Revista Multidisciplinar do Nordeste Mineiro,
v4,2023/04
ISSN 2178-6925

USO DE REDES DE PETRI PARA CONTROLE DE SEMÁFOROS

USE OF PETRI NETS TO CONTROL TRAFFIC LIGHTS

Jéssica Flávia dos Reis
Especialista em docência, Microempreendedora, Brasil
E-mail: jessicaf.reis@outlook.com

Rômulo Jesuíno da Silva
Especialista em Manutenção, Gerdau, Brasil
E-mail: romulojesuino@hotmail.com

Guilherme Gomes da Silva
Doutor em Engenharia Elétrica, UFSJ, Brasil
E-mail: guilhermegomes@ufsj.edu.br

Recebimento 11/04/2023 Aceite 18/04/2023

Resumo

Analisar e pesquisar o trânsito em centros urbanos é algo em constante crescimento devido aos congestionamentos em vias públicas, os quais vem se tornando um problema considerável para a gestão do tráfego. Nesse contexto, a utilização de simuladores computacionais para auxilio ao planejamento da mobilidade urbana é vem aumentando. Diante disso, esse trabalho tem como objetivo apresentar um tutorial para modelar e controlar os semáforos presentes em um cruzamento, através das Redes de Petri (RP). Essa escolha ocorreu porque tais redes são capazes de descrever o comportamento dinâmico das luzes pertencentes a cada semáforo, devido sua natureza discreta. Outra justificativa é a potencial aplicação prática das RP. Como resultado, foram evidenciados os procedimentos para a elaboração da modelagem e do controle dos semáforos, podendo ser utilizado por outros pesquisadores como um tutorial para aplicação dessa técnica e como base para novas pesquisas sobre o tema. Além disso, a Rede de Petri obtida após as pesquisas foi incorporada no ambiente computacional HPSim, onde foi testado seu funcionamento.
Palavras-chave: Trânsito; Controlador; Simulação; Tutorial.

Abstract

Analyzing and researching traffic in urban centers is constantly growing due to congestion on public roads, which is becoming a considerable problem for traffic management. In this context, the use of computational simulators to help plan urban mobility is increasing. Therefore, this work aims to present a tutorial to model and control the traffic lights present at an intersection, through Petri Nets (RP). This choice was made because such networks are capable of describing the dynamic behavior of the lights belonging to each traffic light, due to their discrete nature. Another justification is the potential practical application of PR. As a result, the procedures for the elaboration of the modeling and control of traffic lights were evidenced, which can be used by other researchers as a tutorial for the application of this technique and as a basis for further research on the subject. In addition, the Petri Net obtained after the research was incorporated into the HPSim computational environment, where its operation was tested.
Keywords: Traffic; Controller; Simulation; Tutorial.

1. Introdução

Congestionamentos são um problema para uma mobilidade urbana mais efetiva (PONTES *et al.*, 2022). Uma forma de evitá-los é através do funcionamento eficiente de semáforos, de acordo com as necessidades das vias. Diante disso, o tema abordado no presente artigo é a modelagem e o controle de semáforos, através de RP, sendo o objetivo geral a criação de um tutorial capaz de explicitar com clareza os procedimentos necessários para tal atividade, o qual poderá ser utilizado por outros pesquisadores do tema.

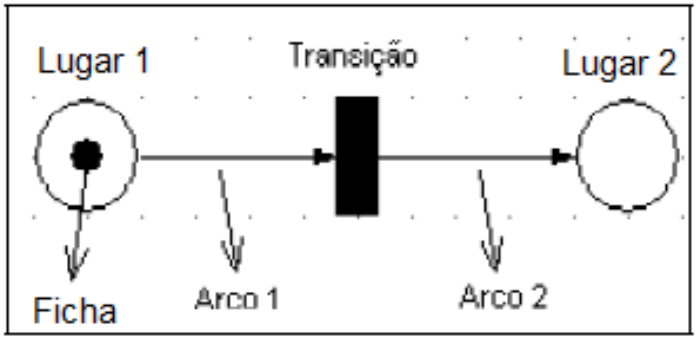
Já os objetivos específicos foram: identificar conceitos sobre a técnica escolhida; detalhar a situação a ser considerada na modelagem e no controle; simular em ambiente computacional os resultados obtidos. A principal hipótese verificada foi que o através do controle por invariância de lugar os semáforos podem ser controlados. Justifica-se a escolha do tema por sua relevância e necessidade de estudos na área da mobilidade urbana.

