reconstruí-la. Além disso, por estar implementado separadamente do simulador, o controle externo pode ser utilizado diretamente em sistemas reais. Logo, sistemas públicos de bicicletas reais, como o Integra UFRJ, poderiam utilizar diretamente o controle im-

plementado neste trabalho para organizar as rondas do seu veículo de redistribuição.

Finalmente, dentre as estratégias de balanceamento apresentadas, pode-se concluir que a estratégia mais eficiente dentre as apresentadas neste trabalho é a apresentada na seção 7.4.5. A combinação da estratégia adaptativa apresentada na seção 6.2 com a estratégia preditiva apresentada na seção 6.3 otimizou simultaneamente a resposta do sistema aos picos de utilização e o planejamento da ronda do veículo, priorizando as escolhas das estações que mais precisam de balanceamento ao invés de todas.

Mesmo que este trabalho tenha proposto um novo tipo de rede de Petri para permitir a implementação de diferentes estratégias para rondas, através de resoluções de conflitos, a rede não possui características que descrevem, formalmente, a capacidade de um algoritmo externo de alterar os pesos dos arcos, como é feito na estratégia preditiva descrita na seção 6.3. Essa falta de formalismo resultou na necessidade de gerar novamente a rede de Petri sempre que algum peso é alterado, o que aumentou significativamente o tempo de simulação. Portanto, trabalhos futuros podem descrever um novo tipo de rede de Petri que inclua essa característica

Outra característica importante para um sistema público de compartilhamento de bicicletas que realiza balanceamento que não foi abordada neste trabalho é a capacidade de um sistema utilizar dois ou mais veículos de balanceamento que realizam diferentes rondas simultaneamente. Apesar de um veículo ter sido suficiente para o sistema do Integra UFRJ, que é relativamente pequeno, um sistema de grande porte necessitaria da utilização de mais de um veículo. Para realizar essa adição, o modelo em redes de Petri deve ser refeito de tal forma que cada veículo possa operar individualmente, porém de forma organizada para que o balanceamento possa ser feito com êxito.



# Referências Bibliográficas

- [1] ANTUNES, I. *Um Modelo em Redes de Petri para Determinação de Estratégias de Balanceamento de Sistemas Públicos de Compartilhamento de Bicicletas*. Dissertação de mestrado, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Rio de Janeiro, Brasil, 2015.
- [2] ANTUNES, I., CARVALHO, L. K., BASILIO, J. C. “A stochastic Petri net model for simulation-based performance analysis of public bicycle sharing systems”. In: *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 433–439, Aug 2016. doi: 10.1109/COASE.2016.7743438.
- [3] HAYEK, H. E., NACOUZI, J. “Hangzhou Public Bicycle: Understanding Early Adoption and Behavioral Response to Bikeshare in Hangzhou, China”, *Transportation Research Record*, v. 2247, 01 2011.
- [4] CASSANDRAS, C. G., LAFORTUNE, S. *Introduction to Discrete Event Systems*. 2 ed. New York, Springer, 2008.
- [5] DAVID, R., ALLA, H. “Petri nets for modeling of dynamic systems: A survey”, *Automatica*, v. 30, n. 2, pp. 175–202, 1994.
- [6] MURATA, T. “Petri nets: Properties, analysis and applications”, *Proceedings of the IEEE*, v. 77, n. 4, pp. 541–580, 1989.
- [7] MARSAN, M. A., CONTE, G., BALBO, G. “A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems”, *ACM Transactions on Computer Systems (TOCS)*, v. 2, n. 2, pp. 93–122, 1984.
- [8] JOÃO ALBERTO FABRO. “Técnicas de Projeto de Algoritmos Força Bruta”. <http://www.dainf.ct.utfpr.edu.br/~fabro/IF64C/ForcaBruta.pdf>. Acessado: 2019-23-01.
- [9] ROBINET, M. B. P. B. B. C. A. D. L. T. F. L. F. M. L. “Balancing the stations of a self service bike hire system”, *RAIRO-Operations Research*, v. 45, n. 1, pp. 37–61, 2011.

- [10] CALVO, D. C. F. M. R. W. “Bike sharing system: solving the static rebalancing problem”, *Discrete Optimization*, v. 10, n. 2, pp. 120–146, 2013.
- [11] OTTOMANELLI, L. C. M. “A dynamic simulation based model for optimal fleet repositioning in bike-sharing systems”, *Procedia - Social and Behavioral Sciences*, v. 87, pp. 203–210, 2013.
- [12] FUNDO VERDE. “Fundo Verde de Desenvolvimento e Energia para a Cidade Universitária da Universidade Federal do Rio de Janeiro”. [www.fundoverde.ufrj.br/index.php/pt/o-fundo/sobre](http://www.fundoverde.ufrj.br/index.php/pt/o-fundo/sobre), . Acessado: 2019-08-01.
- [13] FUNDO VERDE. “Relatório de 2017 do projeto Integra UFRJ”. [www.fundoverde.ufrj.br/phocadownload/Relatrio%20de%20utilizacao2017\\_09.01.18.pdf](http://www.fundoverde.ufrj.br/phocadownload/Relatrio%20de%20utilizacao2017_09.01.18.pdf), . Acessado: 2019-08-01.
- [14] “Mapa do projeto Integra UFRJ”. [www.integraufrj.com.br/mapaestacao.aspx](http://www.integraufrj.com.br/mapaestacao.aspx). Acessado: 2019-08-01.
- [15] FUNDO VERDE. “Relatório do mês de novembro do projeto Integra UFRJ”. [drive.google.com/file/d/1gXAfTZkxJ6Cu7LhHf86ocUZhtbz0YwvU/view](https://drive.google.com/file/d/1gXAfTZkxJ6Cu7LhHf86ocUZhtbz0YwvU/view), . Acessado: 2019-08-01.
- [16] FUNDO VERDE. “Relatório do mês de outubro do projeto Integra UFRJ”. <https://drive.google.com/file/d/1GMhvyILvcly-9F433C78gx09cmrn36aq/view>, . Acessado: 2019-08-01.