



## **Escola Politécnica da Universidade de São Paulo**

LUCAS GOMES AVEIRO

PAULO FERRARI ABUD

ORIENTADOR: PAULO EIGI MIYAGI

### **MODELAGEM DE SISTEMAS DE SINALIZAÇÃO DE TRÂNSITO UTILIZANDO REDES DE PETRI**

São Paulo, 2015

Lucas Gomes Aveiro, Nº USP 7133683  
Paulo Ferrari Abud, Nº USP 7627008

## **Projeto de sistemas de sinalização de trânsito utilizando Redes de Petri**

Monografia apresentada no  
Departamento de Engenharia  
Mecatrônica e Sistemas Mecânicos  
da Escola Politécnica da  
Universidade de São Paulo para  
obtenção do título de Engenheiro.  
Área de Concentração: Engenharia  
Mecatrônica

Escola Politécnica da USP – EPUSP  
Engenharia Mecatrônica  
Graduação

Orientador: Prof. Dr. Paulo Eigi Miyagi

São Paulo-SP, Brasil  
2015

## Declaração de originalidade

Este relatório é apresentado como requisito parcial para obtenção do título de Engenheiro na Escola Politécnica da Universidade de São Paulo. É o produto do nosso próprio trabalho, exceto onde indicado no texto. O relatório pode ser livremente copiado e distribuído desde que a fonte seja citada.



Lucas Gomes Aveiro



Paulo Ferrari Abud

#### Catálogo-na-publicação

Aveiro, Lucas Gomes

Modelagem de sistemas de sinalização de trânsito utilizando redes de Petri

/ L. G. Aveiro, P. F. Abud -- São Paulo, 2015.

73 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

## RESUMO

O seguinte trabalho trata de modelagem de sistemas de sinalização em vias públicas de circulação de veículos automotores. Visto a idade já elevada da maioria dos semáforos presentes nos cruzamentos da cidade de São Paulo e dificuldades na manutenção, os mesmos apresentam problemas de funcionamento com frequência relativamente alta e sua solução leva muito tempo por exigir o envio de operadores ao local da pane. A renovação da tecnologia desses semáforos é necessária e seria ideal que os mesmos pudessem ser reinicializados remotamente. Com essa motivação, o trabalho foca na modelagem e análise de cruzamentos por meio da teoria de rede de Petri, com a finalidade de especificar soluções de controle que sejam implementáveis em controladores programáveis. Para a modelagem de cruzamentos, é utilizada a rede de Petri sincronizada temporizada (STPN). Diversos cruzamentos com diferentes números de fases são modelados. É apresentada ainda uma forma de tratar o caso de um conjunto de cruzamentos para operarem de forma sincronizada e é mostrado um exemplo de conversão de um modelo em rede de Petri para um programa em linguagem LD (*ladder diagrams*), amplamente utilizada em controladores programáveis.

**Palavras-chave:** Rede de Petri; Modelagem; Semáforo; Trânsito; STPN; Controlador Programável; Linguagem LD.

## ABSTRACT

The following paper discusses modelling of traffic light systems on public roads for automotive vehicle circulation. Due to the already advanced age of the majority of the traffic lights located at São Paulo's intersections and to the difficulty in the maintenance, the traffic lights malfunction with relatively high frequency and the fix takes a long time because it is necessary to send operators to the site of the breakdown. The renewal of those traffic lights' technology is needed and it would be ideal if they could be remotely restarted. With this motivation, this paper focuses on the modelling and analysis of these intersections through the Petri net theory in order to specify control solutions compatible with programmable controllers. For the intersection modelling, the synchronized timed Petri net (STPN) is used. Several traffic intersections with different number of phases are modelled. It is shown a way to handle the case of a group of intersections to make them work synchronously and it is also presented an example of a Petri net model conversion to a script in LD (*ladder diagrams*) language, widely used in programmable controllers.

**Key-words:** Petri Nets; Modelling; Traffic Light; Traffic Jam; STPN; Programmable Controller; Ladder Diagrams language.

## LISTA DE FIGURAS

Figura 1 – Diagrama do sistema SURTRAC (Adaptado de Smith et al., 2013) .....	6
Figura 2 - Relacionamento entre tempo e espaço entre veículos (Adaptado de Sims e Dobinson, 1980) ...	8
Figura 3 - Metodologia de projeto de sistemas de controle (Adaptado de Miyagi, 1996).....	13
Figura 4 - Procedimento para análise de sistemas por simulação (Reproduzido de Miyagi, 2006) .....	15
Figura 5 - Modelo em rede de Petri realizado por Chen et al. (Adaptado de Chen et al., 2014).....	18
Figura 6 - Modelo em rede de Petri híbrida de um cruzamento de duas <b>fases</b> realizado por Zhang et al.	19
Figura 7 - Modelagem de um cruzamento de duas <b>fases</b> utilizando STPN e sistema master-slave .....	21
Figura 8 - Conexão de sinais externos para sincronização das redes de Petri .....	22
Figura 9 – Exemplo de cruzamento para modelagem com rede de Petri temporizada colorida .....	23
Figura 10 – Cruzamento de duas <b>fases</b> que será modelado inicialmente .....	29
Figura 11 - Modelo do Cruzamento 1 .....	30
Figura 12 - Grafo de alcançabilidade do Cruzamento 1.....	31
Figura 13 - Resultados da análise de espaço de estados do Cruzamento 1 .....	31
Figura 14 – Simulação do modelo do Cruzamento 1 .....	33
Figura 15 - Cruzamento 2 (4 <b>fases</b> ).....	34
Figura 16 - Modelo do Cruzamento 2 .....	35
Figura 17 - Grafo de alcançabilidade do Cruzamento 2.....	36
Figura 18 - Resultados da análise de espaço de estados do Cruzamento 2 .....	37
Figura 19 – Simulação do modelo do Cruzamento 2 .....	38
Figura 20 - Cruzamento 3 (8 <b>fases</b> ).....	39
Figura 21 - Modelo do Cruzamento 3 .....	40
Figura 22 - Modelagem das quatro situações possíveis para dois semáforos com uma <b>fase</b> em comum .	42
Figura 23 - Grafo de alcançabilidade do Cruzamento 3.....	44
Figura 24 - Análise de espaço de estados do Cruzamento 3 .....	44
Figura 25 – Simulação do modelo do Cruzamento 3 - Parte 1 .....	46
Figura 26 – Simulação do modelo do Cruzamento 3 - Parte 2 .....	47
Figura 27 - Modelo do Cruzamento 4 .....	48
Figura 28 - Grafo de alcançabilidade do Cruzamento 4.....	49
Figura 29 – Análise de espaço de estados do Cruzamento 3.....	50
Figura 30 – Simulação do modelo do Cruzamento 4 .....	52
Figura 31 - Cruzamento 5.....	53
Figura 32 - Modelo do semáforo para pedestres .....	54
Figura 33 - Exemplo de implementação do semáforo para pedestres no Cruzamento 5 .....	55
Figura 34 - Simulação do modelo do Cruzamento 5.....	56
Figura 35 - Sequência de aberturas e fechamentos do exemplo de conjunto de cruzamentos .....	57
Figura 36 - Módulo do cruzamento central (master) .....	58
Figura 37 - Módulo dos cruzamentos slave .....	59
Figura 38 - Modelo do conjunto de cruzamentos interligados.....	59
Figura 39 – Interface de entradas e saídas discretas do PLCLogix.....	62
Figura 40 – Modelo em Ladder Diagram (LD) do cruzamento de duas <b>fases</b> .....	63
Figura 41 - Simulação do cruzamento em LD.....	64

## SUMÁRIO

<b>1.</b>	<b>INTRODUÇÃO .....</b>	<b>1</b>
1.1.	Objetivo .....	2
1.2.	Estrutura do texto .....	2
1.3.	Escopo do projeto.....	3
<b>2.</b>	<b>REVISÃO BIBLIOGRÁFICA .....</b>	<b>4</b>
2.1.	Sistema semafórico .....	4
2.1.1.	Semáforo .....	4
2.1.2.	Semáforo inteligente .....	4
2.1.3.	Exemplos de sistemas semafóricos em desenvolvimento .....	5
2.2.	Modelagem de sistemas .....	9
2.2.1.	Sistemas a eventos discretos .....	9
2.2.2.	Rede de Petri .....	9
2.2.3.	Relação entre rede de Petri e LD .....	10
2.2.4.	Softwares para modelagem e análise .....	11
2.3.	Sistemas de semáforos e modelagem dos mesmos em rede de Petri .....	12
2.4.	Metodologia de projeto de sistemas de controle .....	13
<b>3.</b>	<b>ANÁLISE DE REQUISITOS .....</b>	<b>16</b>
<b>4.</b>	<b>TÉCNICAS DE MODELAGEM .....</b>	<b>17</b>
4.1.	Possíveis abordagens .....	17
4.2.	Matriz de decisão.....	23
4.2.1.	Critérios da matriz de decisão .....	24
4.2.2.	Avaliação da matriz resultante .....	25
4.3.	Validação do modelo.....	26
4.4.	Detalhamento do projeto.....	26
<b>5.</b>	<b>ANÁLISE DE RISCO.....</b>	<b>28</b>
<b>6.</b>	<b>MODELAGEM E ANÁLISE.....</b>	<b>29</b>
6.1.	Cruzamento 1.....	29
6.2.	Cruzamento 2.....	34
6.3.	Cruzamento 3.....	39
6.4.	Cruzamento 4 (com sensores) .....	48
6.5.	Cruzamento 5 (com semáforos para pedestres) .....	53
6.6.	Sincronização de um conjunto de cruzamentos .....	57
6.7.	Conversão para linguagem LD .....	60
<b>7.</b>	<b>CONCLUSÃO .....</b>	<b>66</b>
<b>8.</b>	<b>BIBLIOGRAFIA .....</b>	<b>67</b>

## 1. Introdução

O sistema de semáforos das ruas e avenidas da cidade de São Paulo entra em colapso com grande frequência devido ao desgaste dos componentes eletrônicos e intempéries como chuva, raios, calor intenso ou quedas de energia (CET, 2013). Somente no período de chuvas de janeiro a fevereiro em 2013, foram registradas 6.508 ocorrências emergenciais devido a semáforos inoperantes (CET, 2013). Neste mesmo ano, a cidade de São Paulo tinha 6.169 cruzamentos, sendo que a idade média dos semáforos implantados era de 20 anos (CET, 2013). A maioria dos semáforos existentes foram concebidos de modo que na ocorrência de uma pane, o mesmo passa ao estado de sinal luminoso amarelo intermitente ou algum outro estado imprevisto, exigindo que seja enviado um operador até o local para reiniciar o aparelho (Vilanova, 2005).

Além da alta frequência de falhas no sistema devido às intempéries e à idade avançada dos equipamentos, há ainda um problema adicional no atendimento às ocorrências. O tempo médio entre o recebimento das informações de falha e a solução dos problemas é de 9 horas (CET, 2013). Durante esse período, o semáforo fica inoperante até que um operador se desloque ao local e realize o reparo necessário do aparelho.

A maioria dos semáforos existentes na cidade de São Paulo não possui meios de alterar seus tempos de controle de acordo com a demanda ao longo do dia (Christianini e Hagiwara, 2008). Uma otimização na operação do sistema de semáforos, como a alteração de seus tempos de controle conforme a demanda, é capaz de gerar uma melhora significativa no trânsito (Vilanova, 2006). Sendo assim, é cada vez mais relevante o estudo da implementação de semáforos com tempos variáveis de controle, que visam otimizar o fluxo do trânsito da cidade em cruzamentos de grande movimentação.

A necessária renovação dos semáforos da cidade de São Paulo é uma oportunidade especial para revisar a tecnologia semafórica e implantar um sistema de controle para que os semáforos possam se reiniciar automaticamente em caso de falhas simples, evitando a necessidade de envio de um operador ao local e reduzindo o tempo no qual o semáforo fica inoperante. A capacidade de alterar os tempos de controle dos semáforos para

atender às demandas ao longo do dia também é uma característica desejável ao sistema.

A utilização de controladores programáveis permite maior flexibilidade e facilidade de reconfiguração. Além disso, com estes, é possível reiniciar o sistema remotamente (Hjalmarsson e Johansson, 2011). Para implementá-los, é necessário fazer um estudo e criar modelos que capturem o comportamento dos cruzamentos e as soluções de controle resultantes sejam convertíveis em programas de controladores programáveis.

Uma ferramenta poderosa para realizar esse estudo de sistemas semafóricos é a rede de Petri, capaz de descrever, analisar e especificar a solução de controle de sistemas a eventos discretos, como é o caso dos sistemas de controle de tráfego de veículos (List et al., 2004). Outro aspecto a ser considerado é que já existem ferramentas para a conversão de modelos em rede de Petri para programas de controladores programáveis.

### **1.1. Objetivo**

O objetivo desse trabalho é fazer a análise e modelagem de cruzamentos semafóricos utilizando-se rede de Petri, de modo a facilitar a implementação do controle semafórico por controladores programáveis.

### **1.2. Estrutura do texto**

Na seção 2 é apresentada a revisão bibliográfica. Nela são introduzidos conceitos básicos da engenharia de tráfego, fundamentos de modelagem de sistemas e rede de Petri, trabalhos correlatos em modelagem de sistemas de semáforos, uma metodologia de projeto de sistemas de controle e o estado da arte de sistemas semafóricos.

Na seção 3 são apresentados e analisados os requisitos do trabalho. Na seção 4 é feita uma análise de soluções utilizando matriz de decisão para selecionar a melhor abordagem para o problema. Na seção 5 é realizada uma análise de riscos, que mostra as incertezas que podem prejudicar os resultados finais do trabalho. Na seção 6 é feita a modelagem e análise de diversos tipos de cruzamento, além de ser mostrado um exemplo de transcrição do modelo

para a linguagem LD. Finalmente, na seção 7, são apresentadas as conclusões do trabalho.

### **1.3. Escopo do projeto**

Este trabalho engloba a modelagem e análise de cruzamentos utilizando-se rede de Petri para implementação em controladores. Sendo assim, busca-se chegar a modelos validados que possam ser implementados em controladores programáveis para realizar o controle de cruzamentos semaforicos.

Não estão no escopo deste trabalho a melhoria do fluxo de veículos, a criação de um sistema de semáforos "inteligentes", nem a redução da frequência de falha dos semáforos. Busca-se apenas analisar e modelar cruzamentos de tipos diversos em rede de Petri que possam ser implementados em controladores programáveis. A implementação do controle semaforico com estes facilita a reinicialização dos semáforos após uma pane, não exigindo o envio de um operador ao local e, conseqüentemente, reduzindo o tempo em que o semáforo ficará inoperante.

Neste trabalho também não é feita a implementação dos modelos em controladores programáveis. Além disso, não são feitos modelos válidos para todos os tipos de cruzamentos existentes. Ao invés disso, busca-se desenvolver modelos para cruzamentos representativos que possam ser adaptados ou utilizados como base para chegar a modelos para cruzamentos específicos.

## 2. Revisão bibliográfica

### 2.1. Sistema semafórico

#### 2.1.1. Semáforo

O semáforo é um dispositivo de controle que alterna o direito de passagem de veículos e pedestres em cruzamentos de vias por meio de sinais luminosos (DENATRAN, 1984). O "sinal luminoso verde" indica que a passagem está liberada. O "sinal luminoso amarelo" indica que o veículo deve parar a não ser que isso cause risco para a segurança do tráfego. Já o "sinal luminoso vermelho" proíbe a passagem de veículos ou pedestres pela via.

O tempo total para que seja executada a sequência de indicação de um sinal verde para amarelo e posteriormente vermelho, voltando depois para verde, é denominado **ciclo**. O **ciclo** é composto de **estágios**, que são os tempos nos quais os sinais luminosos não mudam de estado. Há ainda o **período entreverdes**, que é o tempo entre o fim da indicação verde de um **ciclo** e o início do outro (DENATRAN, 1984). Define-se ainda **fase** como sendo cada um dos estados em que um cruzamento semafórico pode estar.

Os semáforos podem ter controladores de tempo fixo, com tempos de **estágios** constantes, ou controladores por demanda de tráfego, que regulam o tempo dos **estágios** em um cruzamento de acordo com a demanda, sendo assim chamados "semáforos inteligentes" (DENATRAN, 1984).

#### 2.1.2. Semáforo inteligente

Os semáforos chamados "inteligentes" fazem uso de sensores para ajustar o tempo de cada **estágio** baseado no fluxo de veículos nas vias adjacentes. Basicamente, o que é feito é aumentar o tempo do **estágio** verde em vias com maior fluxo de veículos com consequente aumento do tempo do **estágio** vermelho das com menor demanda (Chavan *et al.*, 2009). Utilizando-se tal tecnologia de controle em tempo real dos semáforos, é observado que há uma redução do tempo médio de espera dos veículos (Hu e Wang, 2010).

Além dos congestionamentos, problemas relacionados aos semáforos de tempo fixo, como a necessidade de espera mesmo sem tráfego e carros de emergência presos no trânsito, podem ser solucionados ou amenizados com o uso de semáforos inteligentes (Chavan *et al.*, 2009).

Na ocorrência de eventos especiais, como jogos esportivos ou mesmo acidentes, um semáforo inteligente é capaz de absorver e suprir a demanda súbita gerada. Os tempos dos **estágios** em semáforos desse tipo podem ser alterados automaticamente ou manualmente em uma central de comando (Xie, 2013).

### **2.1.3. Exemplos de sistemas semaforicos em desenvolvimento**

São identificados aqui dois sistemas semaforicos que resultam de conhecimento acumulado e sistematizado na área de sistemas de controle de trânsito.

#### **a) Sistema SURTRAC: *Scalable Urban Traffic Control***

Na *Carnegie Mellon University*, em Pittsburgh nos Estados Unidos encontra-se em desenvolvimento um sistema piloto chamado SURTRAC: *Scalable Urban Traffic Control* – um sistema de controle de tráfego urbano escalonável (Smith *et al.*, 2013).

Este sistema integra conceitos de teoria de controle de tráfego com trabalhos recentes na área de planejamento com multi-agentes. Num certo nível, o SURTRAC controla cada cruzamento de maneira independente, sendo capaz de receber e transmitir informações entre semáforos a respeito do tráfego a cada segundo. Ou seja, trata-se de um sistema que opera em tempo real (Smith *et al.*, 2013).

Esta troca de informações, que visa a diminuição do tempo total de espera dos veículos atualizada segundo-a-segundo, é realizada por uma rede de dispositivos interconectada de geração de itinerários (Smith *et al.*, 2013).

O sistema de controle de cada cruzamento é concebido no contexto de uma arquitetura orientada a serviços (do inglês SOA – *Service-Oriented Architecture*), como mostra a Figura 1 (Smith *et al.*, 2013).

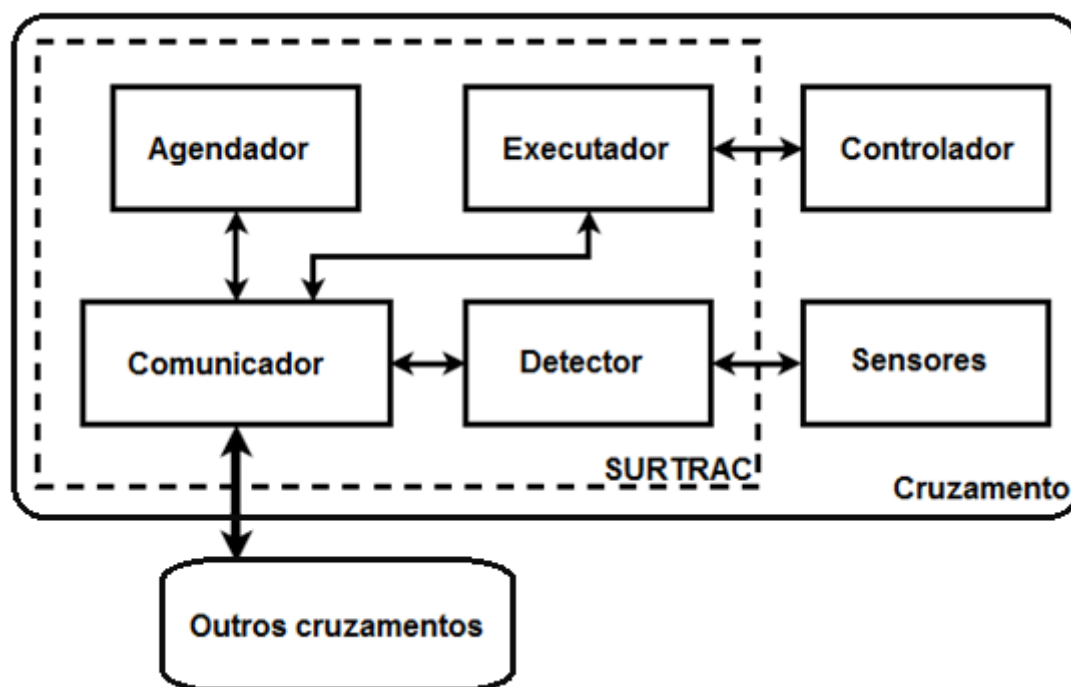


Figura 1 – Diagrama do sistema SURTRAC (Adaptado de Smith et al., 2013)

O serviço do “Comunicador” é lidar com o processamento de todas as informações entre diferentes serviços, assim como o compartilhamento de informações entre cruzamentos. O “Detector” é o serviço de interface com todos os sensores de veículos, e que processa informações em tempo real emitindo mensagens que podem ser usadas por serviços locais e remotos. O “Executador” administra a interface com o controlador de sinal de tráfego, lendo informações sobre o estado dos sinais e controlando a duração e sequência das **fases**. O “Agendador” usa dados de outros serviços para programar a distribuição de “tempo de verde” nos cruzamentos (Smith et al., 2013).

Para demonstrar o potencial do SURTRAC, um sistema piloto de nove cruzamentos “inteligentes” foi instalado na cidade de Pittsburgh, Pensilvânia. O espaço médio entre cruzamentos foi de 116 metros, de modo a exigir uma coordenação precisa entre os semáforos. A interligação entre estes semáforos foi feita por meio do uso de cabos de fibra ótica, necessários para o funcionamento do sistema em tempo real (Smith et al., 2013).

Para avaliar a efetividade do SURTRAC, doze rotas foram previamente desenvolvidas e testadas por diversos períodos do dia, durante todos os dias da semana, antes e após a instalação do sistema. Os resultados obtidos mostraram que o tempo gasto para percorrer cada rota diminuiu em média

25,8% após implantação do SURTRAC. O tempo de espera médio foi reduzido em 40,6% e as emissões de poluentes decaíram em 21,5% (Smith *et al.*, 2013).

Os resultados do sistema piloto demonstram de maneira convincente a efetividade e o potencial do controle de sinalização de tráfego descentralizado e adaptativo em trechos urbanos. O sistema SURTRAC passa por constantes processos de aperfeiçoamento, inclusive visando a otimização de tempo de espera de pedestres e a detecção e priorização de transporte público urbano nos cruzamentos (Smith *et al.*, 2013).

### **b) Sistema SCAT**

No início da década de 1980, na cidade de Sydney na Austrália, foi desenvolvido o sistema SCAT: *Sydney Coordinated Adaptive Traffic* – um sistema de coordenação de tráfego adaptável (Sims e Dobinson, 1980).

O SCAT é um sistema de controle de tráfego urbano sinalizado introduzido na cidade de Sydney. Este sistema consiste inteiramente de computadores e é completamente adaptável de acordo com a demanda de tráfego. O sistema possui onze centrais computadorizadas de supervisão distribuídas em uma área de 1500km<sup>2</sup> ao redor da cidade australiana, que são responsáveis por monitorar mais de 1000 controladores (microcomputadores) instalados em inúmeros cruzamentos. A comunicação do sistema é quase que exclusivamente feita através de linhas telefônicas alugadas (Sims e Dobinson, 1980).

A inteligência dos microcomputadores é utilizada para processar dados estratégicos coletados por meio do uso de detectores de veículos, tomar decisões táticas a respeito da operação dos semáforos e perceber o mal funcionamento dos detectores (Sims e Dobinson, 1980).

Estes detectores armazenam informações a respeito do tempo decorrido entre a detecção de veículos e o tempo que cada veículo permaneceu sob um mesmo detector. Como também possuem dados referentes à distância que há entre os detectores, são capazes de calcular a velocidade média do tráfego em um determinado cruzamento (Sims e Dobinson, 1980). As principais variáveis consideradas para a realização deste cálculo estão indicadas na Figura 2.