O artigo tem a seguinte organização: na seção 2 foram abordados conceitos sobre das RP e suas principais características enquanto que na seção 3 foi apresentado o cruzamento a ser modelado e as etapas de elaboração do controlador foram descritas, mostrando a sequência como um tutorial. Na seção 4 foi apresentado o controlador através de simulação computacional e, na seção 5 foram feitas as conclusões do artigo. Por fim, foram listadas as referências utilizadas para a construção dessa pesquisa.

2. Revisão da literatura

Antes de evidenciar a aplicação prática das RP, foi necessário discutir alguns aspectos conceituais sobre as mesmas. Diante disso, em uma Rede de Petri há um conjunto de elementos, os quais são capazes de descrever as diferentes partes de um sistema. Estes são tratados em ciência da computação tais como: concorrência, controle, conflitos, sincronização e compartilhamento (OLIVEIRA, 2022). A composição de uma Rede de Petri é evidenciada na Figura 1, sendo os elementos mostrados graficamente: lugares, fichas, transições e arcos.

Figura 1 – Composição de uma RP



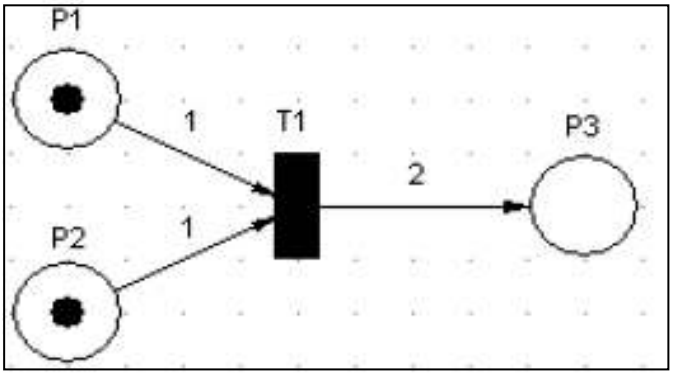
Fonte: Adaptado de Oliveira (2022).

Lugares ou *places,* representados graficamente pelos círculos maiores vazados, indicam um recurso, uma condição ou uma atividade do sistema. Já as fichas, marcas ou *tokens, r*epresentadas graficamente por um círculo menor completamente preenchido dentro do Lugar 1, indicam o estado atual do sistema analisado. Transições, representadas graficamente pelo retângulo preenchido, indicam um evento do sistema. Por fim, os arcos, representados por setas, indicam lugares os quais podem ser de entrada ou de saída para as transições (ABREU, 2022).

Quando se modela um sistema através das RP procura-se representar os estados antes e depois de cada evento neste sistema, sendo que, dos elementos mostrados, o responsável por isso é o *token* (OLIVEIRA, 2022). Dessa forma,

espera-se que uma RP possua uma marcação inicial indicativa do seu estado momentâneo, descrita através da notação *M0 = (1, 0)* ao considerar o exemplo gráfico da Figura 1. Essa notação evidencia que no Lugar 1 há um *tokem*, enquanto que no Lugar 2 não há nenhuma marca, como pode ser comprovado na Figura utilizada como exemplificação. Em uma Rede de Petri também pode haver pesos diferentes associados a cada arco, como visto no exemplo mostrado na Figura 2.

Figura 2 – Pesos diferentes nos arcos de uma RP



Fonte: Adaptado de Reis e Silva (2016).

Quanto a sua definição numérica, uma Rede de Petri pode ser conceituada como o conjunto *R = {M0, PA, A, T, P}* sendo que *M0* é a marcação inicial do sistema; *PA* são os pesos de cada arco; *A* é o conjunto dos arcos; *T* é o conjunto das transições e P é o conjunto dos lugares. Outra definição a RP Temporizada Determinística é aquela que, para cada transição se atribui um valor de tempo (ANDRADE, 2008).

3. Metodologia

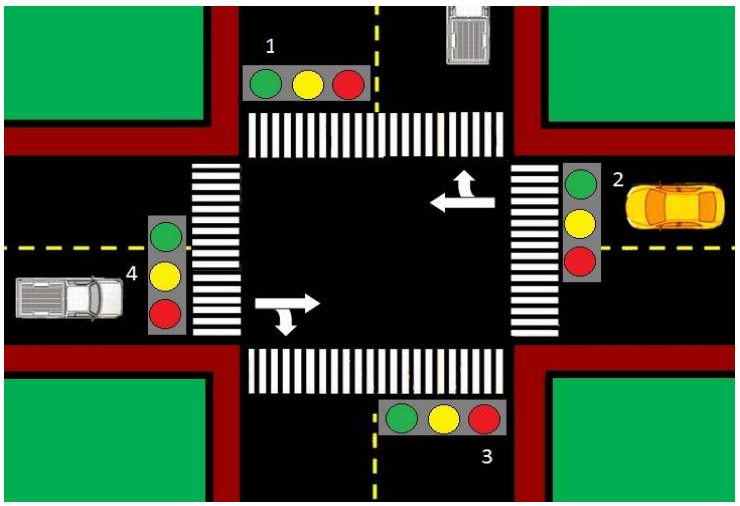
Para indicar o método utilizado na modelagem dos semáforos, foi necessário em primeiro lugar analisar o cruzamento a ser controlado. A não compreensão correta do sistema pode ter como consequência negativa a produção de um modelo o qual não reflete às ações reais e, portanto, não válido. Além da verificação do sistema, mostrada na seção 3.1, os procedimentos para

obtenção do controlador foram evidenciados na seção 3.2 como um tutorial que explicita as ações feitas.

3.1 Verificação do sistema

O sistema estudado trata-se de um cruzamento onde há quatro semáforos, conforme Figura 3. A modelagem dos semáforos foi feita em duplas, visto que o funcionamento desses pares ocorre assim: quando semáforos 1 e 3 estão verdes os semáforos 2 e 4 estão vermelhos; quando semáforos 1 e 3 estão amarelos, os semáforos 2 e 4 estão vermelhos; quando semáforos 1 e 3 estão vermelhos, os semáforos 2 e 4 estão verdes.

Figura 3 – Representação dos semáforos a serem modelados



Fonte: Autores (2023).

Para simplificar o projeto, foi adotado que a sigla *S1* refere-se aos semáforos 1 e 3, enquanto *S2* refere-se aos semáforos 2 e 4. O controlador foi feito também considerando que no estado inicial *S1* estava com o sinal verde acionado enquanto *S2* estava com o sinal vermelho ativo, evitando assim acidentes por falha na sincronização das cores dos mesmos.