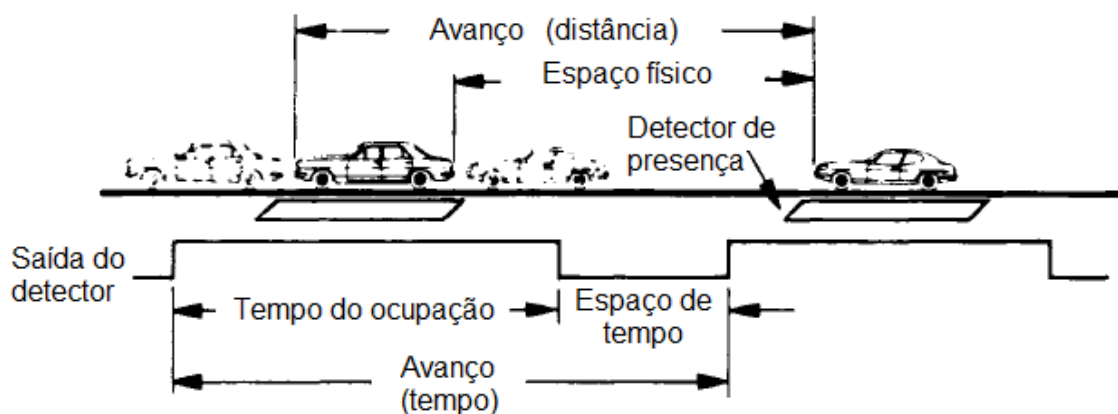


Figura 2 - Relacionamento entre tempo e espaço entre veículos (Adaptado de Sims e Dobinson, 1980)

Os dados coletados são então comparados com uma extensa base amostral localizada na memória dos microcomputadores. Com base nesta comparação é possível determinar se o tráfego está se comportando da maneira esperada ou não (Sims e Dobinson, 1980).

Cada controlador possui quatro programações distintas instaladas em sua base de dados, que são ativadas automaticamente, de acordo com o resultado da análise da situação de velocidade e espaçamento dos veículos, com o intuito de melhorar as condições de tráfego de uma região de cruzamentos (Sims e Dobinson, 1980).

Testes preliminares do sistema SCAT foram realizados em 1974 em uma avenida arterial de Sydney. Os resultados destes testes indicaram uma redução média de tráfego de: 39,5% no período matutino; 14,5% no período entre picos; e 32,8% no período vespertino (Sims e Dobinson, 1980).

O sistema SCAT é atualmente utilizado em 27 países e em mais de 37,000 cruzamentos ao redor do mundo (New South Wales Government, 2015).

Tanto o sistema SCAT quanto o sistema SURTRAC mostram ser possível reduzir o tráfego de maneira significativa em centros urbanos por meio da aplicação de tecnologias que permitem identificar as diferentes situações do tráfego e a devida coordenação na atuação dos semáforos.

Confirma-se assim a necessidade de um estudo mais detalhado do fluxo de veículos característicos de cada cidade para melhor explorar estas tecnologias.

## 2.2. Modelagem de sistemas

### 2.2.1. Sistemas a eventos discretos

Sistemas a eventos discretos são aqueles nos quais as mudanças de estado acontecem exclusivamente devido à ocorrência de eventos em qualquer ponto discreto no tempo. Tais eventos podem ser ativados em consequência de *delays* e em momentos específicos que alteram o estado do sistema (Banks, 1999).

Em sistemas a eventos discretos, as variáveis necessárias para determinar o estado presente se mantêm constantes e só mudam de valor quando ocorre um evento, enquanto em um sistema de tempo contínuo as variáveis mudam continuamente ao longo do tempo (Banks, 1999).

### 2.2.2. Rede de Petri

Rede de Petri é uma técnica utilizada para a modelagem e análise de sistemas a eventos discretos. Nela, os sistemas são descritos por elementos como: Lugares, que representam condições e são representados graficamente por círculos; Transições, ou seja, eventos que podem ocorrer, que são representadas graficamente por barras; Arcos, que são representados graficamente por setas e que podem ser de entrada, que ligam um lugar a uma transição, ou de saída, que ligam uma transição a um lugar. Os arcos também são rotulados com seus pesos, onde um arco com peso  $k$  pode ser interpretado como  $k$  arcos paralelos (Murata, 1989).

Há ainda a marcação com tokens nos lugares, que define o estado de uma rede de Petri. A movimentação desses tokens é responsável pela representação da dinâmica do sistema e é estabelecida por meio do disparo de transições, que descreve mudança de estados. Mais informação sobre as regras de disparo das transições pode ser encontrada em Murata (1989) e Miyagi (1996).

A rede de Petri tem propriedades comportamentais que devem ser estudadas. Algumas dessas propriedades são a alcançabilidade, a limitabilidade e a segurança (Murata, 1989).

Um estado  $M_n$  é alcançável a partir de  $M_0$  se existe uma sequência de disparos que fazem com que, a partir do estado  $M_0$ , o estado  $M_n$  seja atingido. Uma rede é dita  $k$ -limitada quando o número de tokens em cada lugar da

mesma nunca ultrapassa o número finito  $k$ . Uma rede é chamada segura quando a mesma é 1-limitada, ou seja, nunca há mais de um token em qualquer lugar da mesma (Murata, 1989).

Outro conceito importante em rede de Petri é o deadlock. Este consiste em uma situação de travamento em que, ao ser atingida, não é possível disparar nenhuma transição, sendo assim, na maioria dos casos, uma situação indesejável que deve ser evitada (Murata, 1989).

Ao longo do tempo, surgiram diversas extensões desta técnica, como a rede de Petri temporizada, na qual existem transições que disparam após o decorrimento de um tempo definido; a rede de Petri temporizada sincronizada (STPN), na qual é possível acionar transições com sinais externos à rede; a rede de Petri temporizada colorida (CTPN), na qual cada token tem um valor; e a rede de Petri híbrida, que utiliza variáveis contínuas, além de estados discretos.

### **2.2.3. Relação entre rede de Petri e LD**

Uma das vantagens da rede de Petri é que um modelo (grafo) que representa o comportamento de um sistema pode ser usado para especificar o que este sistema deve executar. Isto é, adotando a abordagem de "seguidor de modelo" um sistema de controle realiza suas funções de acordo com a evolução do modelo em rede de Petri do objeto de controle, ou seja, o modelo em rede de Petri é o programa de controle do sistema. Neste contexto, existem métodos para a conversão de um modelo de rede de Petri para um programa em linguagens existentes de controladores programáveis, como é a linguagem LD (Miyagi *et al.*, 2011).

A linguagem LD (*Ladder Diagrams*) é a mais amplamente utilizada na programação de controladores programáveis. Nela, a lógica é representada por contatos e bobinas (Lucas and Tilbury, 2005).

Um método de conversão de modelos em rede de Petri para LD foi proposto por Miyagi *et al.* (2011). Segundo tal método, deve-se:

- 1) Criar variáveis auxiliares, que representam todos os estados dos lugares e transições existentes na modelagem em rede de Petri.
- 2) Cada transição deve ser definida por uma bobina e associada a uma linha. Os lugares de entrada de uma transição são denotados por contatos

normalmente abertos, enquanto os lugares de saída são representados por contatos normalmente fechados. As variáveis de entrada associadas a essa transição também são definidas por contatos na mesma linha.

3) Para cada transição criada, conecta-se uma bobina de *reset* para cada arco de entrada e uma bobina de *set* para cada arco de saída.

4) Para a ativação das saídas, uma bobina comum é criada para cada variável de saída. Os lugares associados a uma variável são definidos como contatos normalmente abertos.

5) Define-se as condições iniciais do programa. Cria-se uma linha de comando onde todos os lugares são contatos normalmente fechados e cada lugar com marcação inicial tem uma bobina de *set* correspondente.

#### **2.2.4. Softwares para modelagem e análise**

Existem vários *softwares* que auxiliam no desenvolvimento e análise de modelos em rede de Petri. Conforme aumenta-se a complexidade do modelo, este torna-se mais difícil de analisar. Assim, a utilização de um *software* torna-se uma boa alternativa.

Um exemplo de *software* que está disponível ao público para edição e análise de modelos em rede de Petri é o PIPE2 (*Platform-Independent Petri Net Editor 2*). Nesse programa, os modelos são construídos com as opções disponíveis em uma barra de ferramentas que inclui lugares, transições, arcos e tokens. Além disso, o *software* permite a realização de animações para auxiliar o usuário a verificar o comportamento dos seus modelos. O PIPE2 ainda conta com módulos de análise especializados que realizam análises de desempenho e estruturais da rede de Petri, sendo ainda possível que o usuário implemente seus próprios módulos para adaptar a funcionalidade da ferramenta aos seus requisitos (Dingle *et al.*, 2009).

Outro *software* disponível ao público é o HPSim, que é capaz de gerar relatórios no formato CSV (*comma-separated values*) com os resultados das simulações nele realizadas. Arquivos no formato CSV são compatíveis com o *software* Microsoft Excel e, com isso, podem ser facilmente analisados e organizados para extrair as informações da simulação da forma desejada (HP, 2006).

### **2.3. Sistemas de semáforos e modelagem dos mesmos em rede de Petri**

A capital paulista conta com mais de 6.000 cruzamentos com semáforos instalados (CET, 2013). Para cada tipo de cruzamento existente na cidade, há uma configuração diferente para os semáforos do mesmo. Como a própria CET - Companhia de Engenharia de Tráfego - informa, a operação semafórica em São Paulo é elaborada por uma equipe técnica especializada, com o intuito de atender da melhor maneira possível aos perfis de tráfego do local em condições de normalidade (Christianini e Hagiwara, 2008). Há também um projeto em andamento da CET de substituição de semáforos antigos em 4.800 cruzamentos da cidade por semáforos modernos com *no-breaks* e controladores que garantam seu funcionamento mesmo sob falta momentânea de energia (CET, 2013). Com exceção de aproximadamente 1.000 cruzamentos, que possuem um sistema semafórico chamado "inteligente" (semáforos atuados pelo tráfego) o restante possui apenas uma configuração de funcionamento que não se altera ao decorrer do dia (Gomes, 2014).

Entretanto, diversos estudos (como por exemplo: sistema SURTRAC, sistema SCAT) confirmam um resultado geral melhor quando os semáforos dos cruzamentos alteram seus tempos de controle de acordo com o tráfego corrente (Sims e Dobinson, 1980; Smith *et al.*, 2013). Várias técnicas de inteligência artificial também têm sido estudadas para otimizar fluxos de tráfego (Robertson e Bretherton, 1991).

Uma abordagem de sistema híbrido, composto tanto por variáveis contínuas como por eventos discretos, de controle semafórico foi adotada na Universidade de Tecnologia de Beijing (Zhang *et al.*, 2008). Neste caso, demonstra-se que um modelo de rede de Petri híbrida pode lidar com diversas situações diferentes (como um acidente de carro), além de ser capaz de replicar o mesmo modelo para inúmeros tipos de cruzamentos distintos, exigindo menor complexidade computacional.

Para a análise de tráfego em cruzamentos, uma modelagem derivada da rede de Petri também foi adotada por Huang *et al.* (2014), a STPN (*Synchronized Timed Petri Net*) – rede de Petri temporizada sincronizada – para modelagem de um sistema de controle de semáforos de oito, seis ou duas

**fases.** O STPN permite dois tipos de transição: a imediata, também chamada de transição escrava, que ocorre de maneira instantânea; e a determinística, cujo acionamento sempre envolve um *delay* de tempo pré-determinado.

Os trabalhos já publicados em relação a este assunto mostram que a modelagem de sistemas semafóricos por rede de Petri não é só possível, como já foi adotada inúmeras vezes para situações específicas e com resultados positivos. Em combinação com informações específicas do trânsito paulistano, entende-se assim que é possível desenvolver um controle para as características de tráfego e cruzamentos aqui presentes.

## 2.4. Metodologia de projeto de sistemas de controle

Uma metodologia geral de projeto de sistemas de controle de SEDs é sugerida por Miyagi (1996). Ela considera uma série de etapas para o projeto e desenvolvimento de um sistema de controle (Figura 3).

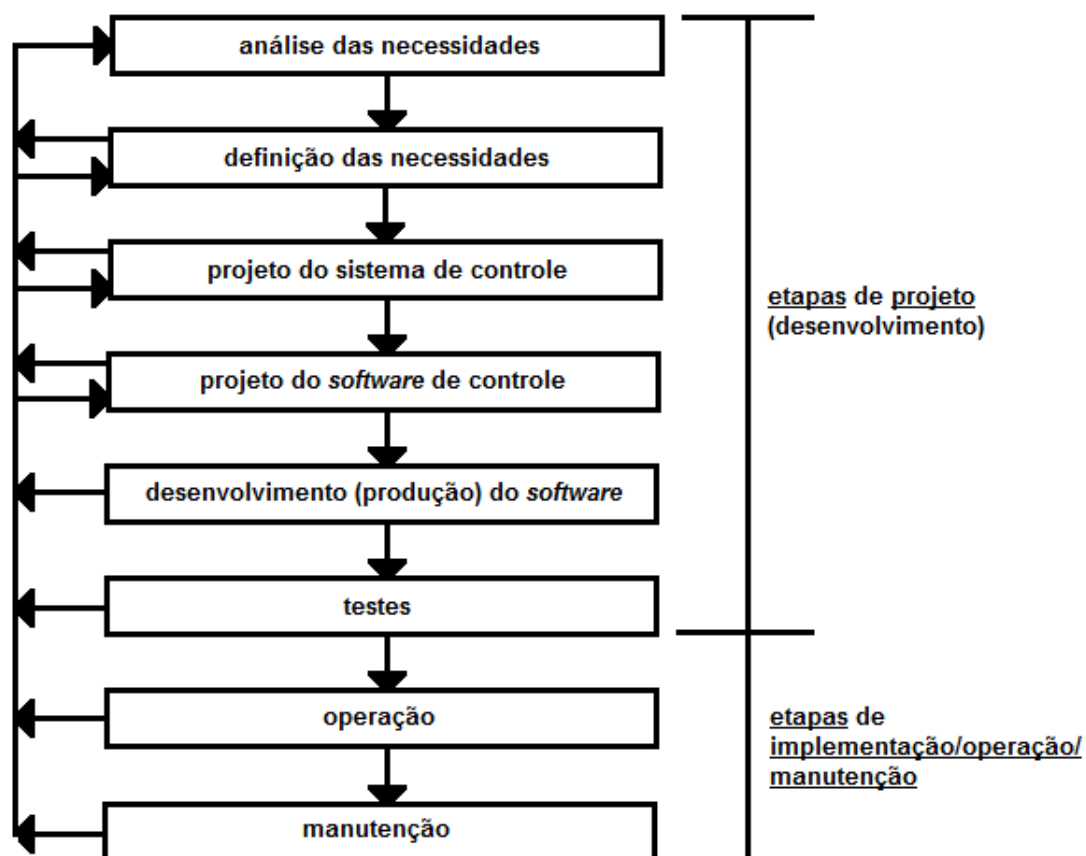


Figura 3 - Metodologia de projeto de sistemas de controle (Adaptado de Miyagi, 1996)

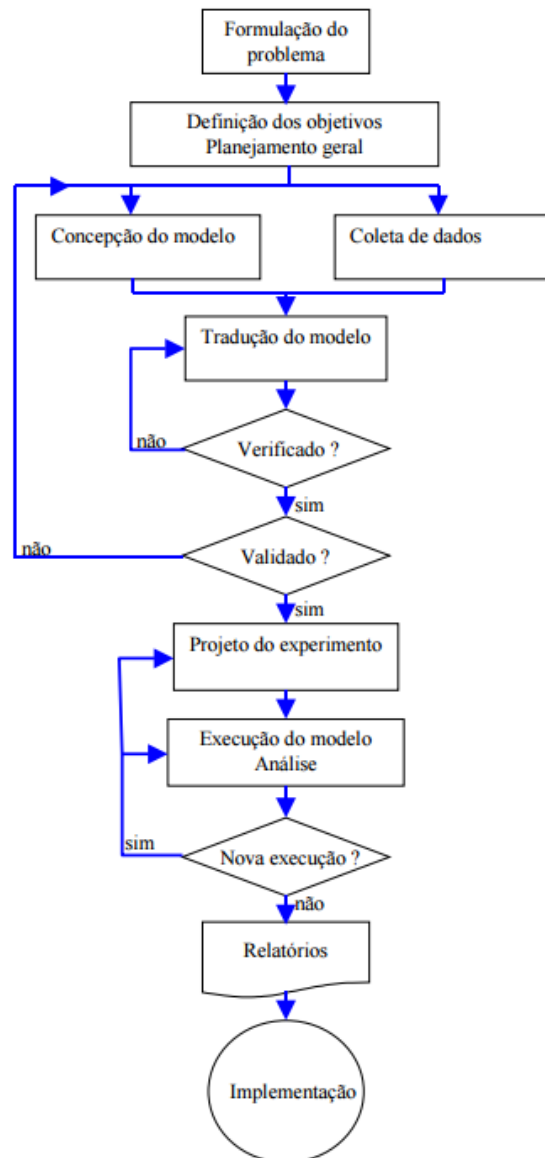
A metodologia define e organiza o que deve ser feito em cada uma dessas etapas. Na etapa de "análise das necessidades", deve ser feita a identificação do objetivo final do sistema; a compreensão do objeto de controle, instalações e equipamentos; a organização dos conhecimentos sobre o sistema de controle; e a abstração e análise das funções de controle. A próxima etapa é a "definição das necessidades" e nela devem ser realizadas as definições das funções de controle e do fluxo destas (Miyagi, 1996).

A etapa de "projeto do sistema de controle" é composta pela divisão das funções e definição das interfaces; pela definição e alocação dos sinais de entrada e saída; e pela definição da estrutura do programa de controle. Tal etapa é seguida pelo "projeto do *software* de controle", que, por sua vez, consiste nos projetos da reutilização, dos programas e de programas não padronizados (funções ainda não catalogadas como módulos funcionais padrões).

Por fim, a etapa de "desenvolvimento do *software*" consiste no desenvolvimento do programa e seu carregamento nas máquinas. Após isso, realiza-se o teste por unidade e, finalmente, o teste do sistema, que compõem a etapa de "testes". Outras etapas estão diretamente relacionadas à implementação, operação e manutenção do sistema de controle. Maior detalhamento da metodologia pode ser encontrado em Miyagi (1996).

Essa metodologia tem eficácia comprovada e é aplicada no projeto e implementação de diversos sistemas de controle de SED na área de automação. A aplicação prática indica que a metodologia garante o cumprimento do cronograma previsto e das especificações técnicas mesmo com equipes pouco experientes em projeto de sistemas de controle (MIYAGI, 1996).

Um procedimento para a análise de sistemas via técnicas de simulação é também proposto por Miyagi (2006). O fluxograma desse procedimento é ilustrado na Figura 4. Para a obtenção de resultados satisfatórios em uma análise por simulação, é fundamental que tal procedimento seja respeitado.



*Figura 4 - Procedimento para análise de sistemas por simulação (Reproduzido de Miyagi, 2006)*

As metodologias propostas nas Figuras 3 e 4 são complementares. Identifica-se uma **fase** preliminar de concepção, uma **fase** de desenvolvimento e uma **fase** de testes e implementação, cada qual com condições que podem exigir o reprojeto, caso não cumpridas. Isso é aplicável a um projeto de engenharia e está de acordo com o objetivo deste trabalho, podendo ser utilizado como base para sua realização.

É necessário ressaltar, entretanto, que o escopo desse trabalho se estende apenas até a etapa de “projeto do *software* de controle”, descrita na Figura 3. As etapas de implementação e operação não serão executadas.

### 3. Análise de requisitos

O foco deste trabalho está na modelagem e análise de cruzamentos semaforicos utilizando-se rede de Petri. São estudadas técnicas de modelagem de cruzamentos já empregadas por outros autores. Estas são utilizadas como base para escolher possíveis soluções e montar uma matriz de decisão capaz de apontar a melhor abordagem a ser utilizada.

O trabalho envolve o uso de softwares de modelagem de redes de Petri, como por exemplo o PIPE2 (Platform-Independent Petri Net Editor 2) e o HPSim para a criação dos modelos. A análise e validação dos modelos desenvolvidos deverá ser feita a partir dos resultados gerados pelos *softwares* quando possível. Os modelos desenvolvidos em rede de Petri devem ter a possibilidade de conversão para a linguagem LD seguindo o método proposto por Miyagi *et al.* (2011), de forma que sua implementação em controladores programáveis seja possível.

Os modelos devem ter uma configuração inicial em que todos os semáforos do cruzamento estão em **estágio** vermelho. Com isso, fica possível a implementação de botões de *reset* e *start* no controlador, que fazem, respectivamente, o sistema ir para uma configuração inicial conhecida e iniciar o funcionamento. Todos os cruzamentos devem partir desta configuração inicial até que atinjam o ciclo de funcionamento. Deve-se seguir um método padronizado para o desenvolvimento dos modelos, de forma que os modelos de cruzamentos gerais sigam uma mesma lógica e sejam adaptáveis para casos específicos sem grande esforço.

Outro requisito necessário é que haja possibilidade de alteração do tempo de **estágio** verde dos semáforos de forma intuitiva, fazendo com que esse tempo possa ser modificado manualmente para regular o trânsito de acordo com a demanda ao longo do dia ou no caso de uma ocorrência especial, como um acidente. Além desses requisitos mínimos, é desejável que seja possível, com os modelos, conectar um conjunto de cruzamentos de forma que os mesmos operem de forma sincronizada. Busca-se que alterações no tempo do **ciclo** dos semáforos em um cruzamento se propaguem automaticamente para os cruzamentos que dependem do mesmo.

## 4. Técnicas de modelagem

### 4.1. Possíveis abordagens

Chen *et al.* (2014) modelou um cruzamento de quatro **fases** básicas onde também existem mais quatro **fases** adicionais opcionais para aliviar a pressão do tráfego em cada sentido do fluxo de veículos. Em todos os sentidos, há semáforos distintos para seguir em frente e para realizar as conversões à direita e à esquerda. A modelagem do cruzamento foi feita utilizando a rede de Petri temporizada. O modelo obtido pode ser visto na Figura 5.

Na abordagem de Chen *et al.* (2014), cada **fase** é modelada por um lugar. Isso torna a modelagem mais simples e intuitiva mesmo ao se tratar de cruzamentos de elevada complexidade. Entretanto, isso dificulta a visualização dos **estágios** de cada semáforo.

Essa modelagem está simplificada de modo que considera-se que os semáforos só têm um **estágio** vermelho e um verde, que se inicia imediatamente quando o sinal vermelho de um semáforo adjacente foi ativado. Um problema nesse tipo de abordagem é o fato de a mesma não levar em consideração os tempos em que alguns semáforos estão no **estágio** amarelo. Também não são considerados os pequenos tempos de transição em que todos os semáforos estão no **estágio** vermelho por segurança, o que ocorre sempre antes da ativação de qualquer **estágio** verde.

Além disso, como nesse modelo cada lugar representa uma **fase**, e não um sinal específico de cada semáforo, a cada disparo de transição, sinais de vários semáforos são ativados ou desativados. Esses problemas citados causam dificuldades na conversão do modelo e implementação em um controlador programável que na prática deve considerar possíveis atrasos na transição de sinais.