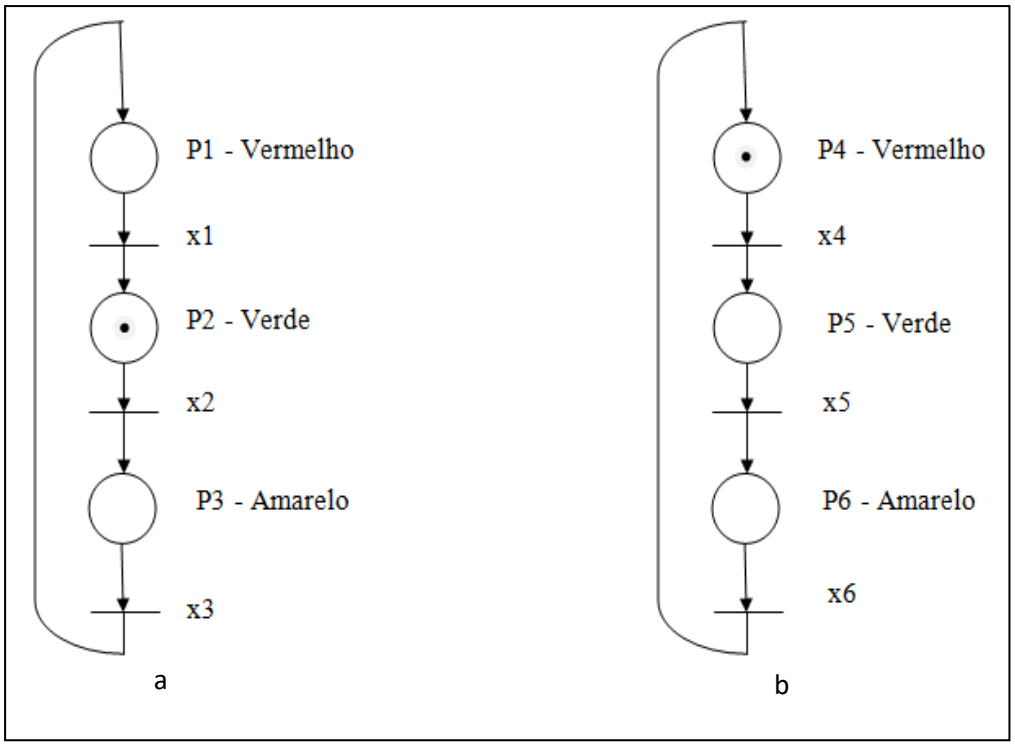
3.2 Procedimentos para elaboração do controlador

As etapas para a elaboração do controle com apoio das Redes de Petri foram: criar RP isoladas para cada um dos semáforos; elaborar a matriz de controle e de incidência do cruzamento; desenvolver a matriz incidência de controle e a sua marcação inicial; obter a RP do controlador (SILVA, 2014). Cada um desses passos foi detalhado nos tópicos a seguir.

3.2.1 RP isoladas

Para controlar os semáforos do cruzamento, foram elaboradas duas RP isoladas primeiramente, seguindo estudo de Silva (2014) além de Reis e Silva (2016), sendo elas apresentadas na Figura 4, onde a primeira é relativa a *S1* e a segunda relativa a *S2*.

Figura 4 – RP isoladas. 4a semáforos S1 e 4b semáforos S2



Fonte: Autores (2023).

3.2.2 Matrizes do sistema

A primeira matriz elaborada foi a de restrições, sendo ela feita pelo método de invariância de lugar. Em uma Rede de Petri com *n places* e *m* transições, o controle tem como objetivo forçar a obediência das regras impostas pelo processo a ser controlado (SILVA, 2014). Isso é feito através da Equação (1):

$$Lx \leq b \rightarrow Lp \leq b \tag{1}$$

As regras que precisam ser seguidas são consideradas como um conjunto restritivo mutuamente excludente (REIS; SILVA, 2016). Exemplificando o cruzamento analisado, jamais poderia ter todas as indicações luminosas verdes acionadas, o que resulta na Equação (2):

$$P2 + P5 \leq 1 \tag{2}$$

Nesse exemplo, a linha inicial da matriz de restrição mostrada na Equação 2 indica que *P2=1*, *P5=1* e demais pares de semáforos são nulos. As demais linhas da matriz restritiva seguem essa lógica, restringindo o acionamento das demais luzes que não podem ser ligadas simultaneamente. A matriz de restrições completa é mostrada na Figura 5:

Figura 5 – Matriz restritiva

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} P1 \\ P2 \\ P3 \\ P4 \\ P5 \\ P6 \end{bmatrix} \leq 1$$

Fonte: Autores (2023).

A segunda matriz necessária é a chamada incidência transposta que é a relação entre *places* e transições (REIS; SILVA, 2016). Quando a transição *x2* é acionada, retira-se um *token* de *P2* e insere-se um *token* em *P3*. Isso foi

demonstrado através da coluna 2 da matriz de incidência gerada, onde *P2=-1* e *P3=1*. Como os outros *places* não são influenciados pelo acionamento dessa transição, eles são nulos. A elaboração das outras colunas ocorreu através dessa lógica detalhada, sendo a matriz de incidência transposta completa mostrada na Figura 6 abaixo.

Figura 6 – Matriz de incidência transposta

	x1	x2	x3	x4	x5	x6
P1	-1	0	1	0	0	0
P2	1	-1	0	0	0	0
P3	0	1	-1	0	0	0
P4	0	0	0	-1	0	1
P5	0	0	0	1	-1	0
P6	0	0	0	0	1	-1

Fonte: Autores (2023).

3.2.3 Elaboração do controle

A partir das matrizes já elaboradas, foi possível desenvolver a matriz do controlador. Ela é calculada por meio da Equação (3) onde *D* é a matriz de incidência transposta do sistema a ser controlado, *L* é a matriz restritiva também desse mesmo sistema e *Dc* é a matriz de incidência transposta do controlador que necessita ser calculada (REIS; SILVA, 2016). Esse cálculo é evidenciado nas Figuras 7, 8 e 9, obtido através de programa computacional matemático.

$$D_c = -LD \tag{3}$$

Figura 7 – Matrizes aplicadas na Equação 3

$$Dc = - \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} . \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

Fonte: Autores (2023).

Figura 8 – Inserção da matriz *L* no programa computacional

```
>> L=[ 0 -1 0 0 -1 0; 0 0 -1  0 0 -1; 0 -1  0 0 0 -1;  0 0  -1 0 -1 0; -1 -1 -1 0 0 0; 0 0 0 -1 -1 -1]

L =

     0    -1     0     0    -1     0
     0     0    -1     0     0    -1
     0    -1     0     0     0    -1
     0     0    -1     0    -1     0
    -1    -1    -1     0     0     0
     0     0     0    -1    -1    -1
```

Fonte: Autores (2023).

Figura 9 – Inserção de *D* em programa computacional e cálculo de *Dc*

```
>> D=[-1 0 1 0 0 0; 1 -1 0 0 0 0; 0 1 -1 0 0 0; 0 0 0 -1 0 1; 0 0 0 1 -1 0; 0 0 0 0 1 -1]

D =

    -1     0     1     0     0     0
     1    -1     0     0     0     0
     0     1    -1     0     0     0
     0     0     0    -1     0     1
     0     0     0     1    -1     0
     0     0     0     0     1    -1

>> L*D

ans =

    -1     1     0    -1     1     0
     0    -1     1     0    -1     1
    -1     1     0     0    -1     1
     0    -1     1    -1     1     0
     0     0     0     0     0     0
     0     0     0     0     0     0
```

Fonte: Autores (2023).

Quanto à marcação inicial do controlador, ela é também calculada, sendo que para isso utiliza-se a Equação (4). Nela, *b* é referente ao termo isolado da matriz de restrições unitário, *Xo* é uma matriz coluna zerada, *L* refere-se à matriz restritiva e x_{co} é a marcação a ser obtida (SILVA, 2014). Também através de programa computacional, tem-se os resultados evidenciados nas Figuras 10 a 12. Por fim, na Figura 13, é mostrada a RP obtida.

$$x_{co} = b - Lx_0 \tag{4}$$

Figura 10 – Matrizes aplicadas na Equação 4

$$Xco = 1 - \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} . \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Fonte: Autores (2023).