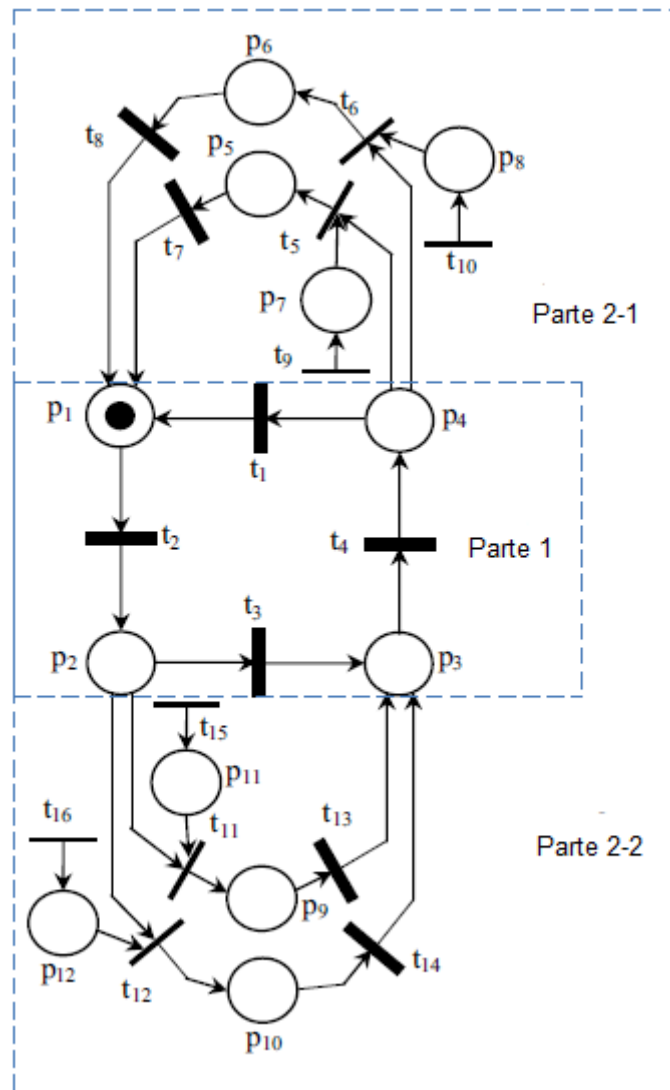


Figura 5 - Modelo em rede de Petri realizado por Chen et al. (Adaptado de Chen et al., 2014)

Na Figura 5, a "Parte 1" representa o cruzamento original de quatro **fases** (P1, P2, P3, P4). Cada **fase** está modelada por um lugar. A "Parte 2-1" e a "Parte 2-2" são semelhantes e representam as quatro **fases** adicionais (P5, P6, P9, P10) que podem ser ativadas para equilibrar o fluxo de veículos entre sentidos opostos (Chen et al., 2014).

Em Zhang et al. (2008) uma outra abordagem para a modelagem é idealizada. Ao se considerar um cruzamento em uma avenida, deve-se levar em consideração também o cruzamento subsequente, para que o instante de tempo entre o início do **estágio** verde de um cruzamento em relação ao outro, seja equivalente àquele que o veículo demora para percorrer o trecho entre os semáforos (Zhang et al., 2008).

Para Zhang *et al.* (2008) a modelagem ideal de um cruzamento semafórico de duas ou quatro **fases** deve ser realizada com a rede de Petri híbrida, ou seja, uma rede de Petri que combina variáveis contínuas e eventos discretos em um mesmo modelo. A passagem de veículos pelo cruzamento é descrita por meio de uma variável contínua, enquanto que a transição das **fases** é vista como um evento discreto. Nesta abordagem, cada **fase** é modelada como um lugar.

Na Figura 6 é possível observar a modelagem para um cruzamento de duas **fases** desenvolvida por Zhang *et al.* (2008). A parte central do diagrama representa a transição entre **fases** dos semáforos. A porção esquerda retrata o fluxo de veículos que percorre o caminho da avenida, enquanto que a porção direita da figura representa os veículos que atravessam o cruzamento (Zhang *et al.*, 2008).

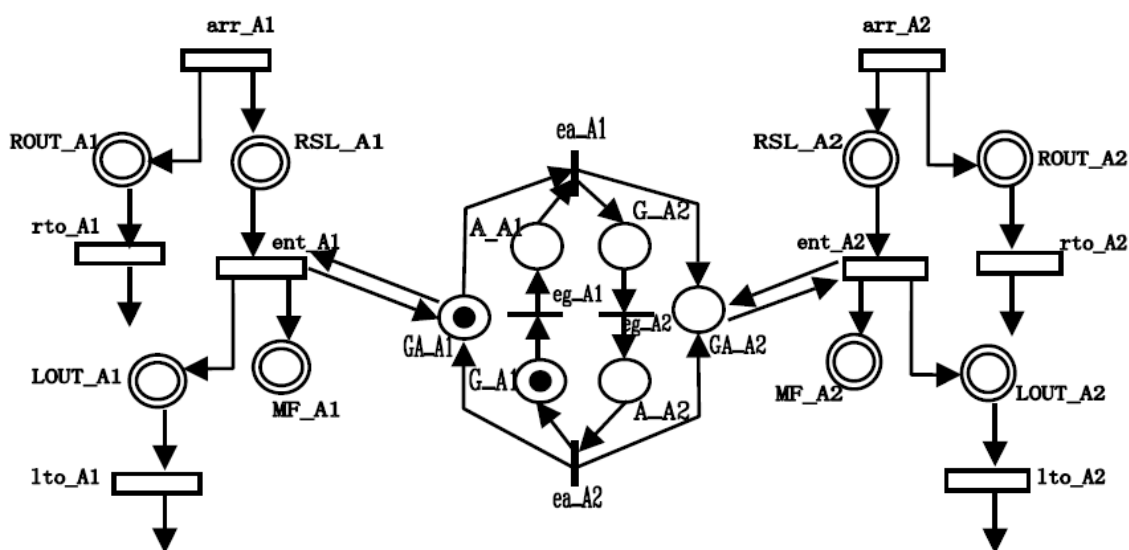


Figura 6 - Modelo em rede de Petri híbrida de um cruzamento de duas **fases** realizado por Zhang *et al.* (Reproduzido de Zhang *et al.*, 2008)

O significado de cada lugar pode ser visto na Tabela 1, enquanto que o significado de cada transição se encontra na Tabela 2.

**Tabela 1 – Lugares da Figura 6** (Adaptado de Zhang *et al.*, 2008)

Lugar	Descrição
GA-Ai	Período verde ou amarelo da i-ésima fase
G-Ai	Período verde da i-ésima fase
A-Ai	Período amarelo da i-ésima fase
ROUT-Ai	Veículos saem à direita
LOUT-Ai	Veículos saem à esquerda
RSL-Ai	Veículos prontos para ir em frente ou virar à esquerda
MF-Ai	Veículos movendo-se para frente

**Tabela 2 – Transições da Figura 6** (Adaptado de Zhang *et al.*, 2008)

Lugar	Descrição
eg-Ai	Término da i-ésima fase verde e início da fase amarela
ea-Ai	Término da i-ésima fase
arr-Ai	Veículos chegam ao cruzamento
ent-Ai	Veículos entram no cruzamento
rto-Ai	Veículos saem à direita
lto-Ai	Veículos saem à esquerda

Huang *et al.* (2014) adotou uma abordagem diferente. Cada **estágio** (verde, amarelo, vermelho) de cada semáforo foi modelado por um lugar. As transições são temporizadas com as durações dos respectivos **estágios**. Isso torna a rede relativamente mais complexa e de difícil visualização. Porém, esse tipo de aproximação garante maior aplicabilidade em controladores programáveis e abrangência, por poder controlar cada semáforo individualmente (Huang *et al.*, 2014).

Para realizar a modelagem, Huang *et al.* (2014) utilizou a STPN (*Synchronized Timed Petri Net*). Com este tipo de rede de Petri, além de ser possível definir transições temporizadas, pode-se também ter transições que são disparadas por eventos externos. Assim, é possível, por exemplo, definir como um evento o disparo de uma transição em uma determinada rede e utilizá-lo como evento de disparo de uma transição em outra rede,

sincronizando as duas redes. Assim, a STPN é adequada para a modelagem de sistemas que são sincronizados com eventos externos (Huang *et al.*, 2014).

Huang *et al.* (2014) utiliza um sistema *master-slave* para semáforos de uma mesma via e fluxos em sentidos opostos, no caso em que eles têm exatamente os mesmos tempos de cada **estágio**.

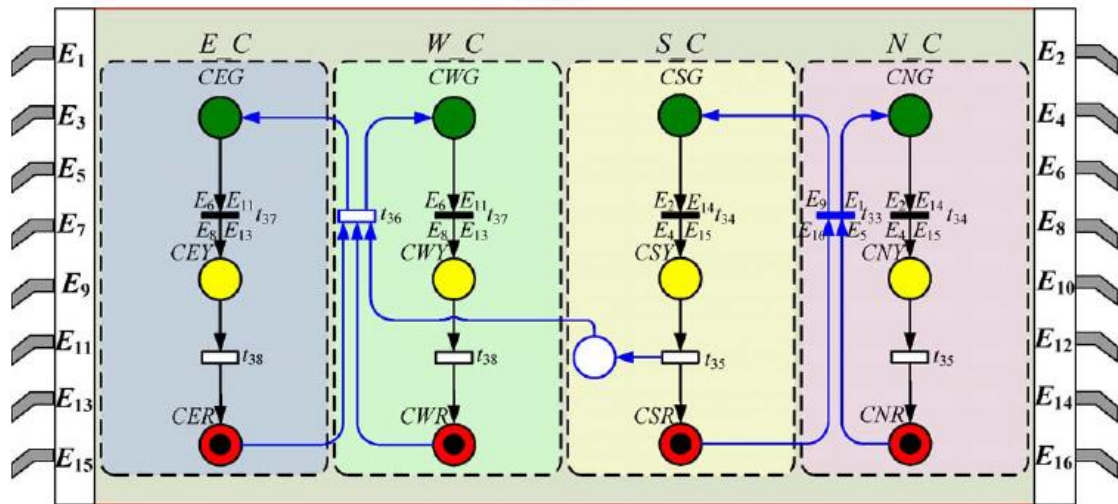


Figura 7 - Modelagem de um cruzamento de duas **fases** utilizando STPN e sistema *master-slave* (Reproduzido de Huang *et al.*, 2014)

Nessa abordagem, modela-se nove cruzamentos interligados. Um deles é um cruzamento de oito **fases** e é considerado o *master*, enquanto os outros (*slaves*) são cruzamentos de duas ou seis **fases**. Para cada cruzamento individual é desenvolvida uma rede de Petri e são definidos os sinais de entrada e saída, como pode ser visto na Figura 7. Após isso, conectam-se os grafos por meio dos sinais definidos, como ilustrado na Figura 8. Nesta, o nome dos módulos (*Class A*, *Class B* e *Class C*) representam o número de **fases** do cruzamento em questão, em que *Class A* é um cruzamento de oito **fases**, *Class B* é um cruzamento de seis **fases** e *Class C* é um cruzamento de duas **fases** (Huang *et al.*, 2014).

A abordagem realizada por Huang *et al.* (2014) mostra-se uma boa solução quando deseja-se interligar e sincronizar um conjunto de cruzamentos (Huang *et al.*, 2014).

De acordo com Huang *et al.* (2014), muitas contribuições anteriores apresentam problemas de implementação em um sistema real, enquanto a STPN tem uma grande aplicabilidade na lógica de controle de sinais.

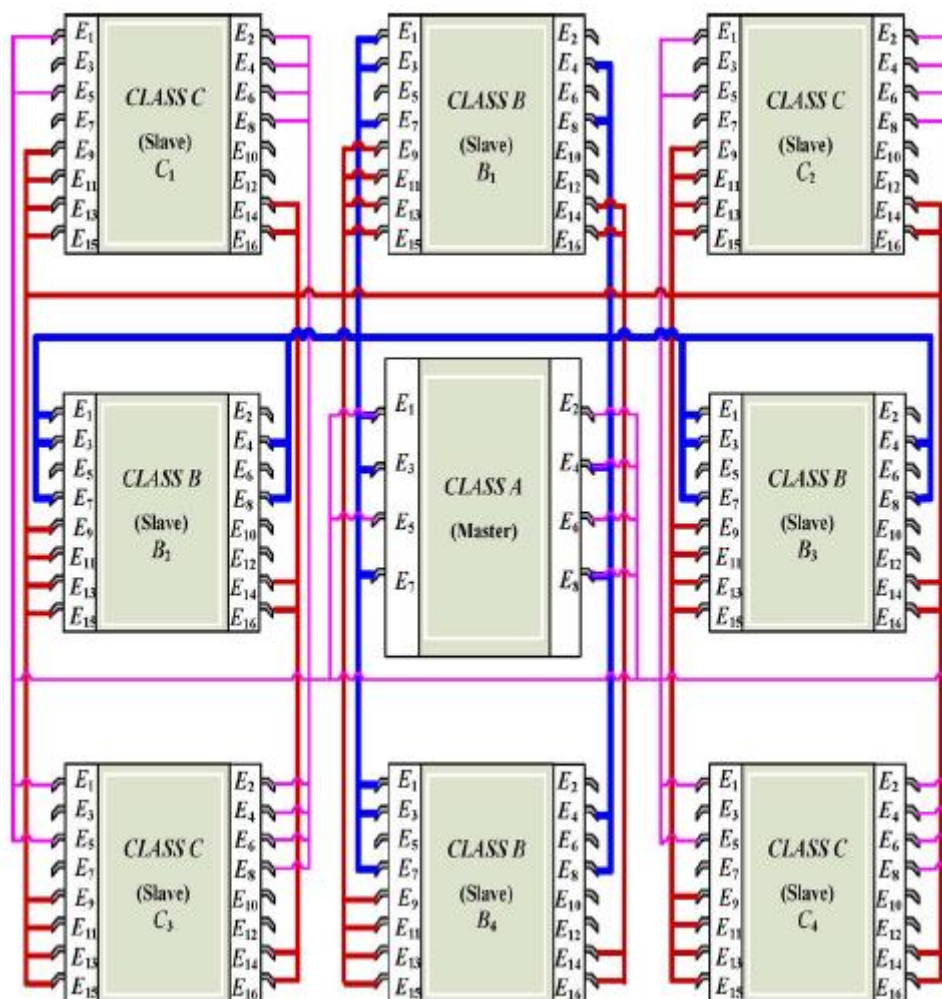


Figura 8 - Conexão de sinais externos para sincronização das redes de Petri (Reproduzido de Huang *et al.*, 2014)

Uma outra abordagem desenvolvida por Dotoli e Fanti (2006) faz uso da chamada CTPN: *Coloured Timed Petri net* – uma rede de Petri temporizada colorida. Nesta abordagem os veículos são vistos como tokens dentro dos lugares, sendo que a coloração de cada veículo/token é definida pela rota que eles realizarão (Dotoli e Fanti, 2006).

Neste trabalho cada local que um veículo pode ocupar em uma via com cruzamentos semafóricos, é modelado por um lugar. Nesta definição estão incluídos: locais no meio de cruzamentos, locais logo antes e depois de cruzamentos e locais em trechos entre cruzamentos. Um esquema que

exemplifica a modelagem de Dotoli e Fanti (2006) pode ser visualizado na Figura 9.

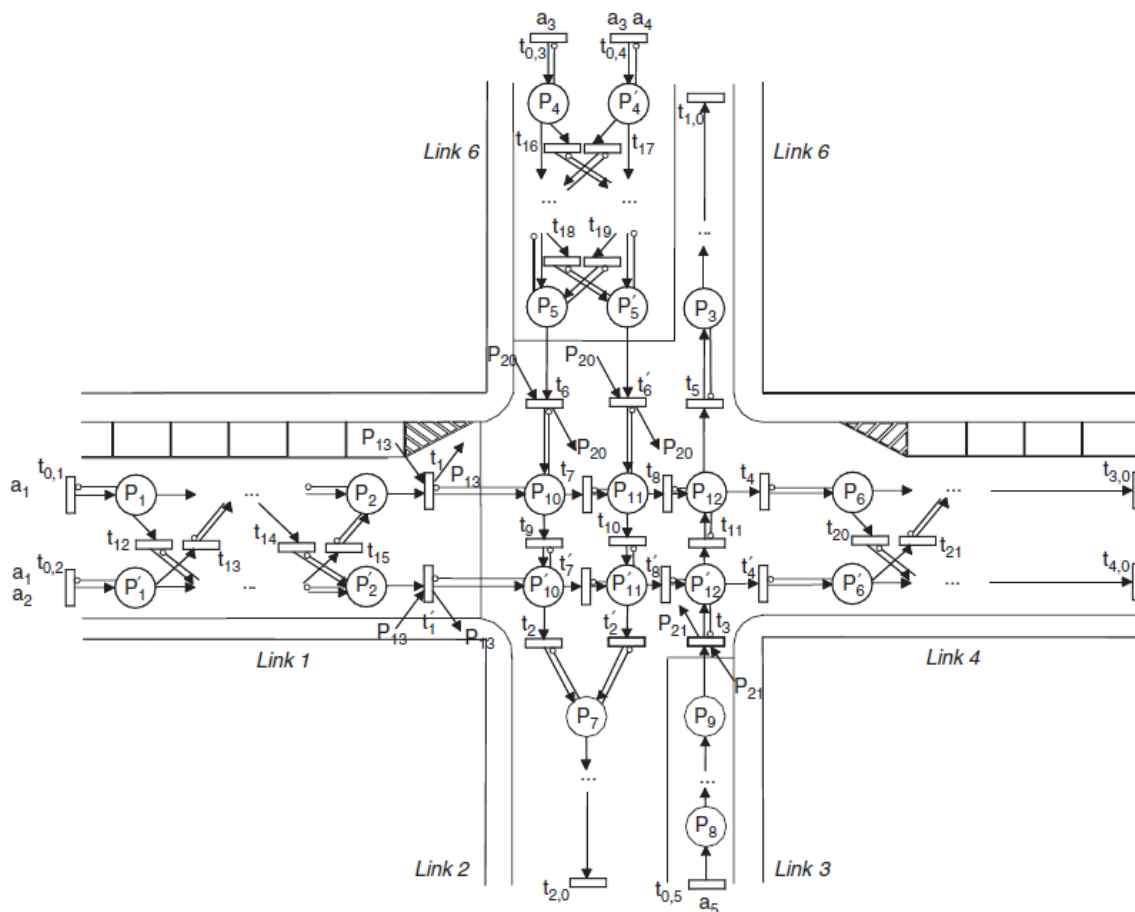


Figura 9 – Exemplo de cruzamento para modelagem com rede de Petri temporizada colorida (Reproduzido de Dotoli e Fanti, 2006)

Para a região dos cruzamentos em si, assim como em Zhang *et al.* (2008), são definidos lugares para cada estágio dos semáforos (verde, amarelo e vermelho) e transições temporizadas são usadas para definir o tempo em cada estágio (Dotoli e Fanti, 2006).

A abordagem de Dotoli e Fanti (2006) é validada por meio de uma comparação dos resultados de um modelo CTPN desenvolvido de um cruzamento real, com informações já coletadas sobre o mesmo.

## 4.2. Matriz de decisão

De maneira a decidir qual a melhor abordagem dentre as anteriormente citadas para a criação de modelos de cruzamentos semafóricos a ser aplicada neste

trabalho, desenvolveu-se uma matriz de decisão. As alternativas podem ser definidas do seguinte modo:

- **Alternativa 1** – Modelagem de **fases** com rede de Petri temporizada (Chen *et al.*, 2014);
- **Alternativa 2** – Modelagem de **fases** com rede de Petri híbrida (Zhang *et al.*, 2008);
- **Alternativa 3** – Modelagem de **estágios** com STPN (Huang *et al.*, 2014);
- **Alternativa 4** – Modelagem de **estágios** com CTPN (Dotoli e Fanti, 2006).

#### 4.2.1. Critérios da matriz de decisão

Além das alternativas, a matriz de decisão também é formada por critérios, cuja função é avaliar de maneira quantitativa qual das alternativas se sobressai em relação às outras.

As notas dadas para cada alternativa em relação a um determinado critério, possuem valores inteiros entre zero e cinco. Os critérios em si também têm peso próprio, variando de um a quatro. Os pesos de cada critério foram atribuídos de acordo com sua importância relativa a este trabalho. Os seguintes critérios, seguidos de seus respectivos pesos, foram definidos para compor a matriz de decisão deste trabalho:

- **Abrangência (Peso 3)** – Avalia a variedade de tipos de cruzamento distintos que a alternativa é capaz de modelar (por exemplo, quanto à quantidade de **fases** e possibilidade de incluir semáforos para pedestres e ciclovias);
- **Aplicabilidade (Peso 4)** – Analisa a facilidade de conversão do modelo para linguagens utilizadas em controladores programáveis, verificando se a alternativa apresentada apresenta dificuldades, ou até mesmo impossibilidades na conversão. A conversão do modelo trata-se de um requisito do projeto, e portanto este é um critério fundamental;
- **Detalhamento (Peso 2)** – Avalia o nível de detalhamento dado às alternativas propostas em cada artigo. Um bom nível de detalhamento é essencial para que o modelo seja compreendido de maneira apropriada;

- **Resultados (Peso 1)** – Este critério analisa os resultados obtidos por cada alternativa e dá uma nota com base na eficiência da modelagem dos cruzamentos alcançada por cada uma delas. Trata-se de um critério com alto grau de subjetividade e, portanto, recebe um peso menor;
- **Simplicidade (Peso 2)** – Este critério avalia a simplicidade que se pode atingir ao modelar um cruzamento de alta complexidade, com **fases** múltiplas e semáforos para pedestres, como é frequente em cidades grandes;
- **Interconectividade (Peso 2)** – Analisa a capacidade da solução de interligar cruzamentos dependentes, controlando-os de forma síncrona de modo a reduzir o tempo de parada dos veículos;
- **Adaptabilidade (Peso 3)** – Avalia a facilidade de alteração rápida dos parâmetros do modelo, como os tempos dos **estágios** de cada semáforo. Leva em consideração ainda a capacidade de adaptar o modelo a uma nova situação (por exemplo, a adição ou remoção de **fases** ao cruzamento e a proibição ou liberação de uma conversão) sem necessidade de realizar uma modelagem completamente nova.

**Tabela 3 – Matriz de decisão**

		Alternativas			
<b>Critérios</b>	<b>Peso</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
<b>Abrangência</b>	3	4x3	3x3	3x3	2x3
<b>Aplicabilidade</b>	4	4x4	2x4	5x4	3x4
<b>Detalhamento</b>	2	3x2	2x2	4x2	4x2
<b>Resultados</b>	1	3x1	0x1	0x1	5x1
<b>Simplicidade</b>	2	5x2	3x2	3x2	2x2
<b>Interconectividade</b>	2	2x2	3x2	5x2	3x2
<b>Adaptabilidade</b>	3	1x3	1x3	2x3	2x3
<b>Total</b>	17	54	36	59	47

#### 4.2.2. Avaliação da matriz resultante

A alternativa com melhor resultado considera a utilização de rede de Petri temporizada sincronizada (STPN), com uma abordagem de projeto modular e configuração *master-slave*, conforme a solução de Huang *et al.* (2014).

Esta alternativa recebeu as notas mais altas nos critérios de aplicabilidade e interconectividade. A solução de interconectividade pode ser visto na figura 8, em que oito cruzamentos semaforicos são conectados a um cruzamento mestre, funcionando de forma síncrona com este. De acordo com Huang *et al.* (2014), "as vantagens da solução proposta são a apresentação clara do comportamento do sistema e a prontidão para implementação". Sendo assim, visto que a facilidade de implementação foi um dos focos do trabalho em questão, esta alternativa apresenta alta aplicabilidade.

O projeto modular consiste em dividir um problema em partes menores que possam ser tratadas independentemente e depois utilizadas em outros sistemas (Huang *et al.*, 2014). Sendo assim, essa alternativa satisfaz o requisito do projeto de padronização dos modelos, pois na implementação em um novo cruzamento, basta utilizar módulos já existentes com pouca necessidade de modificações. Outro requisito para o modelo é que o mesmo possa ser implementado em um controlador programável de forma padronizada. A alternativa escolhida promove a prontidão para implementação e, portanto, também satisfaz esse requisito.

#### **4.3. Validação do modelo**

O modelo é verificado inicialmente com um grafo de alcançabilidade (da rede de Petri), para verificar se todos os estados podem ser atingidos e se não há algum deadlock. Além disso, será feita a análise de espaço dos estados, que pode mostrar se a rede é limitada, segura e se possui alguma situação de travamento (deadlock). Essas simulações podem ser realizadas utilizando-se o *software* PIPE2. Após isso, serão feitas simulações utilizando o *software* HPSim. Este *software*, diferentemente do PIPE2, pode exportar os resultados de uma simulação para um arquivo de texto separado por vírgulas (CSV), facilitando o processamento e análise dos mesmos.

#### **4.4. Detalhamento do Projeto**

O modelo é verificado por meio de simulação nos *softwares* PIPE2 e HPSim. Serão gerados também os grafos de alcançabilidade para verificar se todos os estados podem ser alcançados e se há alguma situação que trava o sistema.

Tendo verificado esse modelo, elementos adicionais podem ser progressivamente incorporados no mesmo a fim de adequá-lo às variações dos casos reais. Serão incluídos semáforos para pedestres e novas **fases**. A conexão entre diversos cruzamentos (cada qual representado por um módulo) utilizando a STPN com configuração *master-slave* também deve ser estudada considerando etapas de validação a cada iteração. Finalmente, a conversão de um modelo em rede de Petri para linguagem LD de controladores lógicos programáveis também será apresentada.