Figura 11 – Inserção da matriz *L* no programa computacional

```
>> L= [0 1 0 0 1 0; 0 0 1 0 0 1; 0 1 0 0 0 1; 0 0 1 0 1 0; 1 1 1 0 0 0; 0 0 0 1 1 1]

L =

     0     1     0     0     1     0
     0     0     1     0     0     1
     0     1     0     0     0     1
     0     0     1     0     1     0
     1     1     1     0     0     0
     0     0     0     1     1     1
```

Fonte: Autores (2023).

Figura 12 – Cálculo em programa computacional da marcação inicial

```
>> X=[0;0;0;0;0;0]

X =

     0
     0
     0
     0
     0
     0
```

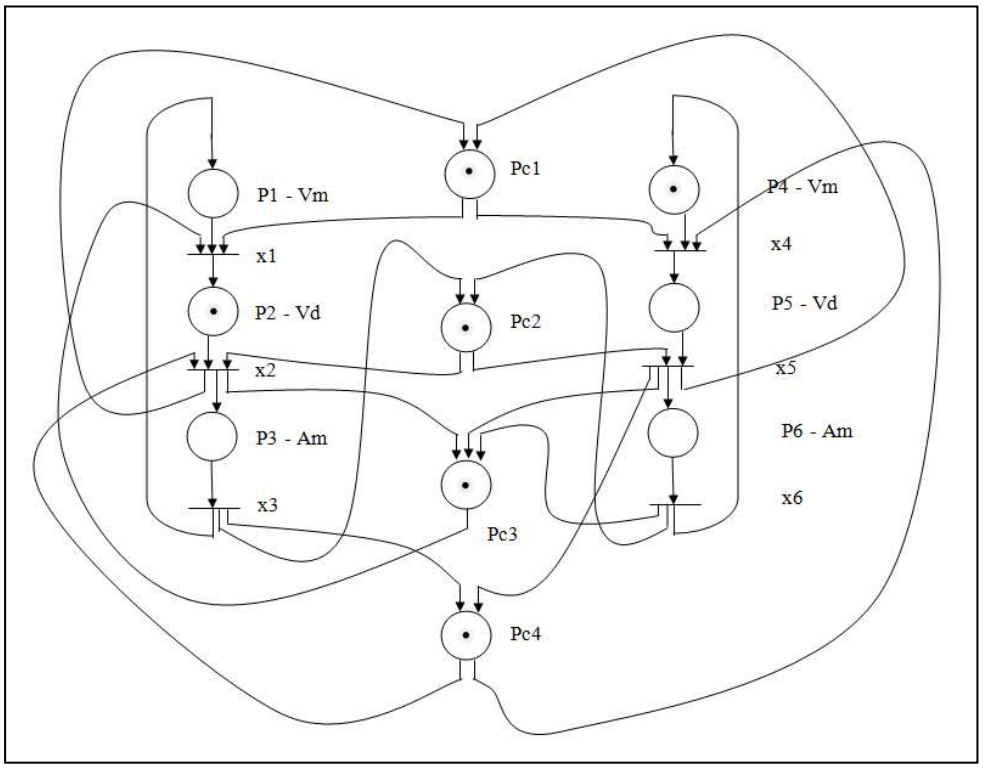
```
>> 1-L*X

ans =

     1
     1
     1
     1
     1
     1
```

Fonte: Autores (2023).

Figura 13 – RP obtida



Fonte: Autores (2023).

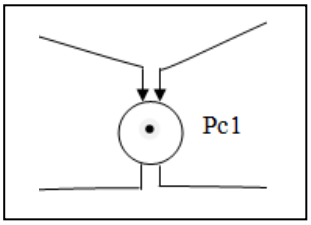
4. Resultados e discussão

Os semáforos foram modelados como RP temporizadas. Cada transição xi (i=1,2,3,4,5,6) está associada a um tempo de disparo. Por exemplo: semáforos 1 e 3 verdes por 15s enquanto semáforos 2 e 4 vermelhos por 15s; semáforos 1 e 3 amarelos por 5s enquanto semáforos 2 e 4 permanecem em vermelhos por mais 5s; semáforos 1 e 3 vermelhos por 15s enquanto semáforos 2 e 4 verdes por 15s; semáforos 1 e 3 vermelhos por mais 5s enquanto semáforos 2 e 4 amarelos por 5s; reinício do ciclo.

Para simulação, foi necessário verificar os tempos de operação dos semáforos sendo que no sistema em questão são: vermelho 20s, verde 15s e amarelo 5s. É através da simulação feita por álgebra Max-plus que se determinam os tempos dentro de uma dada RP, conforme estudos de Silva (2014) além de Reis e Silva (2016).

A álgebra Max-Plus, que é uma ferramenta matemática capaz que descreve equações em espaço de estados uma classe das RP chamada de Grafos de Eventos Temporizados, não pode ser utilizada no caso do semáforo descrito nesse trabalho pois a RP obtida possui mais de duas transições em pelo menos um dos seus locais (SILVA, 2014). Tal questão é evidenciada na Figura 14, sendo este apenas um exemplo ilustrativo, contendo mais dessa situação na RP que contém os semáforos e o controlador.

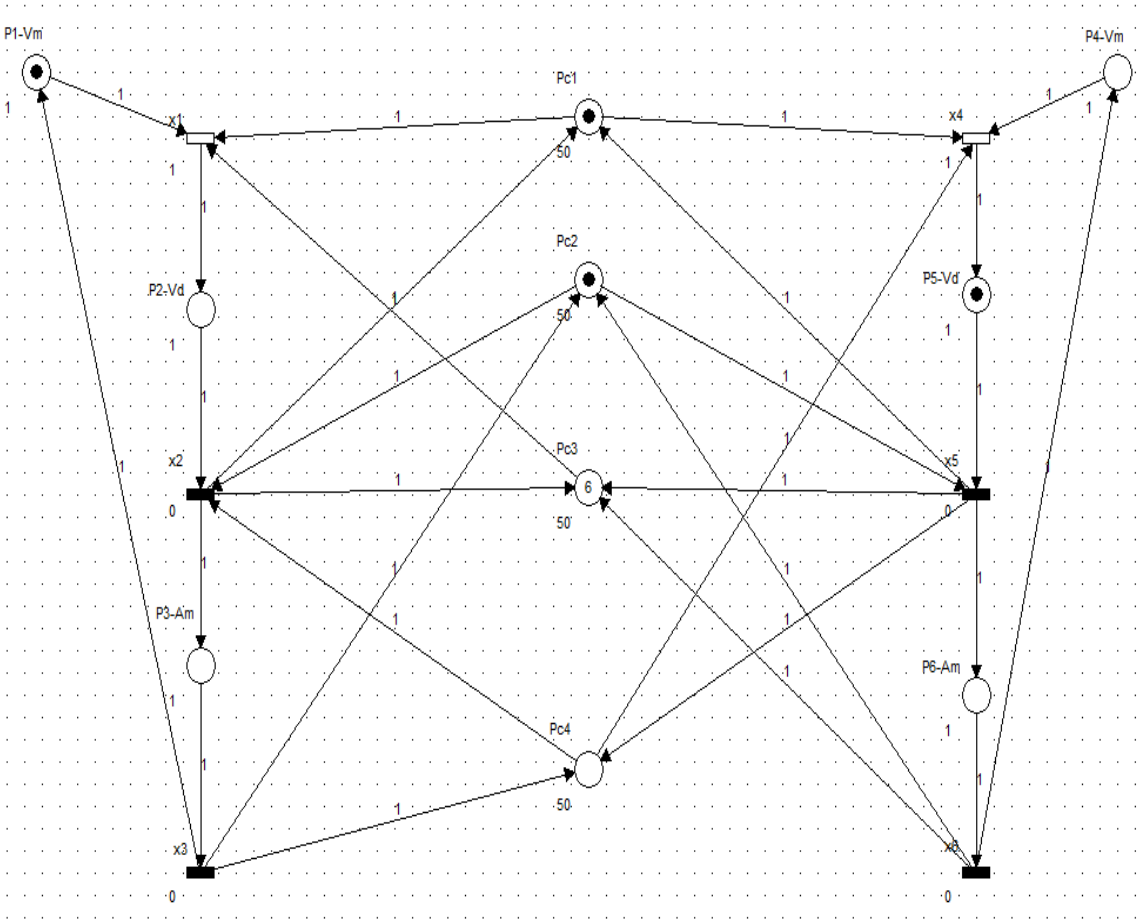
Figura 14 – Exemplo do motivo de não usar álgebra Max-plus