## **5. Análise de risco**

Este trabalho possui certos aspectos que podem não se comportar da maneira desejada. Isto significa que é importante analisa-los na tentativa de prevê-los e preveni-los quando possível.

Um dos principais riscos do projeto é a elaboração de um modelo que não seja capaz de representar de maneira fiel um cruzamento real da cidade de São Paulo, ou então que esteja limitado a apenas poucos tipos de cruzamentos semaforicos existentes.

Outro risco significativo do projeto é a possibilidade de o modelo final não agregar valor ao sistema de sinalização de semáforos já presente em São Paulo, ou seja, não apresentar uma utilidade prática.

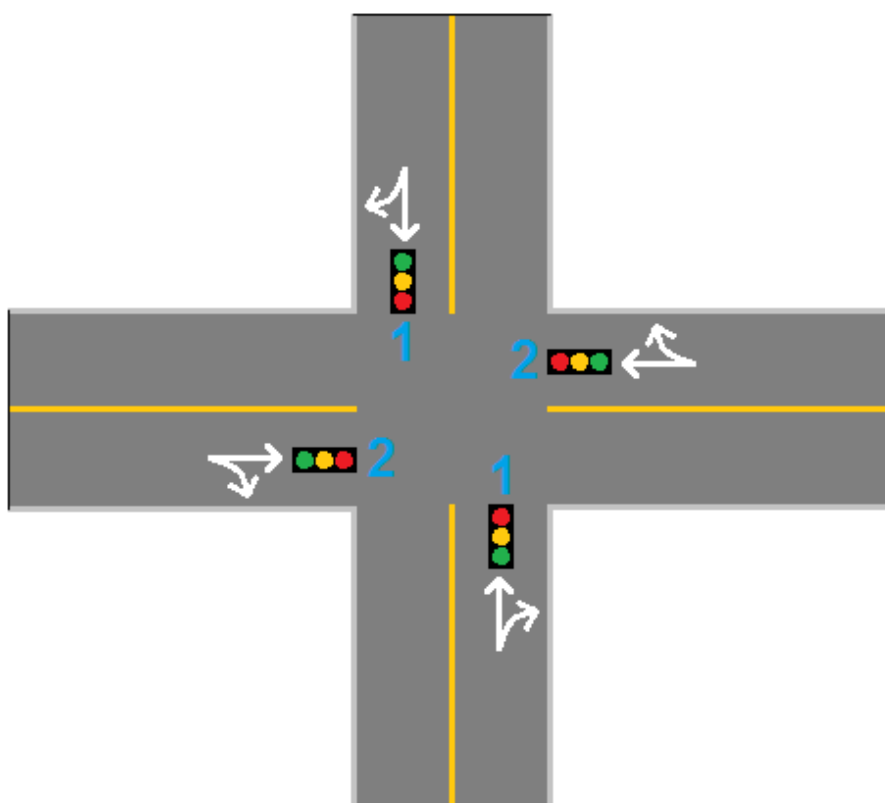
Caso a transcrição do modelo em rede de Petri para linguagem LD não seja feita de maneira meticulosa, e careça de uma revisão adequada, existe o risco do programa final não retratar o modelo inicial. Isto implicaria em um não cumprimento dos requisitos do projeto e também em uma falha grave.

## 6. Modelagem e Análise

### 6.1. Cruzamento 1

Neste projeto, é modelado inicialmente um cruzamento simples entre duas vias. Este cruzamento está ilustrado na Figura 10. Nela, as duas **fases** são representadas por números, onde números iguais representam os semáforos que estão no **estágio** verde em uma mesma **fase**.

O cruzamento a ser modelado tem duas **fases** e é permitido apenas seguir em frente e conversão à direita. Não são incluídos semáforos para pedestres ou ciclistas ou semáforos para conversão à esquerda com **fases** adicionais.



*Figura 10 – Cruzamento de duas **fases** que será modelado inicialmente*

O cruzamento é modelado nos *softwares* PIPE2 e HPSim, sendo que os resultados da simulação são extraídos do HPSim e o grafo de alcançabilidade e outros testes são feitos no PIPE2. Esse cruzamento é referenciado como Cruzamento 1, com a finalidade de organizar os modelos desenvolvidos para os próximos cruzamentos. O modelo para o Cruzamento 1 pode ser visualizado

na Figura 11. Nele, as letras N, S, L e O significam, respectivamente, Norte, Sul, Leste e Oeste e entre parênteses está a indicação da **fase** na qual o sinal verde do respectivo semáforo está aceso.

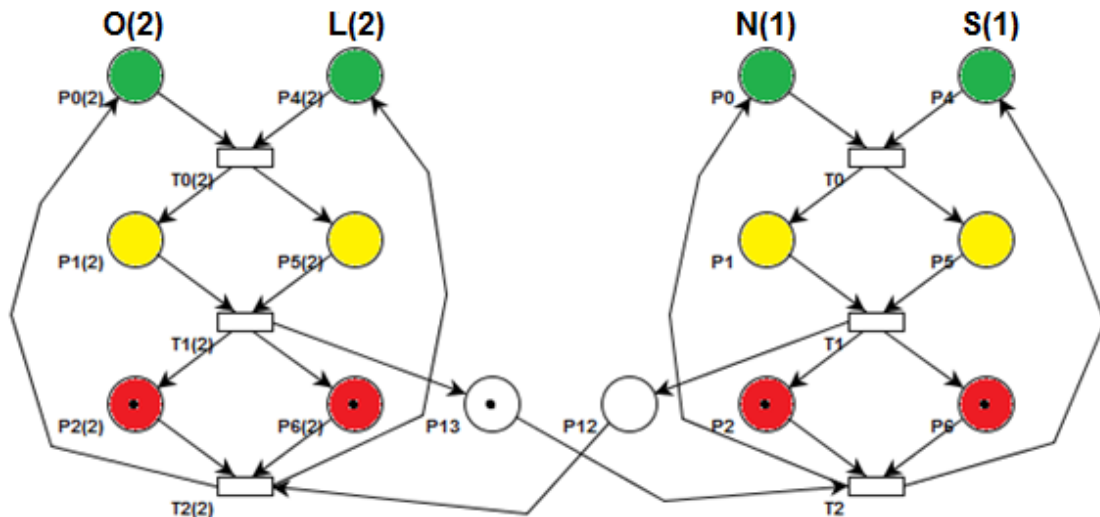


Figura 11 - Modelo do Cruzamento 1

Neste modelo, as cores representam os sinais/**estágios** a que os respectivos lugares correspondem. Inicialmente, todos os semáforos estão no **estágio** vermelho. Decorrido o tempo da transição T2, os semáforos N(1) e S(1) entram em **estágio** verde e, passado o tempo da transição T0, passam para o **estágio** amarelo. Após o tempo da transição T1, chega-se novamente ao estado onde todos os semáforos estão em **estágio** vermelho. Porém, dessa vez, o lugar P12 tem uma marca, habilitando a transição T2(2), que, após decorrido seu tempo, passará os semáforos O(2) e L(2) para o **estágio** verde, estabelecendo assim um ciclo.

É importante notar que, nesse modelo, para alterar o tempo de **estágio** verde de qualquer um dos semáforos, basta variar os tempos das transições T0 e T0(2). As transições T1 e T1(2) representam o tempo de **estágio** amarelo e as transições T2 e T2(2) representam o tempo em que todos os semáforos ficam em **estágio** vermelho.

Adotou-se, para chegar a esse modelo, uma abordagem sequencial, muito similar ao método adotado por Huang *et al.* (2014) em sua modelagem de um cruzamento de duas **fases**. Nesse tipo de abordagem, busca-se fazer

com que os estados da rede de Petri sigam uma ordem fixa e imutável, o que, nesse caso, é garantido pelos lugares P12 e P13. Os mesmos armazenam qual foi a última dupla de semáforos a chegar no **estágio** vermelho e, com isso, definem qual será a próxima dupla a passar para o **estágio** verde. Diferentemente de Huang *et al.* (2014), no entanto, aqui não utiliza-se a configuração *master-slave*, a fim de tornar o modelo mais simples.

O grafo de alcançabilidade é gerado automaticamente pelo PIPE2 e, através do mesmo, percebe-se que todos os estados são alcançáveis e não há nenhuma situação de travamento. Ou seja, não há nenhuma situação em que o modelo pode interromper o funcionamento. Este grafo pode ser visualizado na Figura 12. Nele, cada elipse representa um estado do modelo e cada seta representa a transição disparada. Foi feita também uma análise de espaço de estados. Esta mostrou que a rede é limitada, segura e não contém deadlock (travamento), como pode ser visto na Figura 13.

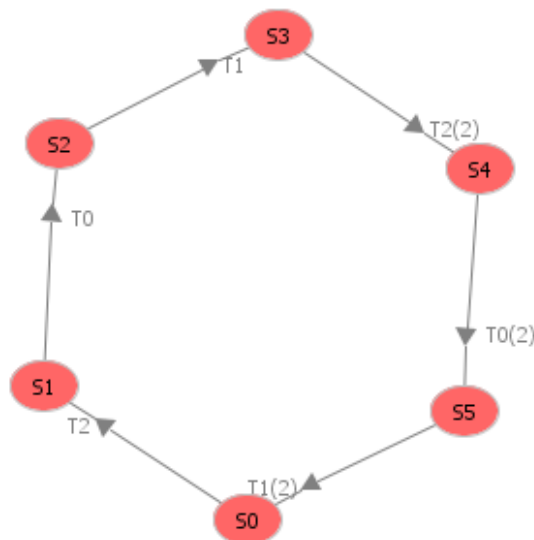


Figura 12 - Grafo de alcançabilidade do Cruzamento 1

Petri net state space analysis results	
Bounded	true
Safe	true
Deadlock	false

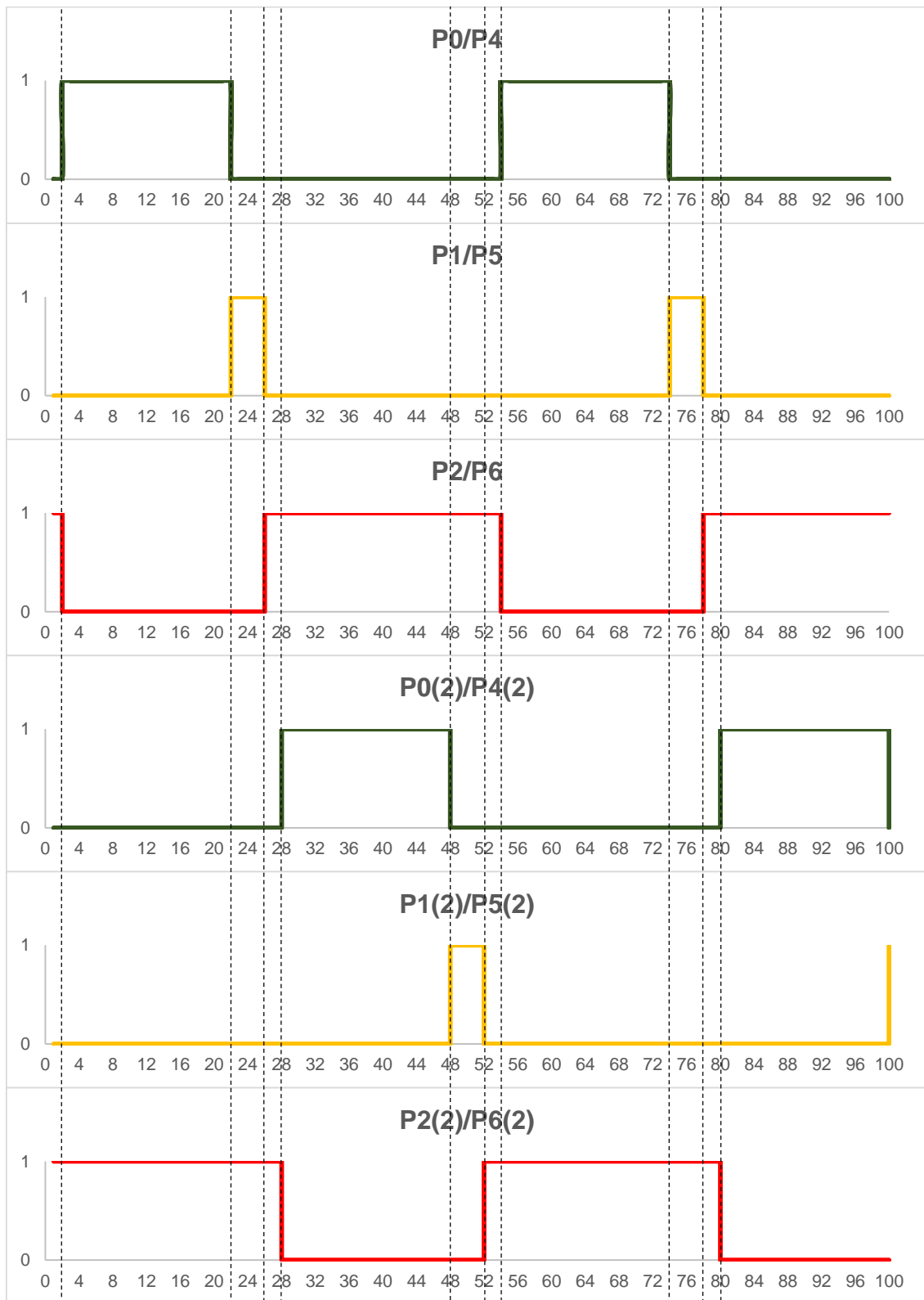
Figura 13 - Resultados da análise de espaço de estados do Cruzamento 1

A evolução do modelo foi simulada no *software* HPSim. Nele foi montado o mesmo modelo apresentado na Figura 11, e foram coletados dados para um período de simulação de 1000 segundos. Estes dados, adquiridos em um formato compatível com o Microsoft Excel, passaram por um processo de organização, e foram então transcritos para um formato gráfico.

O gráfico resultante é visto na Figura 14 e compreende um período de amostragem de 100 segundos de funcionamento, partindo de um estado onde ambos os semáforos se encontram fechados. Os tempos utilizados para cada **estágio** dos semáforos foram: 20 segundos aberto (verde), 4 segundos em transição (amarelo), 28 segundos fechado (vermelho), incluindo um período de 2 segundos em que ambos os semáforos do cruzamento permanecem simultaneamente fechados a cada ciclo de operação.

Na Figura 14, a escala vertical deve ser interpretada da seguinte maneira: o nível “1” representa que um determinado sinal encontra-se aceso, ou seja, visível para um motorista; o nível “0” indica que o sinal encontra-se apagado.

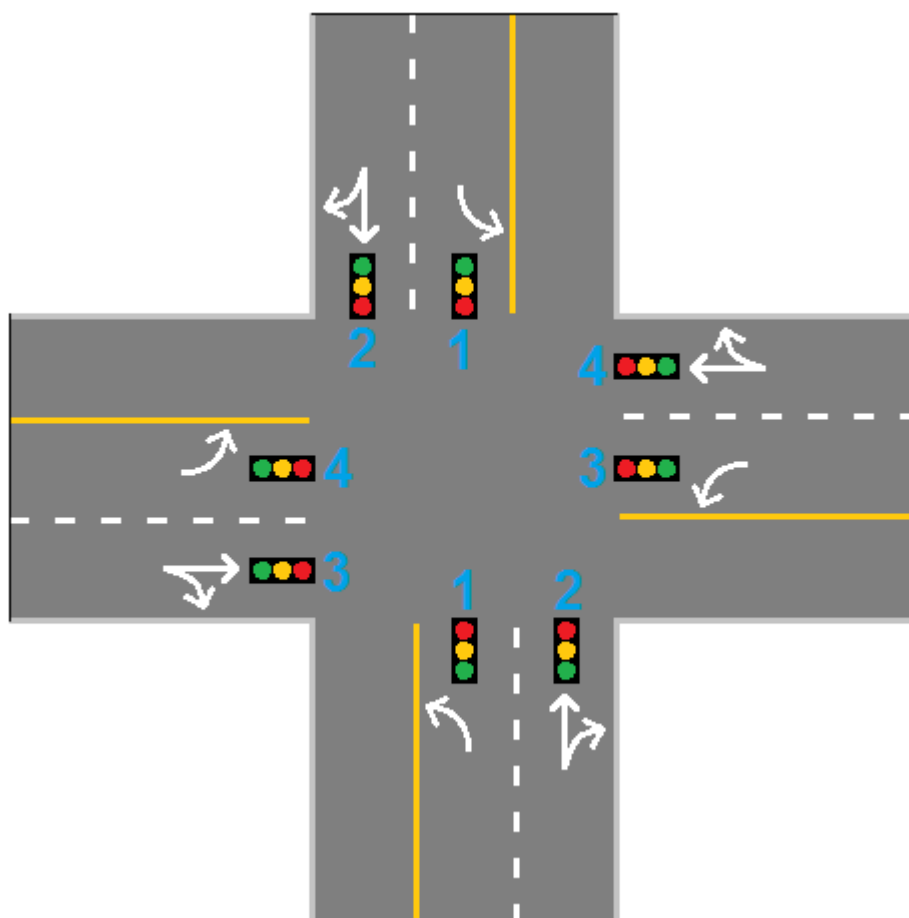
Por meio da análise do gráfico da Figura 14, observa-se o correto funcionamento do modelo proposto, onde sempre que um semáforo encontra-se em **estágio** verde, o outro encontra-se em vermelho. Também é possível visualizar a correta operação dos sistemas de transição entre **estágios** (verde-amarelo-vermelho) e de segurança (os períodos de 2 segundos em que ambos os semáforos encontram-se fechados). Os resultados obtidos são condizentes com os resultados obtidos por Huang *et al.* (2014).



*Figura 14 – Simulação do modelo do Cruzamento 1*

## 6.2. Cruzamento 2

O segundo cruzamento modelado é uma intersecção de quatro **fases**. Em relação ao Cruzamento 1, são acrescentadas duas **fases** referentes à conversão para a esquerda em ambas as vias. A Figura 15 ilustra o Cruzamento 2. Nela, as quatro **fases** são representadas por números, onde números iguais representam os semáforos que estão no **estágio** verde em uma mesma **fase**.



**Figura 15 - Cruzamento 2 (4 fases)**

Assim como o primeiro, este cruzamento foi modelado no *software* PIPE2, sendo que o grafo de alcançabilidade e os demais testes foram feitos nele. O modelo para o Cruzamento 2 pode ser visualizado na Figura 16. Nele, as letras N, S, L e O significam, respectivamente, Norte, Sul, Leste e Oeste. Os números entre parênteses referem-se às **fases** do cruzamento de acordo com a Figura 15.

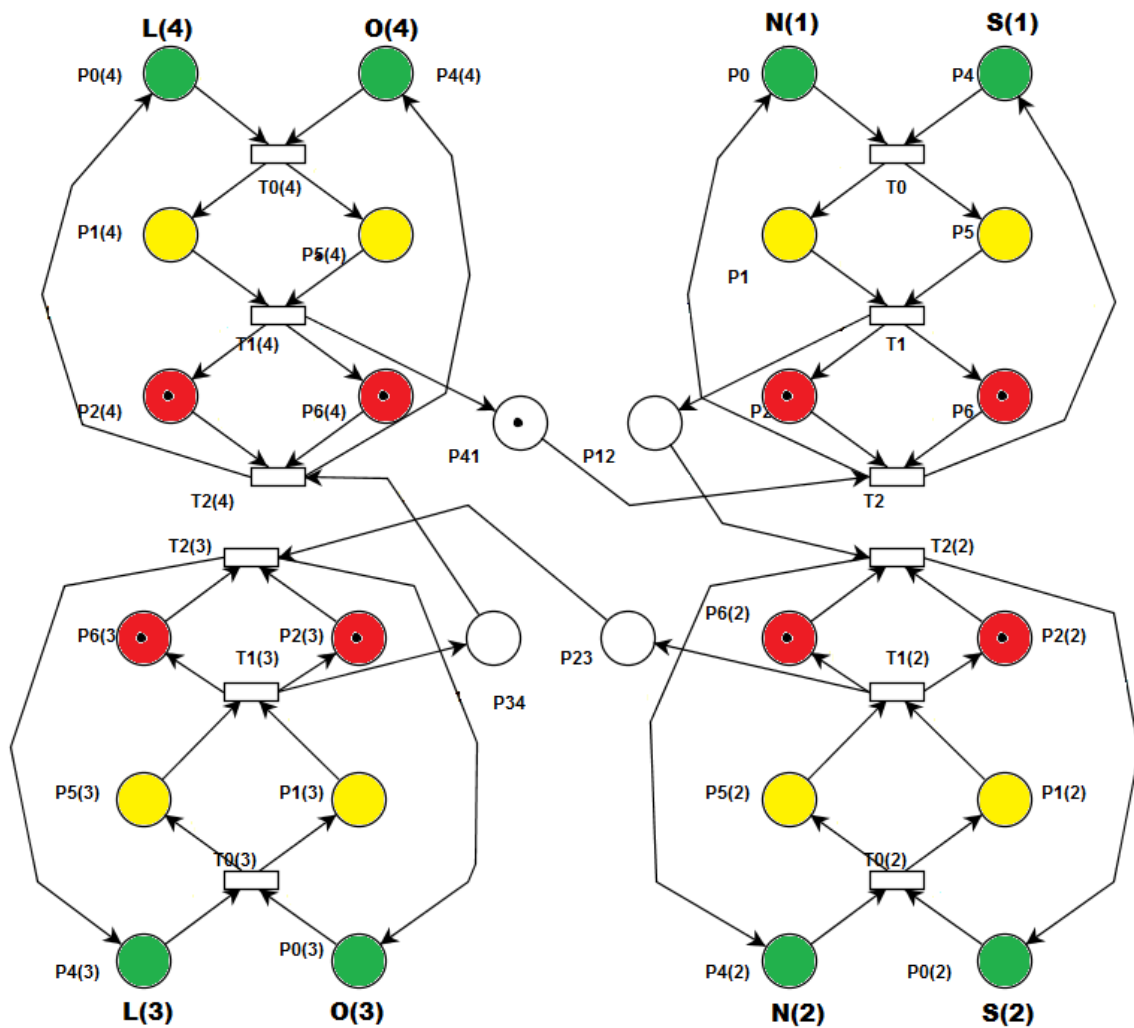
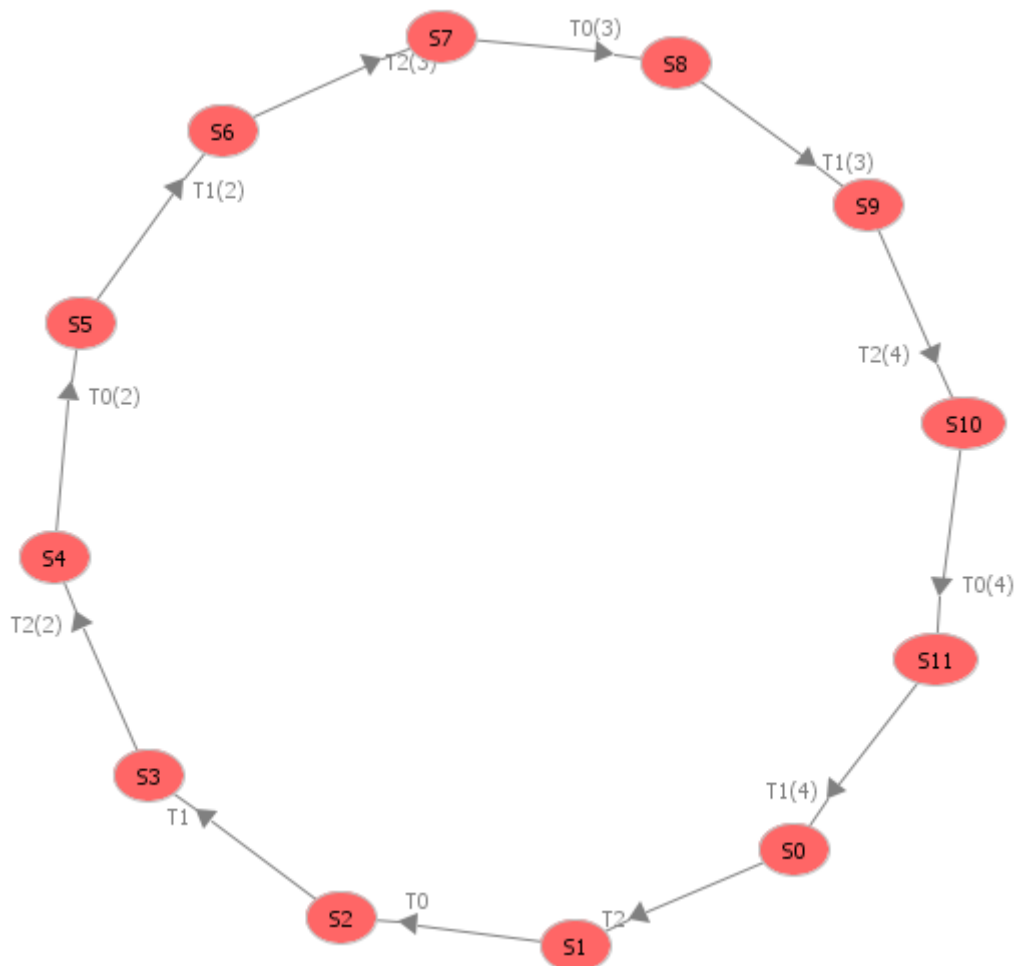


Figura 16 - Modelo do Cruzamento 2

Da mesma forma que no Cruzamento 1, neste modelo as cores representam os sinais a que os respectivos lugares correspondem. Inicialmente, todos os semáforos estão no **estágio** vermelho. Decorrido o tempo da transição T2, os semáforos N(1) e S(1) entram em **estágio** verde e, passado o tempo da transição T0, passam para o **estágio** amarelo. Após o tempo da transição T1, chega-se novamente ao estado onde todos os semáforos estão em **estágio** vermelho. Porém, dessa vez, o lugar P12 tem uma marca, habilitando a transição T2(2), que, após decorrido seu tempo, passará os semáforos N(2) e S(2) para o **estágio** verde. Este processo se repete por meio do acionamento das transições T2(3) e T2(4), quando então um ciclo completo é finalizado (o lugar P41 volta a ter uma marca), preparando o cruzamento para o início de um novo ciclo.

Neste modelo, a abordagem sequencial é mantida sem alterações. A implementação da mesma se dá exatamente da mesma forma que foi feita no Cruzamento 1. No Cruzamento 2, os lugares P12, P23, P34 e P41 definem o sequenciamento das duplas de semáforos a passarem para o **estágio** verde.

O grafo de alcançabilidade é gerado automaticamente pelo PIPE2 e, através do mesmo, percebe-se que todos os estados são alcançáveis e não há nenhuma situação de travamento. Este grafo pode ser visualizado na Figura 17. Nele, cada elipse representa um estado do modelo e cada seta representa uma transição. A análise de espaço de estados mostrou que a rede é limitada, segura e não contém deadlock (travamento), como pode ser visto na Figura 18.



*Figura 17 - Grafo de alcançabilidade do Cruzamento 2*

Petri net state space analysis results	
Bounded	true
Safe	true
Deadlock	false

Figura 18 - Resultados da análise de espaço de estados do Cruzamento 2

A simulação foi realizada no *software* HPSim. Nele foi montado o mesmo modelo apresentado na Figura 16, e foram coletados dados para um período de simulação de 1000 segundos. Estes dados, adquiridos em um formato compatível com o Microsoft Excel, passaram por um processo de organização, e foram então transcritos para um formato gráfico. Todo este processo foi análogo ao utilizado no Cruzamento 1.

O gráfico resultante é visto na Figura 19, e compreende um período de amostragem de 170 segundos de funcionamento (aproximadamente dois ciclos de funcionamento), partindo de um estado onde os quatro semáforos se encontram fechados. Os tempos utilizados para cada **estágio** dos semáforos foram: 15 segundos aberto (verde), 4 segundos em transição (amarelo), 21 segundos fechado (vermelho), incluindo um período de 2 segundos em que todos os semáforos do cruzamento permanecem simultaneamente fechados a cada ciclo de operação.

Na Figura 19, assim como na Figura 14, a escala vertical deve ser interpretada da seguinte maneira: o nível “1” representa que um determinado **estágio** encontra-se aceso, ou seja, visível para um motorista; o nível “0” indica que o **estágio** encontra-se apagado.

Por meio da análise do gráfico da Figura 19, observa-se o correto funcionamento do modelo proposto, onde sempre que um semáforo encontra-se em **estágio** verde, os outros encontram-se em vermelho. Também é possível visualizar a correta operação dos sistemas de transição entre **estágios** (verde-amarelo-vermelho) e de segurança (os períodos de 2 segundos em que todos os semáforos encontram-se fechados).

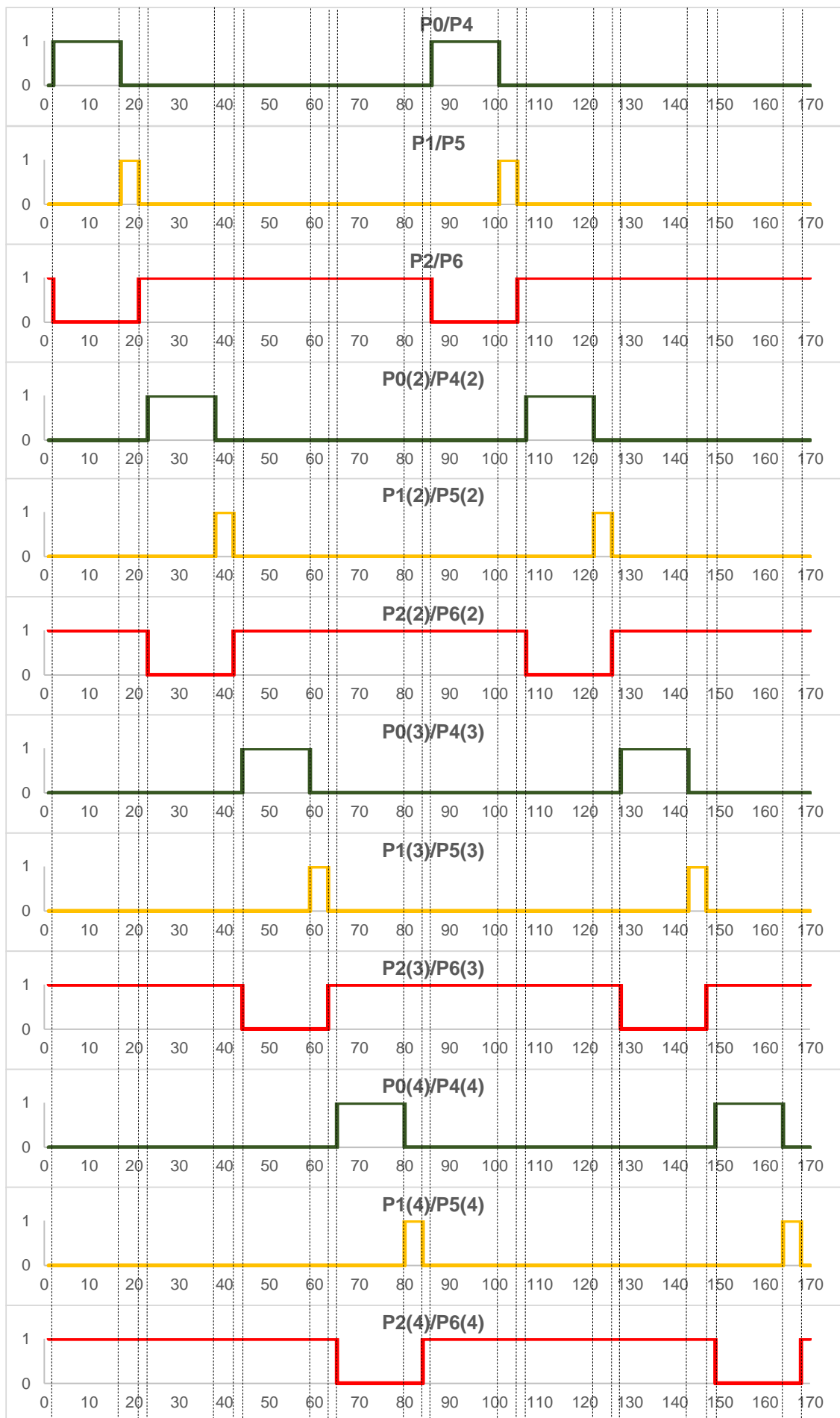


Figura 19 – Simulação do modelo do Cruzamento 2

### 6.3. Cruzamento 3

O terceiro cruzamento modelado é uma intersecção de oito **fases**. Este tipo de cruzamento é incomum, limitando-se a algumas intersecções entre grandes avenidas. Uma característica peculiar desse tipo de cruzamento é que cada semáforo fica em **estágio** verde em duas **fases** distintas. Não é possível agrupar os semáforos de **fases** semelhantes e isso torna o modelo mais complexo. A Figura 20 ilustra o Cruzamento 3. Assim como nos cruzamentos anteriores, as oito **fases** são representadas por números, onde números iguais representam os semáforos que estão no **estágio** verde em uma mesma **fase**.

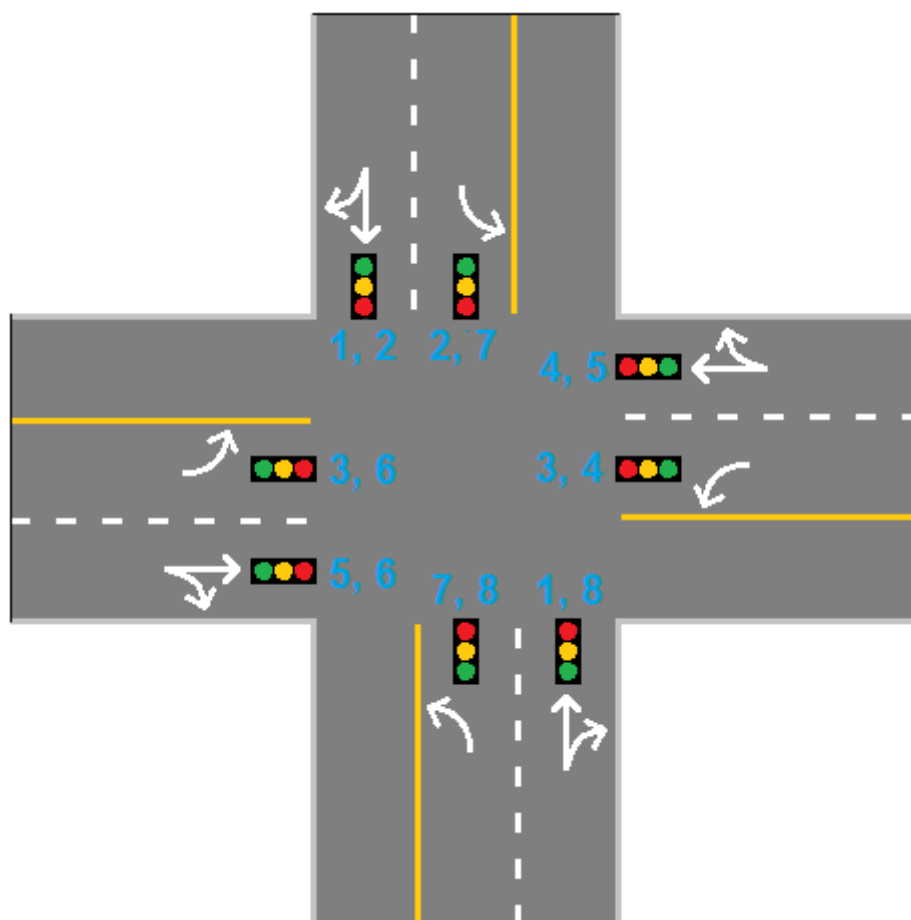


Figura 20 - Cruzamento 3 (8 **fases**)

O cruzamento é modelado no PIPE2 para obter o grafo de alcançabilidade e a análise de espaço de estados e no HPSim para gerar os resultados em um arquivo do tipo CSV, que pode ser analisado no Microsoft Excel. A Figura 21 mostra o modelo do Cruzamento 3. Nele é adotada a

mesma convenção dos cruzamentos anteriores, mas nesse caso cada semáforo é associado com duas **fases** ao invés de apenas uma.

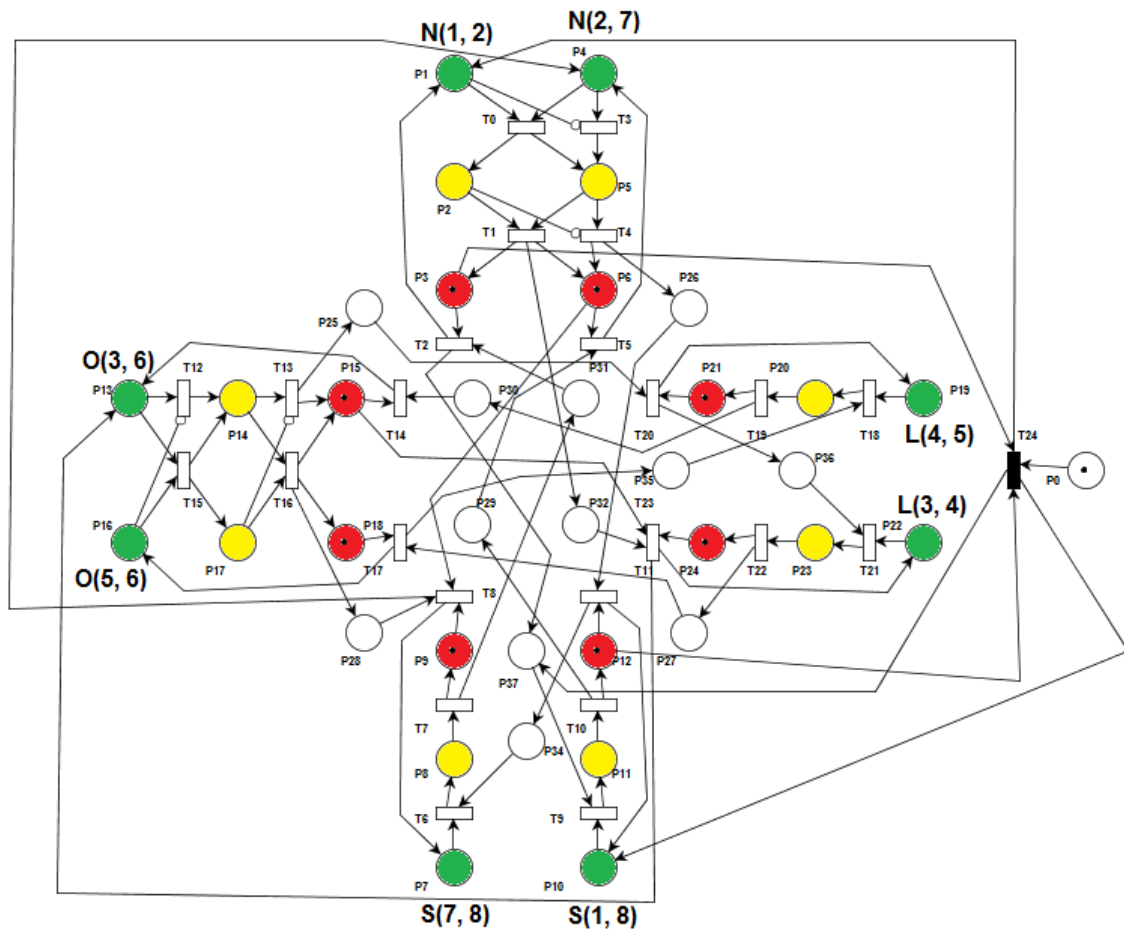


Figura 21 - Modelo do Cruzamento 3

Como nos demais cruzamentos, inicialmente, todos os semáforos estão no **estágio** vermelho. Nesse caso, para que isso seja possível, é necessário adicionar um lugar (P0) com marcação inicial e uma transição imediata (T24) que ative diretamente o **estágio** verde dos semáforos N(1, 2) e S(1, 8). Decorrido o tempo da transição T9, apenas o semáforo S(1, 8) entra em **estágio** amarelo e, posteriormente, em vermelho. Após isso, o semáforo N(2, 7) entra em **estágio** verde e, decorrido o tempo da transição T0, os semáforos N(1, 2) e N(2, 7) passam simultaneamente para o **estágio** amarelo e, em seguida, vermelho.

Em seguida, os semáforos da **fase** 3 passam simultaneamente para o **estágio** verde. Decorrido o tempo da transição T12, apenas o semáforo O(3, 6)

passa para o **estágio** amarelo e, eventualmente, vermelho. Assim, o semáforo L(4, 5) entra em **estágio** verde e inicia-se a **fase** 4. O processo continua dessa forma até retornar à **fase** 1 e o ciclo recomeçar. O lugar P0 nunca é atingido novamente até que haja uma interferência que reinicie o sistema do cruzamento.

Diferentemente dos cruzamentos apresentados anteriormente, a modelagem do cruzamento de 8 **fases** exige adaptações no método sequencial. A primeira das modificações é a adição do lugar P0 e da transição imediata T24. Estes forçam o sistema a se iniciar sempre na **fase** 1 após reiniciado. Nesse cruzamento, os semáforos opostos não são sincronizados e, portanto, não disparam sempre ao mesmo tempo, não podendo compartilhar as mesmas transições. Sendo assim, cada semáforo deve ter suas próprias transições.

Em todas as **fases**, dois semáforos devem estar em **estágio** verde ao mesmo tempo. Não necessariamente isso ocorre com ambos os semáforos chegando e saindo do **estágio** verde ao mesmo tempo (situação modelada na Figura 22a). Há três outras possibilidades, que ocorrem nesse cruzamento: ambos chegando ao **estágio** verde ao mesmo tempo, mas saindo em tempos diferentes (Figura 22b); ambos chegando e saindo do **estágio** verde em tempos diferentes (Figura 22c); e ambos chegando ao **estágio** verde em tempos diferentes, mas saindo do mesmo simultaneamente (Figura 22d). As maneiras de modelar as quatro situações estão ilustradas na Figura 22.

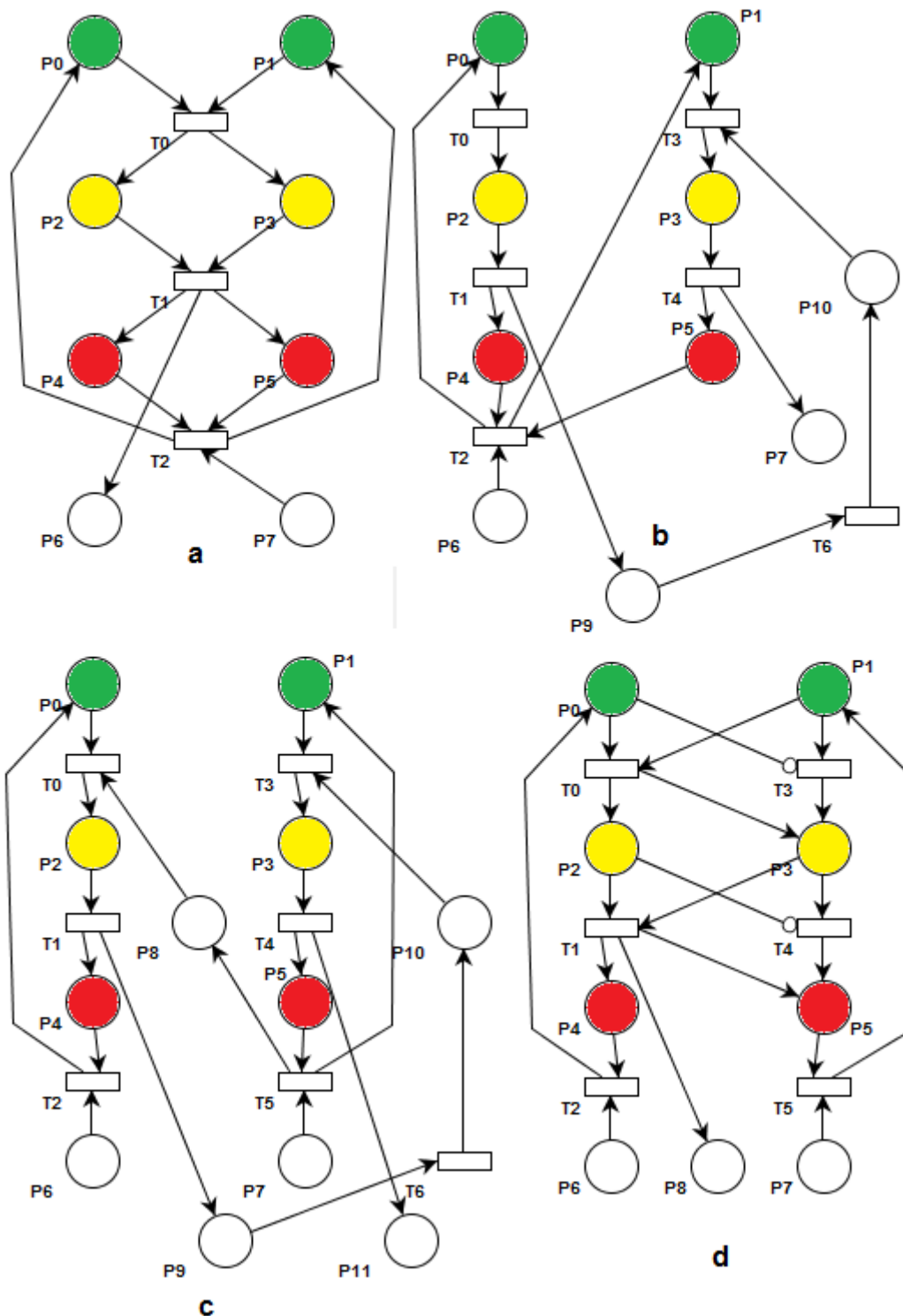


Figura 22 - Modelagem das quatro situações possíveis para dois semáforos com uma **fase** em comum

Na Figura 22a, é modelada a situação em que os dois semáforos entram e saem do **estágio** verde ao mesmo tempo. Uma marca no lugar P7 dispara a transição T2 e ambos os semáforos passam para o **estágio** verde

simultaneamente. As transições T0 e T1 garantem que todas as mudanças de **estágio** ocorram simultaneamente, sendo que a transição T1 habilita o início de uma nova **fase** em outros semáforos, modelada pelo lugar P6.

A Figura 22b mostra a modelagem da situação em que os dois semáforos entram no **estágio** verde simultaneamente, mas saem em tempos diferentes. A transição T2 passa ambos para o **estágio** verde, mas a transição T3 só é habilitada após o início da **fase** seguinte, que ocorre quando a transição T6 é disparada após o disparo de T1. Essa situação ocorre, por exemplo, com os semáforos O(3, 6) e L(3, 4) do Cruzamento 3.

Na Figura 22c está modelada a situação em que ambos os semáforos chegam e saem do **estágio** verde em tempos diferentes. Ao chegar um token em P6, a transição T2 é disparada, mas T0 não está habilitada até que haja marca no lugar P8. Ao chegar um token em P7, a transição T5 é disparada, mas T3 não está habilitada até que haja marca em P10. T0 dispara, seguida por T1, que habilita T6, que inicia a **fase** seguinte e, finalmente, habilita T3 para o fechamento do segundo semáforo. Essa situação ocorre, por exemplo, com os semáforos L(3, 4) e L(4, 5) do Cruzamento 3.

Finalmente, na Figura 22d, está modelada a situação em que os semáforos chegam em tempos diferentes ao **estágio** verde, mas saem juntos. Com a chegada de um token no lugar P6, a transição T2 é disparada. O disparo de T0 só é possível, porém, quando há uma marca em P1, o que ocorre após o disparo de T5 ao chegar um token em P7. Após isso, a transição T3 está desabilitada pela presença de marca em P0, fazendo com que apenas T0 possa ser disparada, fazendo os dois semáforos saírem simultaneamente do **estágio** verde. Essa situação ocorre, por exemplo, nos semáforos N(1, 2) e N(2, 7) do Cruzamento 3.

O grafo de alcançabilidade é gerado automaticamente pelo PIPE2 e, através do mesmo, percebe-se que todos os estados são alcançáveis e não há nenhuma situação de travamento. O estado S0 só é atingido no início do funcionamento, após o que é atingido um ciclo infinito. Este grafo pode ser visualizado na Figura 23. A análise de espaço de estados mostra, assim como nos cruzamentos anteriores, que a rede é limitada, segura e não contém deadlock, como pode ser visto na Figura 24.

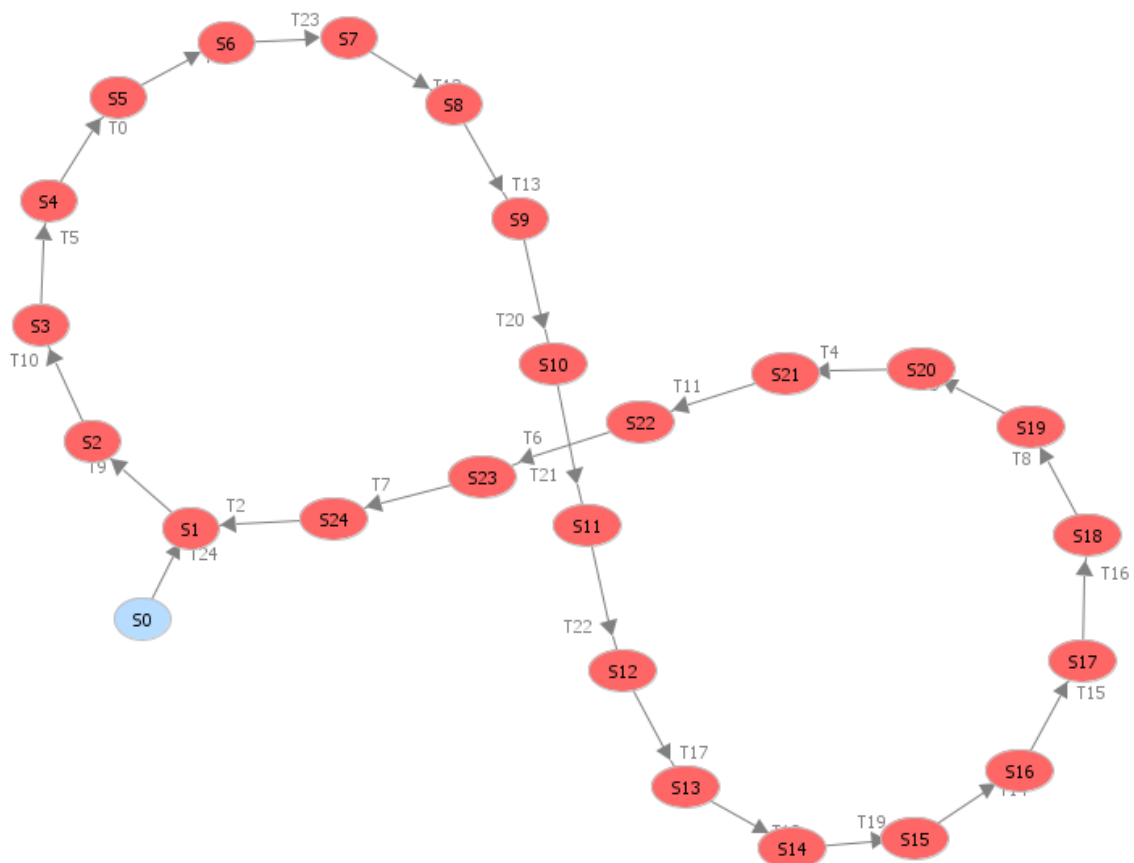


Figura 23 - Grafo de alcançabilidade do Cruzamento 3

Petri net state space analysis results	
Bounded	true
Safe	true
Deadlock	false

Figura 24 - Análise de espaço de estados do Cruzamento 3

A simulação foi realizada no *software* HPSim. Nele foi montado o mesmo modelo apresentado na Figura 21, e foram coletados dados para um período de simulação de 1000 segundos. Estes dados, adquiridos em um formato compatível com o Microsoft Excel, passaram por um processo de organização, e foram então transcritos para um formato gráfico. Todo este processo foi análogo ao utilizado nos Cruzamentos 1 e 2.

O gráfico resultante é visto nas Figuras 25 e 26, e compreende um período de amostragem de 207 segundos (um ciclo completo de

funcionamento), partindo de um estado onde os semáforos **N(1,2)** e **S(1,8)** se encontram abertos, enquanto os seis restantes estão fechados. Os tempos utilizados para cada **estágio** dos semáforos foram: 20 segundos aberto (verde) e 5 segundos em transição (amarelo). Os tempos do **estágio** vermelho variam para cada semáforo, visto que a configuração de abertura e fechamento dos mesmos não é uniforme. Por segurança, foi incluído um período de 2 segundos em que todos os semáforos do cruzamento, com exceção dos que devem se manter em **estágio** verde por **fases** consecutivas, permanecem simultaneamente fechados.

Nas Figuras 25 e 26, a escala vertical deve ser interpretada da seguinte maneira: o nível “1” representa que um determinado **estágio** encontra-se aceso, ou seja, visível para um motorista; o nível “0” indica que o **estágio** encontra-se apagado.

Por meio da análise dos gráficos apresentados nas Figuras 25 e 26, observa-se o correto funcionamento do modelo proposto. Enquanto quaisquer duas **fases** encontram-se em **estágio** verde, todas as outras se mantêm em **estágio** vermelho. Verifica-se que os **estágios** verde dos semáforos N(1,2), L(3,4), O(5,6), S(7,8) e S(1,8) possuem período dobrado, como é proposto no modelo. Também é possível visualizar a correta operação dos sistemas de transição entre **estágios** (verde-amarelo-vermelho) e de segurança (os períodos de 2 segundos em que todos os semáforos, com exceção daqueles que devem permanecer abertos em **fases** subsequentes, encontram-se fechados).

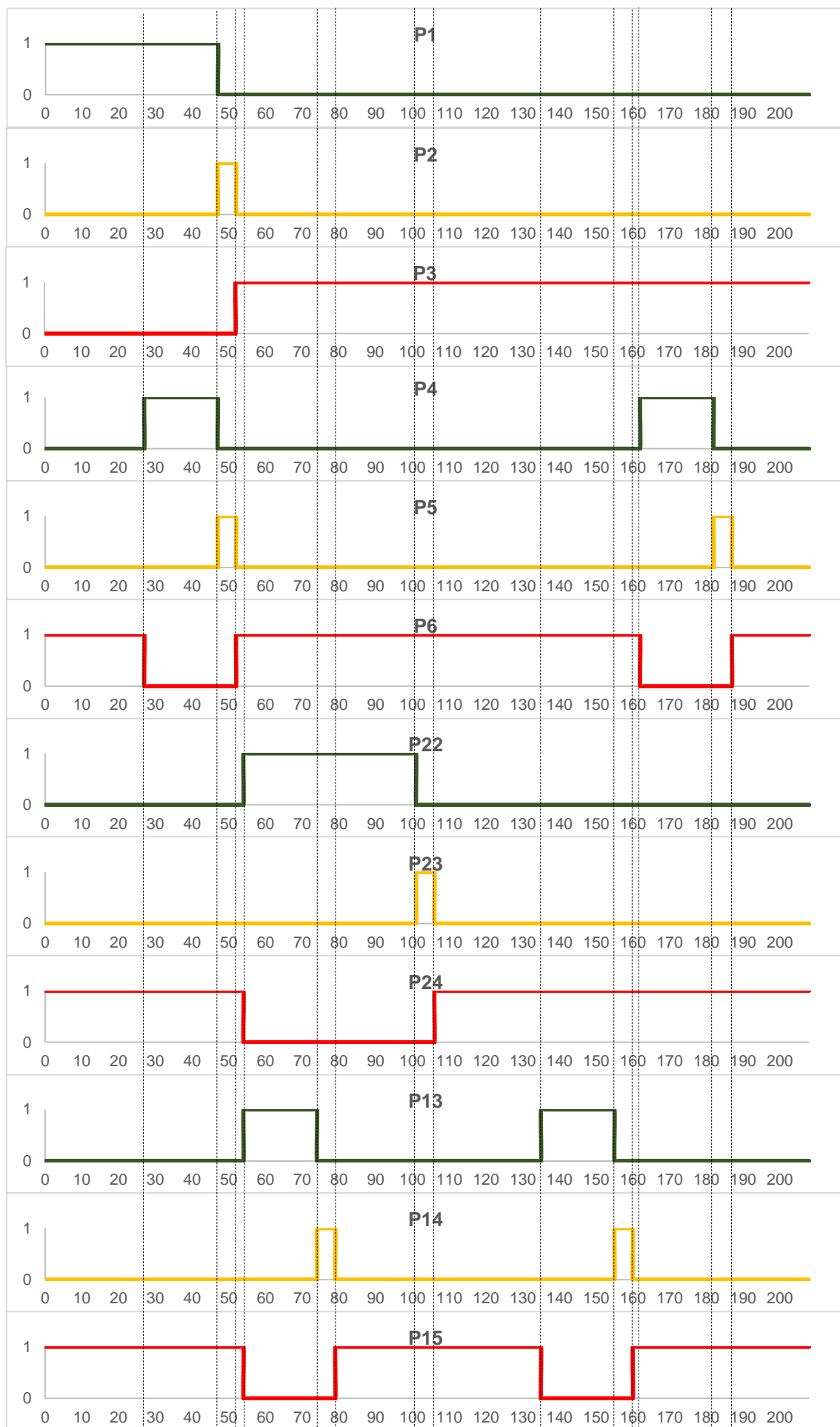


Figura 25 – Simulação do modelo do Cruzamento 3 - Parte 1

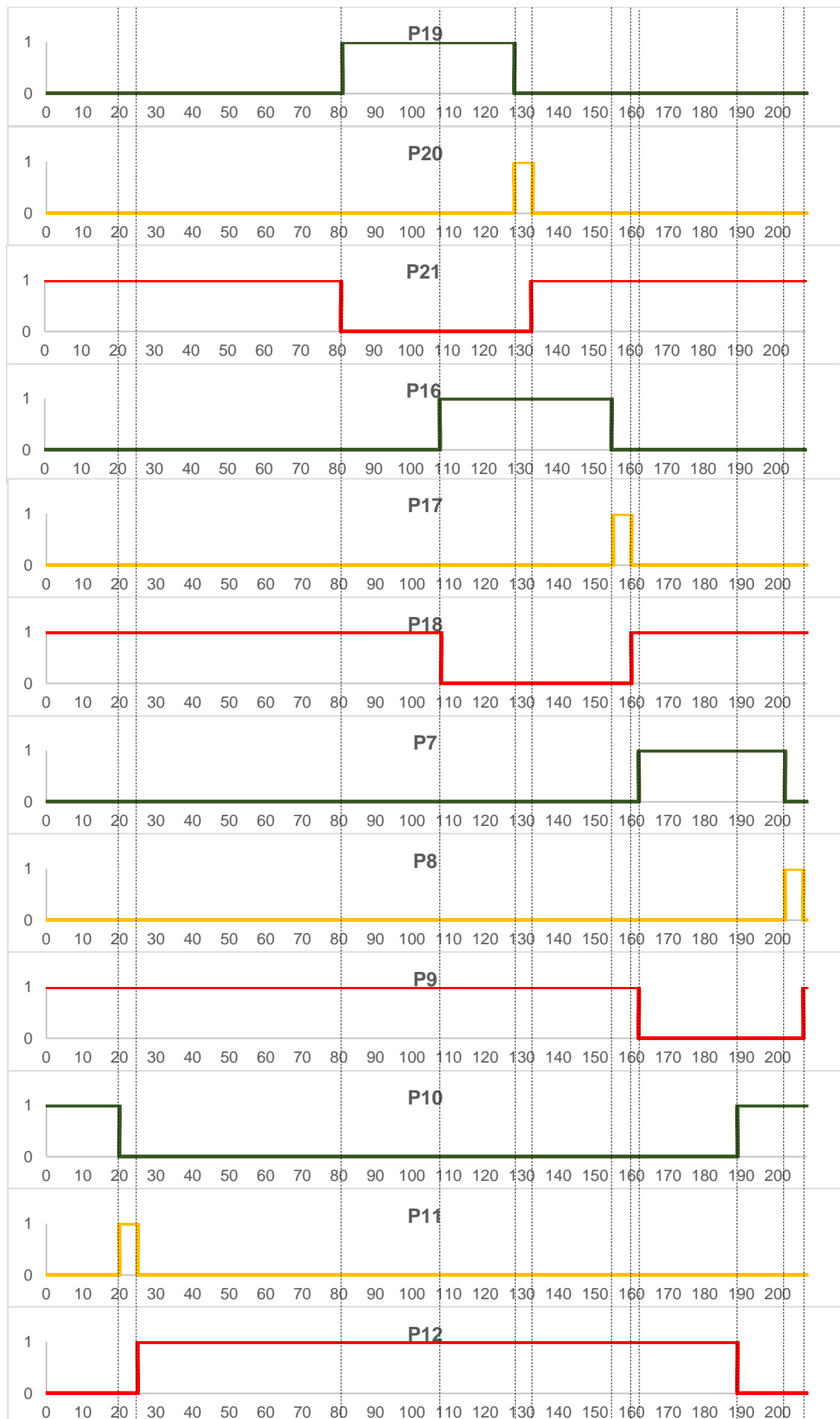


Figura 26 – Simulação do modelo do Cruzamento 3 - Parte 2

#### 6.4. Cruzamento 4 (com sensores)

Foram apresentados modelos de cruzamentos, sem considerar, semáforos para pedestres e outras situações reais. Com o intuito de possibilitar a criação de modelos mais próximos da realidade, considera-se aqui situações que ocorrem em alguns casos na prática.

O Cruzamento 4 consiste em um cruzamento de duas **fases** com sensores para fazer a detecção de veículos em espera e definir os tempos de abertura e fechamento dos semáforos. Este cruzamento é semelhante ao Cruzamento 1, mas com tempos variáveis determinados por sensores.

A forma mais comum de realizar a detecção de veículos para semáforos é com a utilização de laço indutivo. Quando um veículo para ou passa sobre o laço, a indutância do laço é reduzida. Quando ocorre essa redução de indutância, um pulso é enviado ao controlador indicando a passagem de um veículo (McLean, 2006).

Na modelagem do Cruzamento 4, é necessário o uso de STPN. Isso se dá pela necessidade de disparar transições de acordo com o recebimento dos sinais enviados pelo detector/sensor, que é externo à rede. A lógica de STPN não está implementada no *software* PIPE2. Entretanto, no caso do Cruzamento 4, as transições sincronizadas podem ser substituídas por transições imediatas sem causar alterações no grafo de alcançabilidade e na análise de espaço dos estados, possibilitando assim o uso do PIPE2 para realizar tais análises. O modelo do Cruzamento 4 pode ser visualizado na Figura 27.

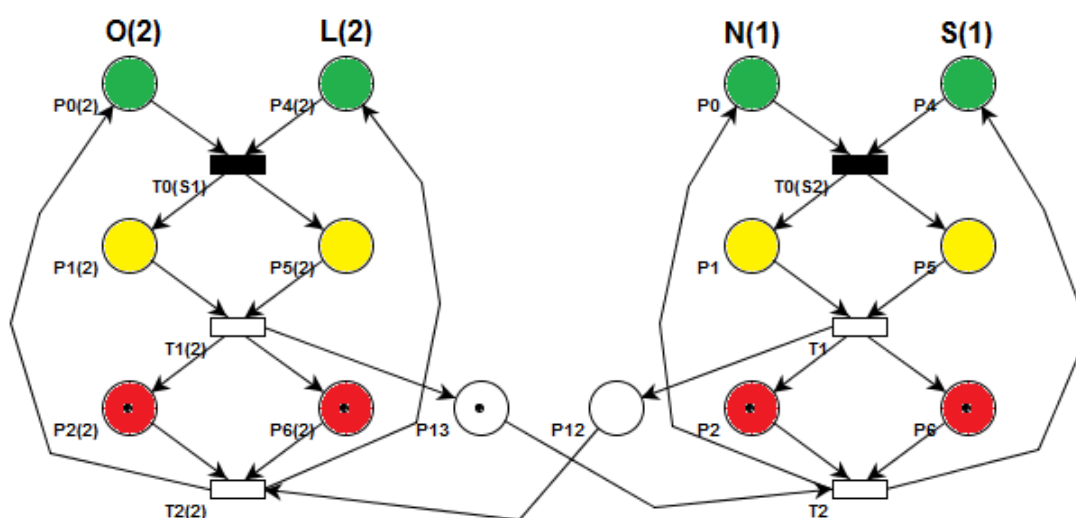
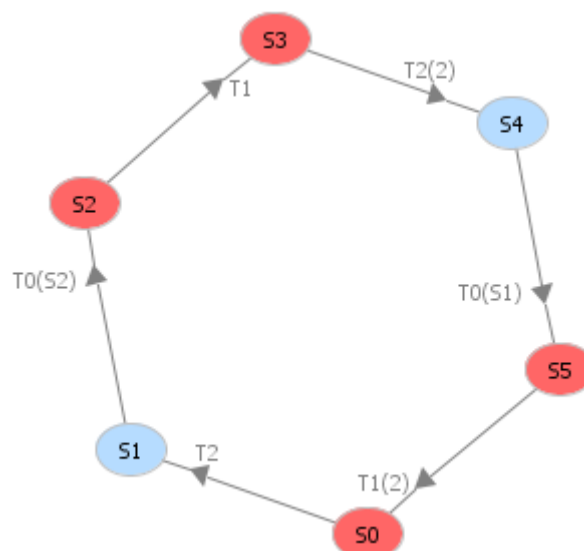


Figura 27 - Modelo do Cruzamento 4

Inicialmente, todos os semáforos estão no **estágio** vermelho. Nesse estado inicial, a transição T2 está habilitada, possibilitando que os semáforos da **fase** 1, N(1) e S(1), passem para o **estágio** verde. A transição T0(S2) é uma transição sincronizada que dispara quando é recebido do elemento externo (detector/sensor) o sinal S2, que indica que há certa quantidade de carros aguardando a abertura dos semáforos de **fase** 2, O(2) e L(2). A partir disso, os semáforos de **fase** 1 passam para o **estágio** amarelo e, posteriormente, vermelho, habilitando a transição T2(2). O procedimento com os semáforos de **fase** 2 ocorre de forma análoga ao que acontece com os semáforos de **fase** 1, estabelecendo assim um ciclo.

O grafo de alcançabilidade é gerado automaticamente pelo PIPE2 e, através do mesmo, percebe-se que todos os estados são alcançáveis e não há nenhuma situação de travamento. Este grafo pode ser visualizado na Figura 28. Nota-se que o grafo é similar ao grafo de alcançabilidade do Cruzamento 1, como o esperado. As elipses com fundo azul representam os estados que utilizam a transição sincronizada, ou seja, dependem de sinais externos para que ocorra a transição. A análise de espaço de estados mostrou, assim como nos cruzamentos anteriores, que a rede é limitada, segura e não contém deadlock, como pode ser visto na Figura 29.



*Figura 28 - Grafo de alcançabilidade do Cruzamento 4*

Petri net state space analysis results	
Bounded	true
Safe	true
Deadlock	false

Figura 29 – Análise de espaço de estados do Cruzamento 3

Para a verificação do funcionamento do modelo apresentado na Figura 27, realizou-se uma simulação no *software* HPSim. No entanto, para tal, é necessário simular também a chegada de carros ao cruzamento, de forma que o tempo de abertura dos semáforos varie de acordo com os sinais enviados pelos detectores/sensores.

Dado que o objetivo dessa simulação é verificar unicamente o funcionamento do modelo do Cruzamento 4, modelou-se o envio de sinais pelo controlador como uma distribuição exponencial com média de um sinal enviado a cada 20 segundos. Para tanto, assume-se que a chegada de carros ao cruzamento é um processo de Poisson, ou seja, os carros chegam continuamente e independentemente com uma taxa média de chegada constante. A distribuição exponencial é a escolha mais comum em teoria de filas para definir a distribuição dos tempos entre chegadas (Winston, 2004). Assim, a mesma será utilizada para modelar a chegada de veículos e, conseqüentemente, envio de sinais pelos detectores/sensores.

O *software* HPSim dispõe de um tipo de transição com tempo de disparo que segue uma distribuição exponencial. Assim, as transições T0(S1) e T0(S2) foram implementadas com esse tipo de transição para a simulação.

Nesta simulação, foram coletados dados para um período de 1000 segundos. Estes dados, adquiridos em um formato compatível com o Microsoft Excel, passaram por um processo de organização, e foram então transcritos para um formato gráfico. Todo este processo foi análogo ao utilizado nos Cruzamentos 1, 2 e 3.

O gráfico resultante é visto na Figura 30, e compreende um período de amostragem de 200 segundos (diversos ciclos de funcionamento), partindo de um estado onde todos os semáforos se encontram fechados. Conforme explicado acima, a chegada de veículos foi modelada por uma função

exponencial. Portanto, era esperado que a duração dos **estágios** verde e vermelho dos semáforos variasse a cada ciclo de operação. Na Figura 30 observa-se que isto realmente ocorre, visto que não há nenhum padrão distinguível de abertura e fechamento dos semáforos no período representado. Por segurança, foi incluído um período de 2 segundos em que todos os semáforos do cruzamento, com exceção dos que devem se manter em **estágio** verde por **fases** consecutivas, permanecem simultaneamente fechados.

Nas Figura 30, a escala vertical deve ser interpretada da seguinte maneira: o nível “1” representa que um determinado **estágio** encontra-se aceso, ou seja, visível para um motorista; o nível “0” indica que o **estágio** encontra-se apagado.

Por meio da análise do gráfico apresentado nas Figura 30, observa-se o correto funcionamento do modelo proposto. Enquanto uma **fase** encontra-se em **estágio** verde, a outra se mantém em **estágio** vermelho. Também é possível visualizar a correta operação dos sistemas de transição entre **estágios** (verde-amarelo-vermelho) e de segurança (os períodos de 2 segundos em que todos os semáforos encontram-se fechados).

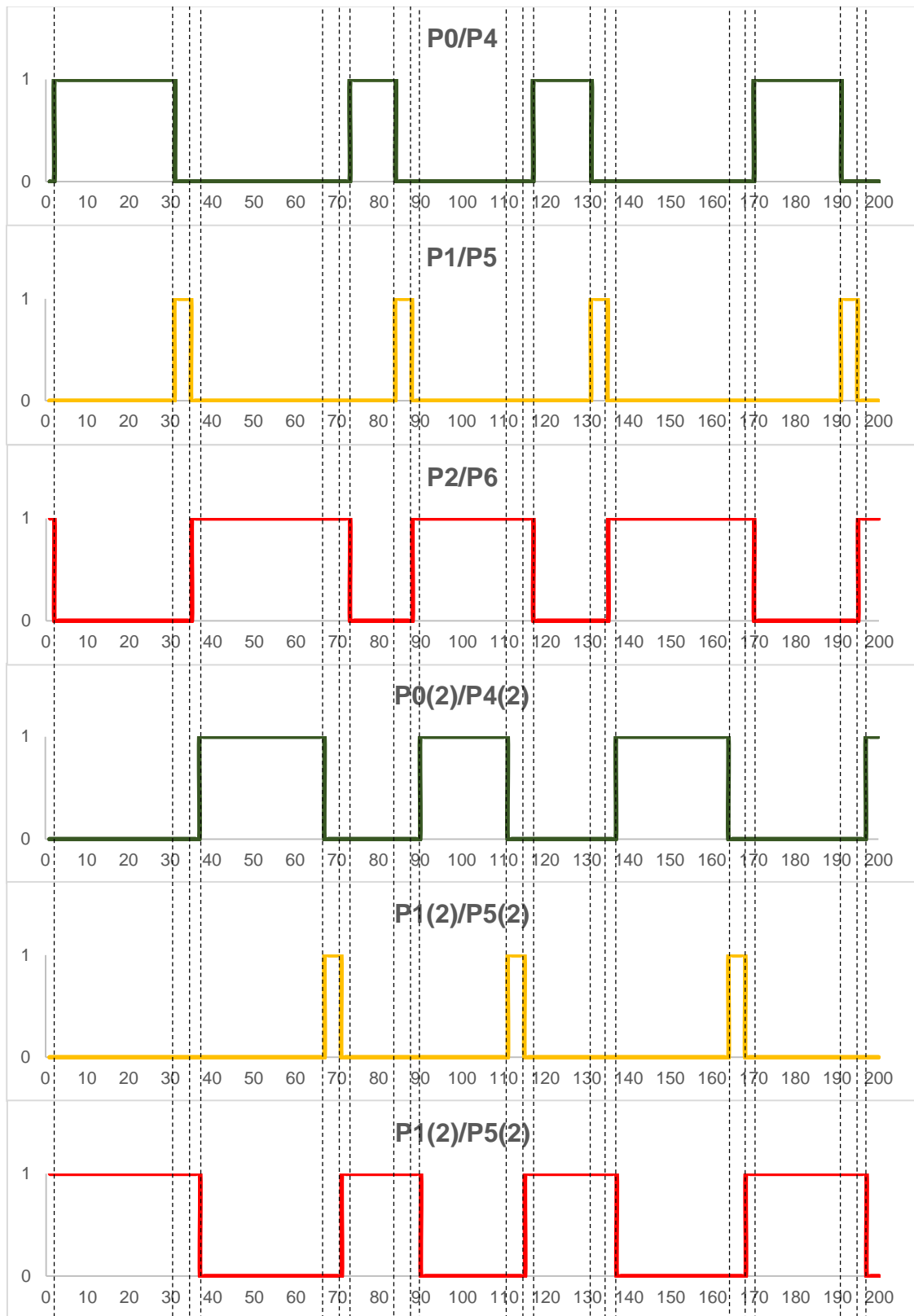


Figura 30 – Simulação do modelo do Cruzamento 4

### 6.5. Cruzamento 5 (com semáforos para pedestres)

O Cruzamento 5 é um cruzamento adaptado de duas **fases** e com semáforos para pedestres. Dessa forma, ele é similar ao Cruzamento 1 e pode ser visualizado na Figura 31. Nessa figura, os números próximos aos semáforos para pedestres mostram a **fase** em que os mesmos estarão em **estágio** verde.

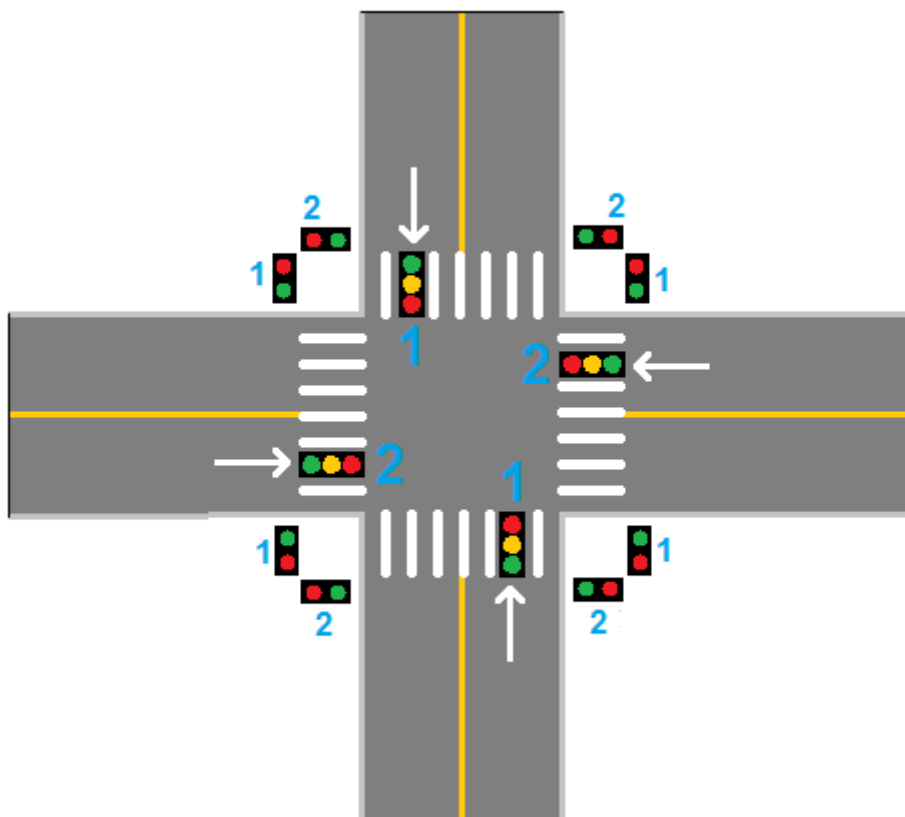
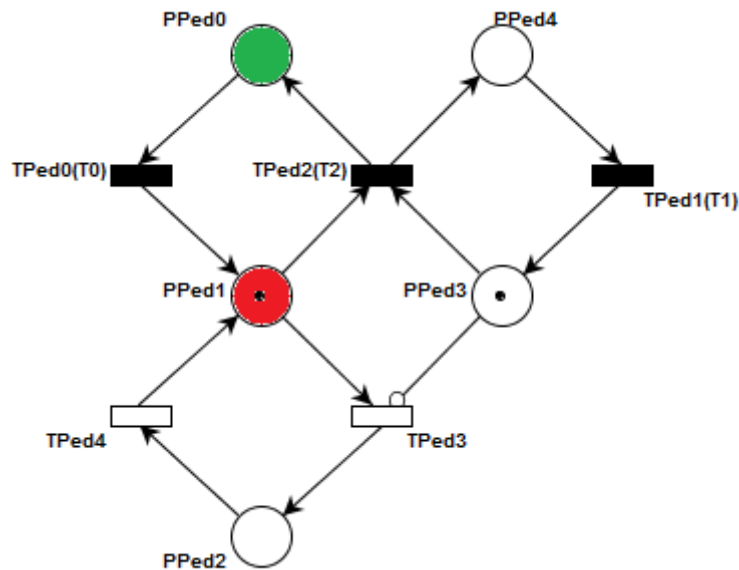


Figura 31 - Cruzamento 5

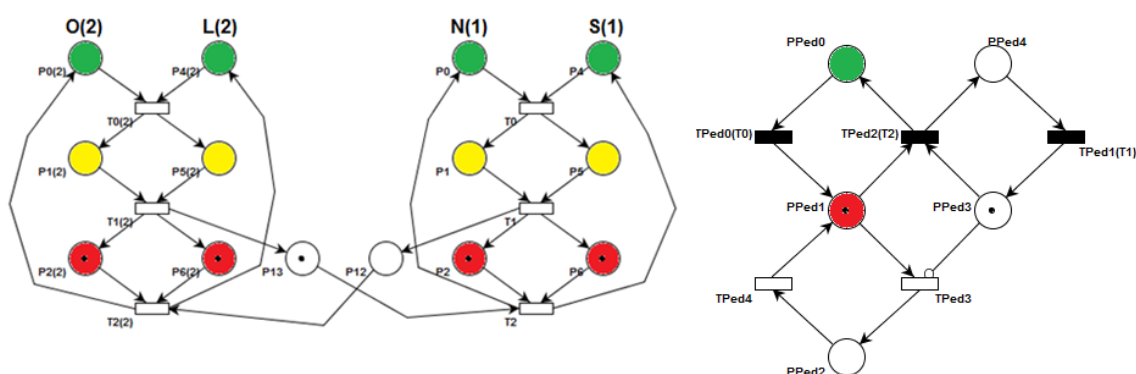
É importante notar que os semáforos para pedestres não possuem sinal amarelo. Ao invés disso, para indicar o fim da permissão de passagem de pedestres, a luz vermelha acende e apaga alternadamente algumas vezes antes de se acender em definitivo. Este comportamento também deve ser modelado.

A evolução do modelo de funcionamento dos semáforos de veículos controla a evolução do modelo dos semáforos de pedestres. Sendo assim, o modelo desenvolvido para o Cruzamento 1 pode ser utilizado sem alterações para o caso dos semáforos para veículos do Cruzamento 5. A Figura 32 mostra um exemplo de modelo em rede de Petri de um semáforo para pedestres e a Figura 33 mostra sua implementação no Cruzamento 5.



*Figura 32 - Modelo do semáforo para pedestres*

As transições imediatas nesse modelo de semáforo para pedestres representam transições sincronizadas com o modelo de funcionamento dos semáforos de veículos do cruzamento. No modelo da Figura 32, por exemplo, a transição TPed0(T0) será disparada, caso haja marca em PPed0, assim que a transição T0 do modelo associado aos veículos for disparada. No estado inicial, este semáforo está em **estágio** vermelho. Com o disparo de T2, TPed2(T2) será disparada e o semáforo entrará em **estágio** verde. Com o disparo de T0, será disparada a transição TPed0(T0) e a luz vermelha se acenderá. Enquanto a transição T1 não disparar, a marca alternará entre os lugares PPed1 e PPed2, fazendo com que a luz vermelha acenda e apague alternadamente. Com o disparo de T1, a marca que estava em PPed4 irá para PPed3 e inibirá o disparo de TPed3, fazendo com que a luz vermelha permaneça acesa. O processo descrito forma um ciclo que será repetido.



*Figura 33 - Exemplo de implementação do semáforo para pedestres no Cruzamento 5*

No modelo desenvolvido, o semáforo para pedestres modelado deve sempre estar sincronizado com um semáforo para veículos com direção concorrente à própria. No exemplo ilustrado na Figura 33, o semáforo para pedestres mostrado pode representar qualquer semáforo para pedestres na direção Leste-Oeste, já que está sincronizado com as transições T0, T1 e T2, que pertencem aos semáforos N(1) e S(1). Para modelar um semáforo para pedestres na direção Norte-Sul, basta sincronizá-lo com as transições T0(2), T1(2) e T2(2).

Para a simulação do modelo mostrado na Figura 33, não foi possível utilizar o *software* HPSim, já que este não suporta a modelagem com STPN, bem como o PIPE2. Por conta disso, a simulação foi realizada manualmente em uma planilha do *software* Microsoft Excel, seguindo o mesmo formato dos dados de saída gerados nas simulações feitas no HPSim. A Figura 34 mostra as cartas de tempo geradas com os dados de simulação para um período de amostragem de 100 segundos.

As cartas de tempo da Figura 34 mostram o funcionamento do modelo proposto, que está de acordo com o esperado. O semáforo para pedestres acompanha o sinal do semáforo para veículos nos **estágios** verde e vermelho. No caso em que o semáforo de veículos está em **estágio** amarelo, o semáforo de pedestres acende e apaga, alternadamente, a luz vermelha.

Os tempos de transição utilizados foram de 20 segundos para a passagem do estágio verde para amarelo, 5 segundos do amarelo para o vermelho, 28 segundos do vermelho para o verde e 1 segundo para o semáforo de pedestre acender ou apagar a luz vermelha.

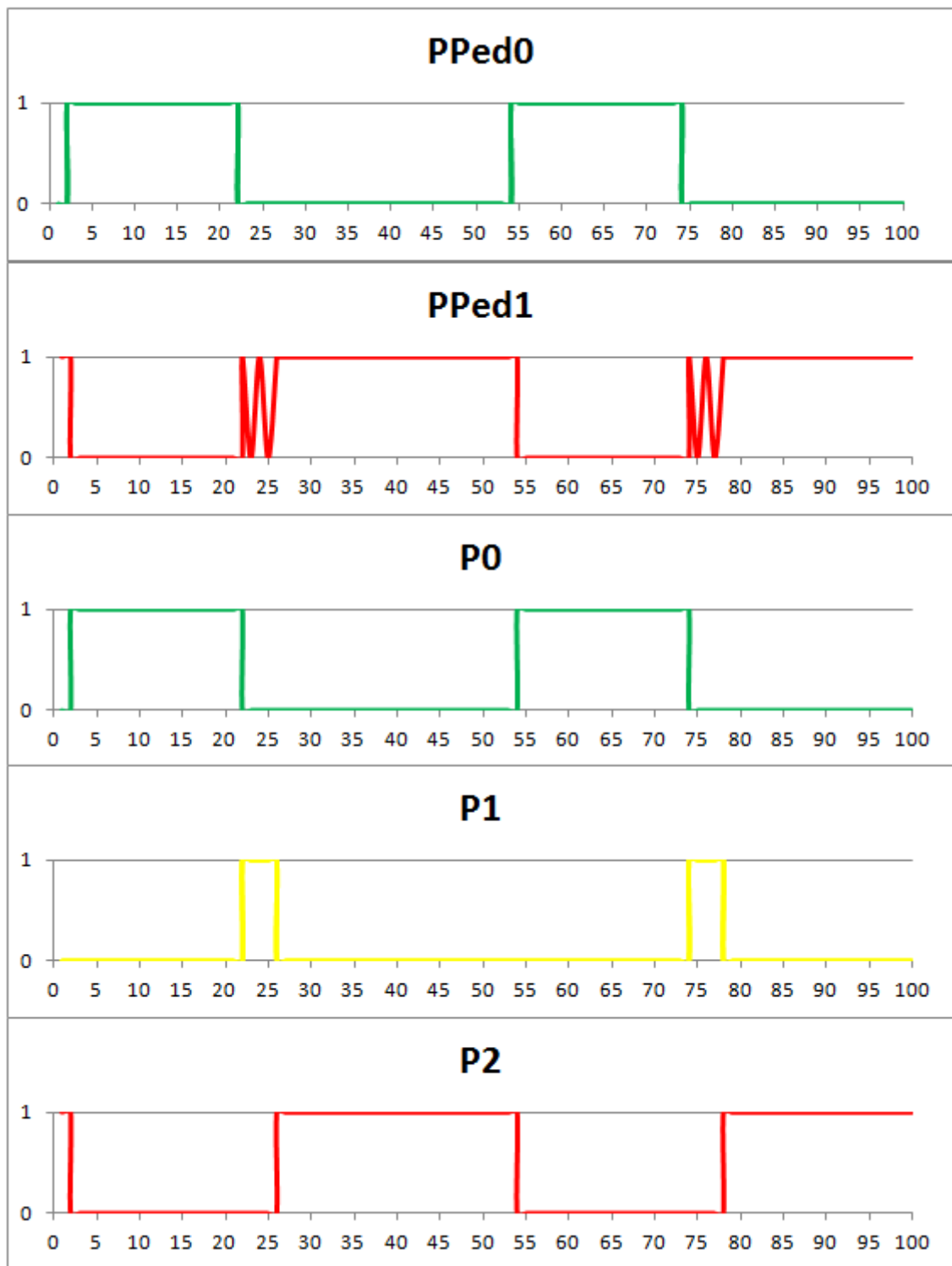


Figura 34 - Simulação do modelo do Cruzamento 5

## 6.6. Sincronização de um conjunto de cruzamentos

Como mostrado por Huang *et al.* (2014), uma das principais vantagens do uso de STPN para modelar cruzamentos semafóricos é a facilidade de sincronizar um conjunto de cruzamentos dependentes entre si.

Para implementar essa sincronização, utiliza-se um sistema *master-slave*, onde o cruzamento principal (*master*) envia sinais para os outros cruzamentos (*slaves*), de modo a comandá-los.

É apresentado a seguir um exemplo de utilização de STPN para interligar cruzamentos que seguem os modelos propostos nesse trabalho. O conjunto de cruzamentos a ser sincronizado consiste em um cruzamento central principal de quatro **fases** (*master*) e oito cruzamentos secundários de duas **fases** (*slaves*). A Figura 35 mostra a sequência de aberturas e fechamentos do conjunto de cruzamentos.

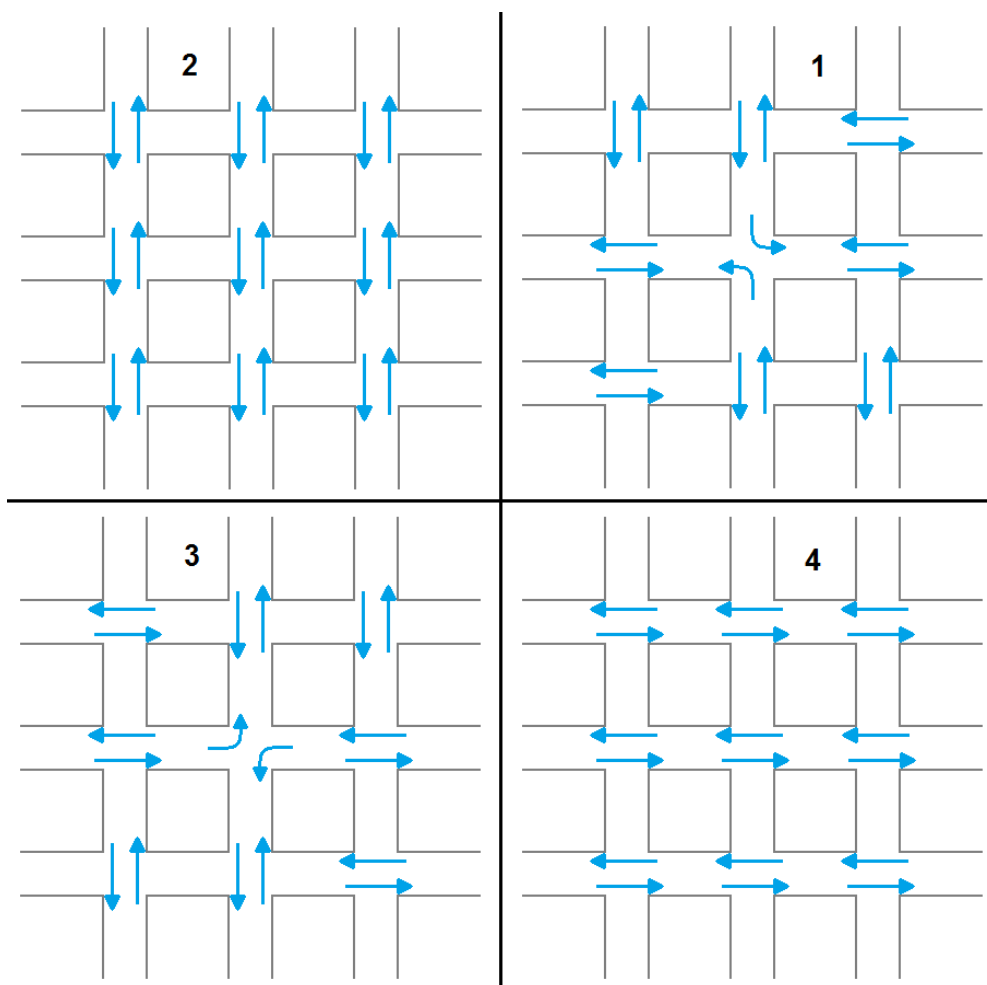


Figura 35 - Sequência de aberturas e fechamentos do exemplo de conjunto de cruzamentos

A sequência mostrada no exemplo da Figura 35 foi estabelecida com a finalidade de maximizar o fluxo de veículos pelas vias principais (que formam o cruzamento central). A intersecção principal foi implementada como um cruzamento de 4 **fases** semelhante ao Cruzamento 2. Sendo a *master* do sistema, o modelo de funcionamento em rede de Petri desse cruzamento envia sinais para ser sincronizada com os modelos *slave*. O módulo com o modelo desse cruzamento e os respectivos sinais de saída está mostrado na Figura 36.

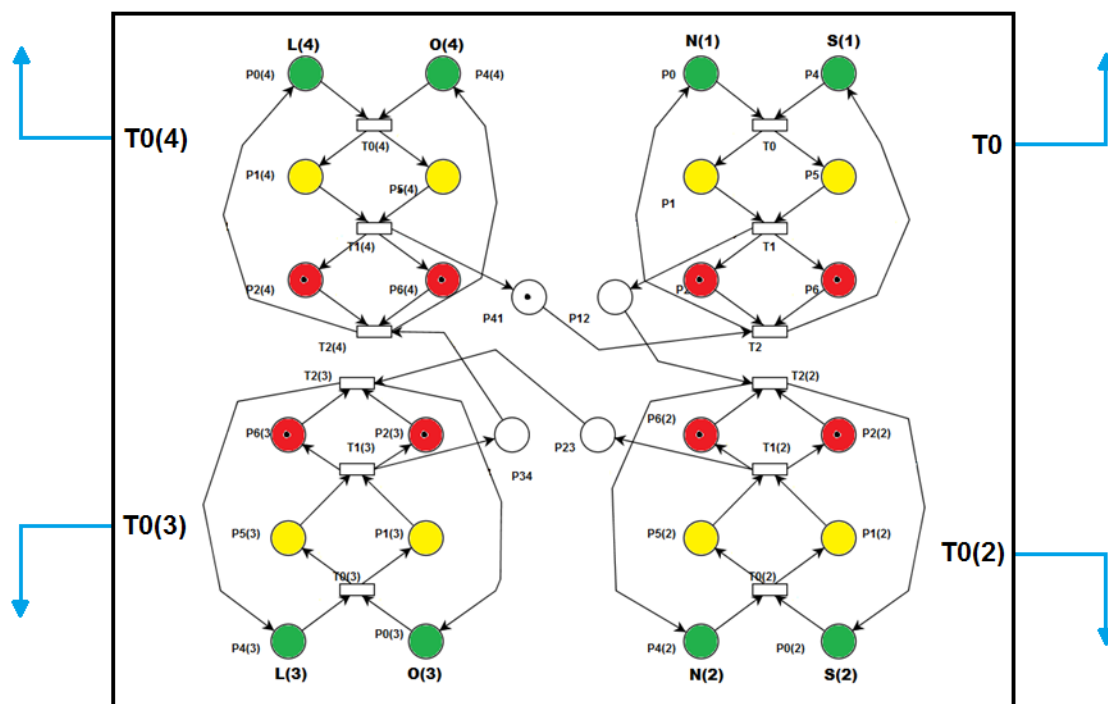


Figura 36 - Módulo do cruzamento central (*master*)

As demais intersecções foram implementadas como cruzamentos de 2 **fases** semelhantes ao Cruzamento 1, mas com o tempo de **estágio** verde definido pelos sinais de entrada recebidos do modelo *master*. O módulo típico com o modelo desses cruzamentos e os respectivos sinais de entrada está mostrado na Figura 37.

No modelo da Figura 37, com a chegada do sinal S1, os semáforos de direção norte-sul passam para o **estágio** amarelo e, em seguida, vermelho, passando ainda os semáforos de direção leste-oeste para o **estágio** verde. Com a chegada do sinal S2, a transição T0(S2) é disparada e o processo ocorre de forma análoga, estabelecendo um ciclo.

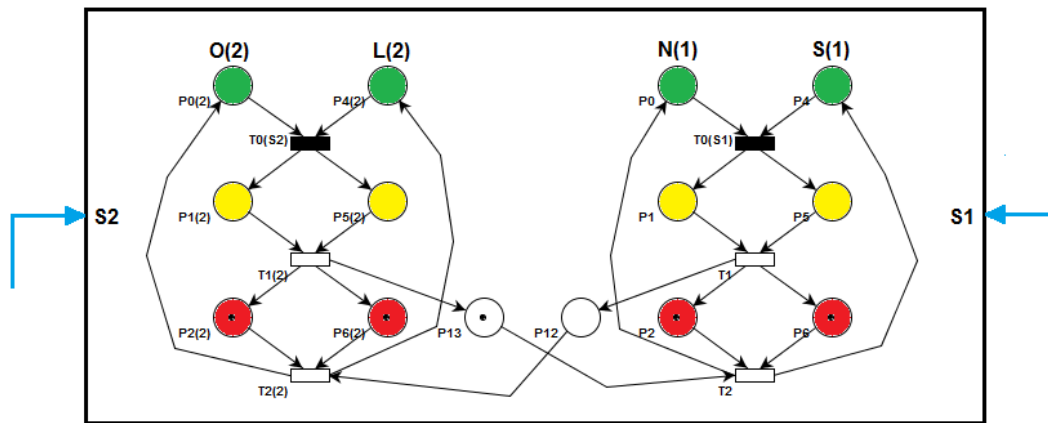


Figura 37 - Módulo dos cruzamentos slave

Os módulos mostrados nas Figuras 36 e 37 são interligados de forma a respeitar a sequência da Figura 35. O modelo final do conjunto de cruzamentos sincronizados pode ser visto na Figura 38. Nessa figura, a conexão dos módulos de cada cruzamento está representada de forma análoga à representação da conexão de circuitos integrados em uma placa. Os números dentro das caixas dos módulos mostram o número de **fases** do respectivo cruzamento.

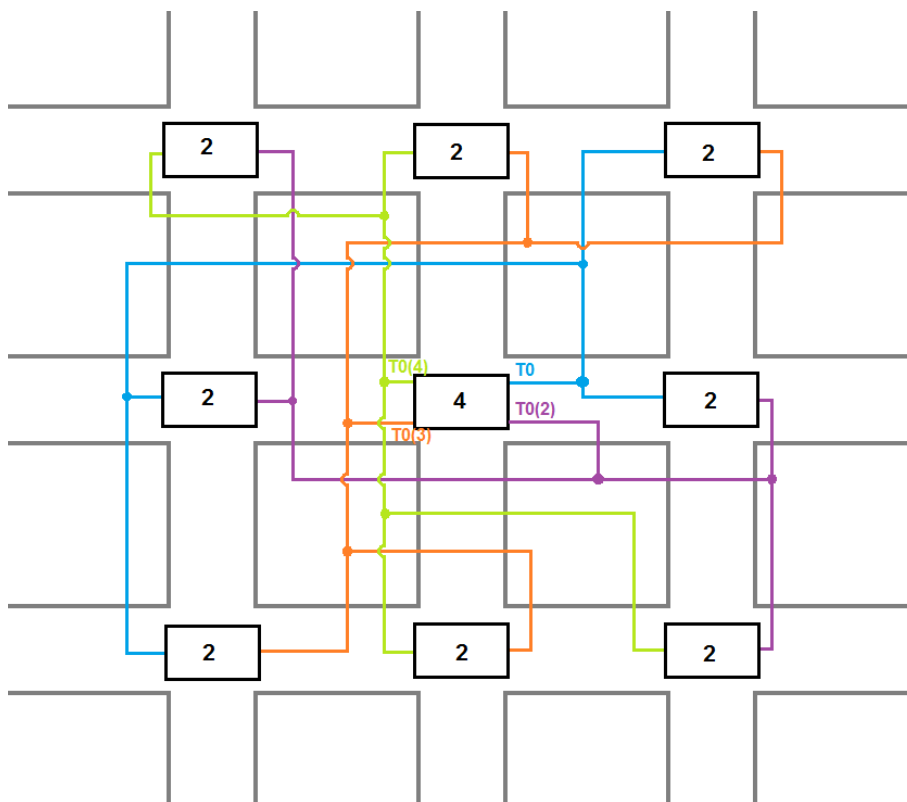


Figura 38 - Modelo do conjunto de cruzamentos interligados

A configuração mostrada na Figura 38 apresenta o comportamento ilustrado na Figura 35. Como pode ser notado nesse exemplo, a conexão entre os modelos dos cruzamentos é feita de forma relativamente simples.

Deve ser observado que os módulos podem ser facilmente adaptados e reaproveitados para outros conjuntos de cruzamentos. Nesse exemplo, o módulo da Figura 36 foi reaproveitado do Cruzamento 2 e o módulo da Figura 37 foi adaptado e reaproveitado do Cruzamento 1.

### 6.7. Conversão para linguagem LD

Os modelos de cruzamentos apresentados neste trabalho possuem o mesmo princípio de funcionamento, isto é, todos, após iniciarem o processo de abertura e fechamento de **fases**, entram em um ciclo infinito.

Um dos objetivos deste trabalho é a implementação do controle semafórico (desenvolvido em rede de Petri) por controladores programáveis. Ao converter-se um modelo em rede de Petri para um programa em linguagem *Ladder Diagram* (LD) de CLP's, pode ser acrescentado um mecanismo de reinicialização, feita à distância, de qualquer cruzamento semafórico que apresente qualquer tipo de falha. Para isto, é necessário o desenvolvimento de um mecanismo que possua “botões” de **start** (dispara o ciclo de funcionamento a partir de um estado seguro) e **reset** (retorna o sistema a uma posição pré-estabelecida).

O desenvolvimento desse mecanismo requer o uso de transições acionadas por eventos na rede de Petri. Desta forma, na conversão para a linguagem *Ladder Diagram* foi feita também a adição de “botões” de **start** e **reset**. O *software* PLCLogix, foi utilizado para construção e simulações do modelo LD desenvolvido. Esta ferramenta é de um simulador de Controladores Lógicos Programáveis, onde é possível escrever, editar e compilar programas baseados na lógica *Ladder*.

O modelo desenvolvido no PLCLogix, refere-se ao cruzamento de duas **fases** (Cruzamento 1). A realização da conversão do modelo em rede de Petri para linguagem LD segue o método proposto por Miyagi *et al.* (2011). Por se tratar de um método passo-a-passo, o processo de transformação para um programa em linguagem LD de qualquer outro cruzamento, se dá de maneira análoga.

Para fazer a conversão almejada, primeiramente cria-se variáveis que representam todos os lugares e transições existentes nos modelos em rede de Petri. Para o cruzamento de duas **fases**, isso significa que são criadas 20 variáveis auxiliares, pois o mesmo conta com 14 lugares e 6 transições. No entanto, este modelo conta com dois conjuntos de semáforos que possuem processos de funcionamento exatamente iguais – O(2), L(2) e N(1), S(1). De modo a simplificar a construção do programa em LD, considerou-se apenas um dos semáforos de cada conjunto, o que pode ser feito sem que haja perda de qualquer característica. Portanto, definiu-se 14 variáveis.

O *software* PLCLogix possui uma interface de visualização de múltiplas entradas e saídas discretas, apresentada na Figura 39. Nesta interface estão presentes dois módulos para entradas (Slot 1 e Slot 3) e dois para saídas (Slot 2 e Slot 4), cada um com dezesseis pontos de acesso, que são usados para armazenar as variáveis do problema.

Para o cruzamento de duas **fases** definiu-se que as **fases** dos semáforos Norte-Sul utilizariam as saídas 0, 1 e 2 do Slot 2, enquanto que Leste-Oeste ocupariam 4, 5 e 6. Os lugares P12 e P13, responsáveis por habilitar os **ciclos** de operação dos semáforos, apresentados na Figura 11, ocuparam-se das saídas 8 e 9 do Slot 2, respectivamente.

Nas entradas 0 e 1 do Slot 3 foram adicionados os “botões” de **reset** e **start**, não presentes nos diagramas em rede de Petri. As seis variáveis provenientes de transições do modelo original foram definidas como elementos *timer* na linguagem LD. O tempo de acionamento de cada *timer* pode ser redefinido com facilidade, alterando os tempos de abertura e fechamento dos semáforos do cruzamento conforme necessidade local.

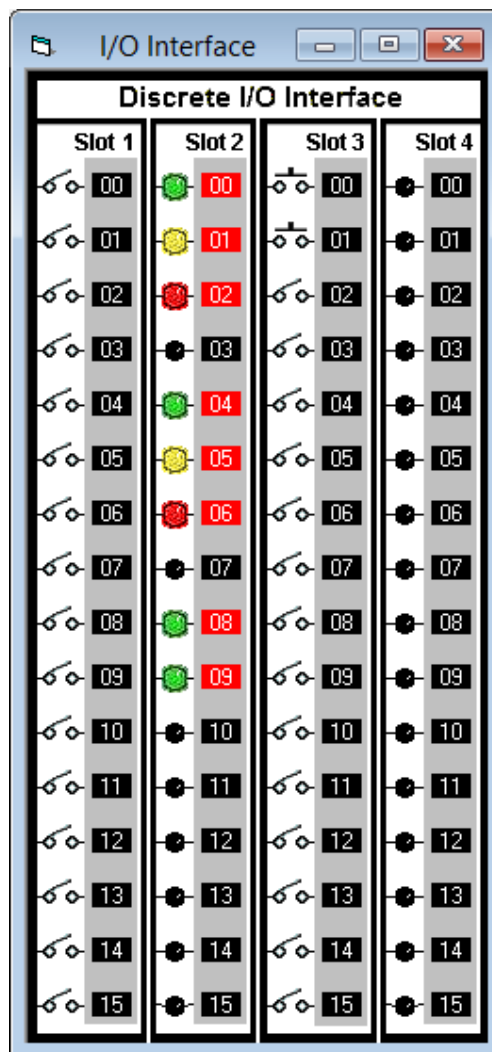
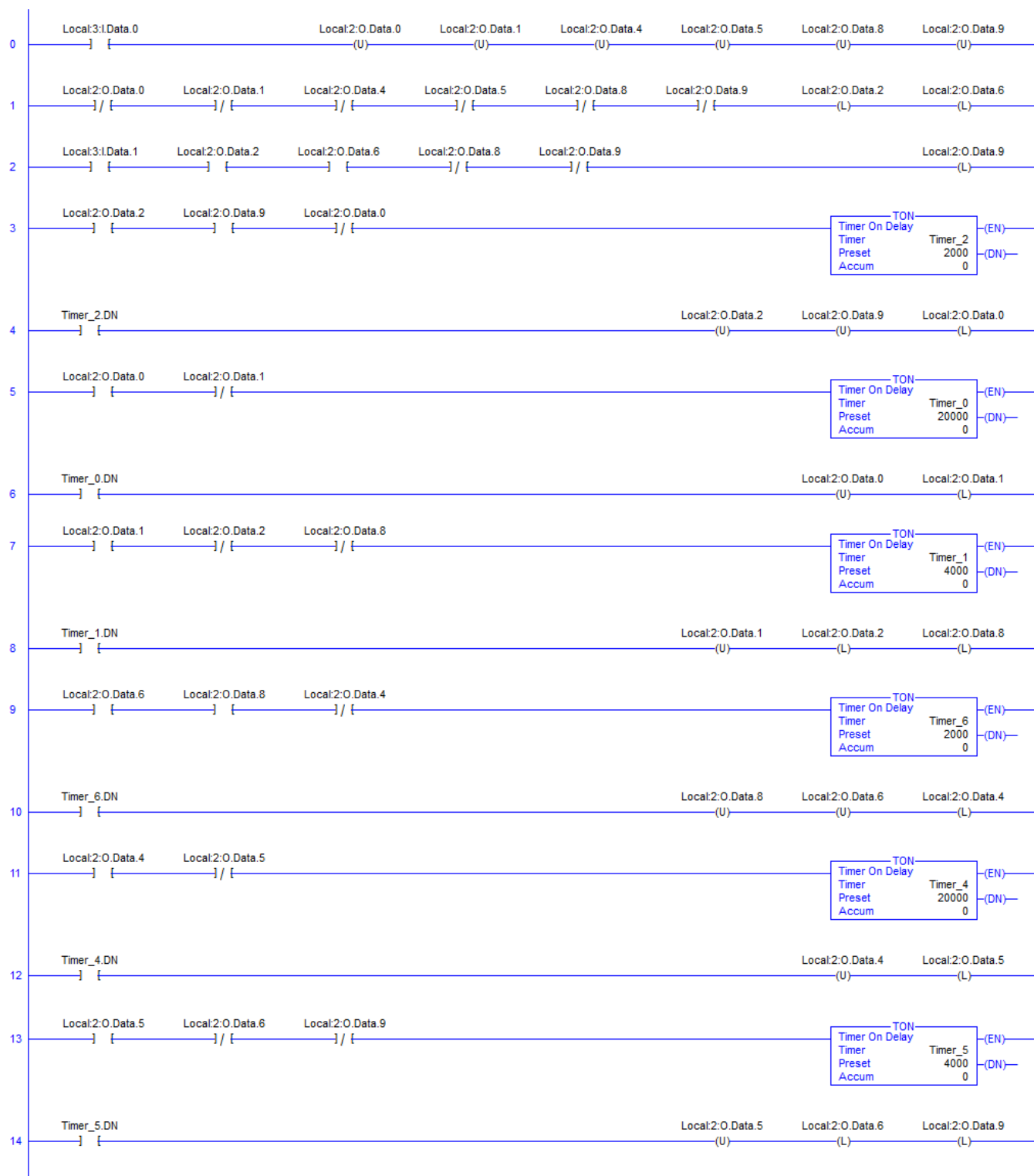


Figura 39 – Interface de entradas e saídas discretas do PLCLogix

Após a definição das variáveis entre contatos e *timers* (transições temporizadas), prossegue-se na montagem do diagrama. A construção do *Ladder* é intuitiva, consistindo na simples replicação do modelo de rede de Petri com uso das variáveis criadas e bobinas de *set* e *reset* para acionar de fato as saídas desejadas. Ao final deste processo, de acordo com o método proposto por Miyagi *et al.* (2011), definiu-se a condição inicial do programa, que nada mais é do que o “botão” de **start**, que já desejava-se implementar. Na Figura 40 é representado o modelo desenvolvido no *software* PLCLogix para o cruzamento de duas **fases**.



*Figura 40 – Modelo em Ladder Diagram (LD) do cruzamento de duas **fases***

O modelo da Figura 40 foi testado com os mesmos tempos de transição usados na simulação do cruzamento em rede de Petri. Na Figura 41 abaixo, está demonstrado um ciclo de funcionamento do cruzamento de duas **fases**, elaborado a partir de saídas da interface do PLCLogix.

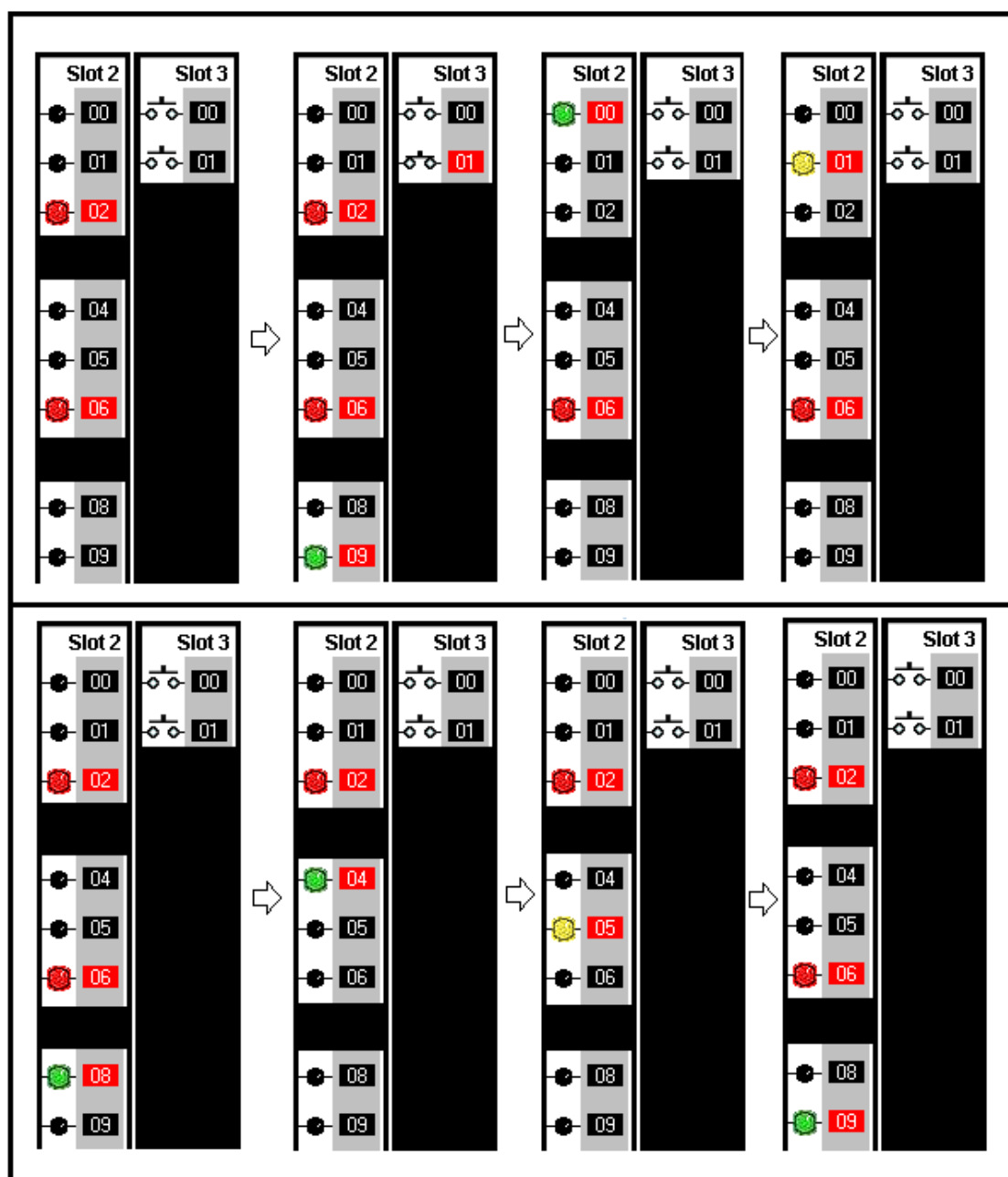


Figura 41 - Simulação do cruzamento em LD

A simulação representada na Figura 41 inicia-se com o sistema em estado de **reset**, com ambos os semáforos fechados e nenhum habilitador de **ciclo** (saídas 8 e 9) ligado. Em seguida observa-se que o botão **start** é pressionado (entrada 1), o que desencadeia na ativação da saída 9, responsável por habilitar o ciclo dos semáforos norte e sul. A partir deste momento o cruzamento entra em um loop infinito, que só será interrompido espontaneamente caso o botão de **reset** seja novamente pressionado.

Nota-se no primeiro quadro inferior da Figura 41, que há um período em que ambos os cruzamentos permanecem simultaneamente fechados, o que satisfaz os requisitos de projeto do modelo. A partir destes resultados é possível admitir que os modelos em redes de Petri e em linguagem LD são equivalentes e funcionam de modo apropriado.

## 7. Conclusão

O sistema semafórico presente na cidade de São Paulo é obsoleto e propenso a falhas. Apesar de já existirem cruzamentos com semáforos modernos e de alta tecnologia, sua aplicação ainda ocorre em baixa escala. Este projeto advém da possibilidade de se desenvolver modelos de cruzamentos que sejam aplicáveis em controladores lógicos programáveis e implementáveis em larga escala, para aumento da confiabilidade de sistemas semafóricos. O projeto possui como objetivo a análise e criação de modelos de cruzamentos semafóricos utilizando-se a rede de Petri.

A análise dos diversos modelos de cruzamentos com diferentes graus de complexidade desenvolvidos no projeto, indicam que a área de estudo que engloba o uso de controladores lógicos programáveis em cruzamentos semafóricos deve ser aprofundada. A rede de Petri mostrou-se ótima ferramenta para a representação de sistemas a eventos discretos, assim como importante porta de entrada para conversão de seus modelos para programas em linguagem *Ladder Diagram*. A validação dos modelos criados se deu por meio da análise dos resultados de simulações feitas com os mesmos, indicando tratar-se de um procedimento de baixo custo que pode ser iniciado imediatamente após o término do desenvolvimento de um modelo.

Além do aumento de confiabilidade e redução em tempos de manutenção de semáforos previsto com a implantação de CLP's em cruzamentos, este sistema também possui a possibilidade de se alterar períodos de **estágios** verde, amarelo e vermelho dos semáforos de um cruzamento, de acordo com a necessidade/vontade de seu operador, de modo a atuar como ferramenta de controle de tráfego.

O projeto atendeu aos objetivos propostos inicialmente.

## 8. Bibliografia

- BANKS, J. Discrete Event Simulation. Proceedings of the 1999 Winter Simulation Conference, pp. 7-13, Estados Unidos da América, 1999.
- CET. Sinalização semafórica – Audiência pública (2013). Em: <<http://www.cetsp.com.br/media/209128/reestruturacao semaforica.pdf>>. Acesso em: 22 de março de 2015.
- CHAVAN, S. S.; DESHPANDE, R. S.; RANA, J. G. Design of intelligent traffic light controller using embedded system. Second International Conference on Emerging Trends in Engineering and Technology. pp. 1086-1091, 2009.
- CHEN, F. et al. A novel hybrid Petri net model for urban intersection and its application in signal control strategy. Journal of the Franklin Institute. Vol. 351, Agosto de 2014, p. 4357–4380.
- CHRISTIANINI, W. A. C.; HAGIWARA, S. S. Operação de trânsito: um desafio permanente. São Paulo, Companhia de Engenharia de Tráfego, 2008. 56 p. – (Boletim Técnico da CET, 44).
- DENATRAN Manual de semáforos. 2ª ed., Brasília, 1984. 172 p.
- DINGLE, N. J.; KNOTTENBELT, W. J.; SUTO, T. PIPE2: A Tool for the Performance Evaluation of Generalised Stochastic Petri Nets. ACM SIGMETRICS Performance Evaluation Review (Special Issue on Tools for Computer Performance Modelling and Reliability Analysis), Vol. 36(4), 2009, p. 34-39.
- DOTOLI, M.; FANTI, M. P. An urban traffic network model via coloured timed Petri nets. Dipartimento di Elettrotecnica ed Elettronica, Politecnico di Bari, 2006.
- GOMES, A. P. Os Benefícios da Operação de Semáforos em Tempo Real. CET – São Paulo, Nota Técnica 236, 2014.
- HJALMARSSON, M.; JOHANSSON, S. Wireless remote control of a PLC system. Tese de mestrado para Chalmers University of Technology. Suécia, 2011.

- HP (HEWLETT-PACKARD). HP Systems Insight Manager Installation and Configuration Guide for Microsoft® Windows. 5th Edition. Houston, Texas. 2006
- HU, C.; WANG, Y. A Novel Intelligent Traffic Light Control Scheme. 2010 Ninth International Conference on Grid and Cloud Computing, pp. 372-376, 2010.
- HUANG, Y. S.; WENG, Y. S.; ZHOU, M. C. Modular Design of Urban Traffic-Light Control Systems Based on Synchronized Timed Petri Nets. IEEE Transactions on Intelligent Transportation Systems, Vol. 15, No. 2, 2014, p. 530-539.
- LI LIN et al. Implementation of Traffic Lights Control Based on Petri Nets. Xi'an Shiyou University, China, 2003.
- LIST, G. F. et al. Modeling Traffic Signal Control Using Petri Nets, IEEE Trans. on Intelligent Transportation Systems, vol. 5, pp 177-187, 2004.
- LUCAS, M. R.; TILBURY, D. M. Methods of measuring the size and complexity of PLC programs in different logic control design methodologies. The International Journal of Advanced Manufacturing Technology, London, p. 436-447, 2005.
- MCLEAN, V. A. Traffic Detector Handbook: Third Edition. U.S. Department of Transportation. Vol. 1. Outubro de 2006.
- MIYAGI, P. E. Controle Programável: fundamentos do controle de sistemas a eventos discretos. 1. ed. São Paulo: Editora Blucher, 1996.
- MIYAGI, P. E. Introdução à Simulação Discreta. Escola Politécnica da Universidade de São Paulo, 2006. Disponível Em: <[http://sites.poli.usp.br/d/pmr2460/arquivos/Apostila\\_Simulacao.pdf](http://sites.poli.usp.br/d/pmr2460/arquivos/Apostila_Simulacao.pdf)>. Acesso em: 04 de abril de 2015.
- MIYAGI, P. E. et al. A Transcription Tool From Petri Net to CLP Programming Languages. Proceedings of COBEM 2011, 21<sup>st</sup> Brazilian Congress of Mechanical Engineering, October 24-28, 2011, Natal, RN, Brazil.
- MURATA, T. Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE, Vol. 77, No. 4, 1989, p. 541-580.
- NEW SOUTH WALES GOVERNMENT, Roads & Maritime Services, 2015. Em: <<http://www.scats.com.au/why-choose-scats-performance.html>>. Acesso em: 14 de abril de 2015.

- ROBERTSON, R.; BRETHERTON., R. D. Optimizing networks of traffic signals in real time—the SCOOT method, IEEE Trans. Veh. Technol., vol. 40, pp. 11-15, 1991.
- ROESS, R. P.; PRASSAS, E. R.; McSHANE, W. R. Traffic Engineering. Pretince Hall, 3ª ed., 2004.
- SIMS, A. G., DOBINSON, K. W. The Sydney Coordinated Adaptive Traffic (SCAT) System Philosophy and Benefits, IEEE Transactions on Vehicular Technology, Vol. VT-29, NO.2, May 1980.S
- SMITH, S. F.; BARLOW, G. J.; XIE, X. F.; RUBINSTEIN, Z. B. SURTRAC: Scalable Urban Traffic Control. Transportation Research Board 92nd Annual Meeting Compendium of Papers, 2013.
- TEIXEIRA, Célia Regina. O “Estado da Arte”: a concepção de avaliação educacional veiculada na produção acadêmica do Programa de Pós-Graduação em Educação: Currículo (1975-2000). *Cadernos de Pós-Graduação – educação*, São Paulo, v. 5, n. 1, p. 59-66, 2006.
- VILANOVA, L. M. O controle de semáforos em tempo real: a experiência de São Paulo. São Paulo: Companhia de Engenharia de Tráfego, 2005. 70 p. (Boletim técnico da CET, 38).
- VILANOVA, L. M. O controle de Semáforos em Tempo Real no Brasil. Em: < [http://w3.ualg.pt/~mgameiro/Aulas\\_2006\\_2007/transportes/Paper\\_semaforos.pdf](http://w3.ualg.pt/~mgameiro/Aulas_2006_2007/transportes/Paper_semaforos.pdf) >. Acesso em 24 mar. 2015.
- WINSTON, W. L. Operations Research, Applications and Algorithms, 4th ed., Cap. 20. Editora Brooks/Cole - Thomson Learning, Indiana University, 2004.
- XIE, Y. Design of Urban Intelligent Traffic Signal Control System for Large Special Events. Third International Conference on Information Science and Technology, pp. 1149-1151, 2013.
- ZHANG, L. G.; LI, Z. L.; CHEN, Y. Z. Hybrid Petri Net Modeling of Traffic Flow and Signal Control. Beijing University of Technology, China, 2008.