Fonte: Autores (2023).

Com o auxílio da ferramenta computacional HPSim (HPSIM, 2023) foi realizada a simulação da RP, sendo que a rede obtida e inserida no programa é evidenciada na Figura 15.

Figura 15 – Rede de Petri no HPSim

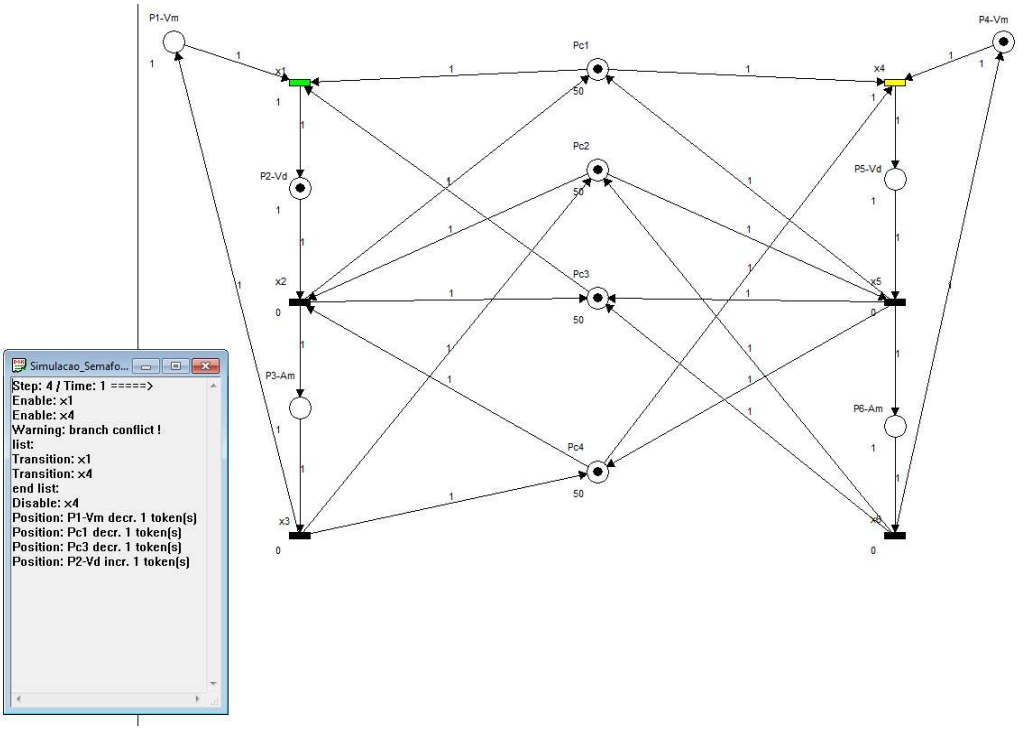


Fonte: Autores (2023).

Comparando-se a técnica de invariância de lugar com trabalhos feitos através da álgebra Max-Plus, conforme apresentado por Andrade (2008) pode-se observar que a álgebra tem como resultado final um controle mais otimizado. Outra situação que cabe destaque é que a simulação não é perfeita pois no programa não há como aplicar tempo de acendimento de cada lâmpada dos semáforos, porém ela evidencia que o controle foi bem executado, visto que as restrições impostas foram atendidas.

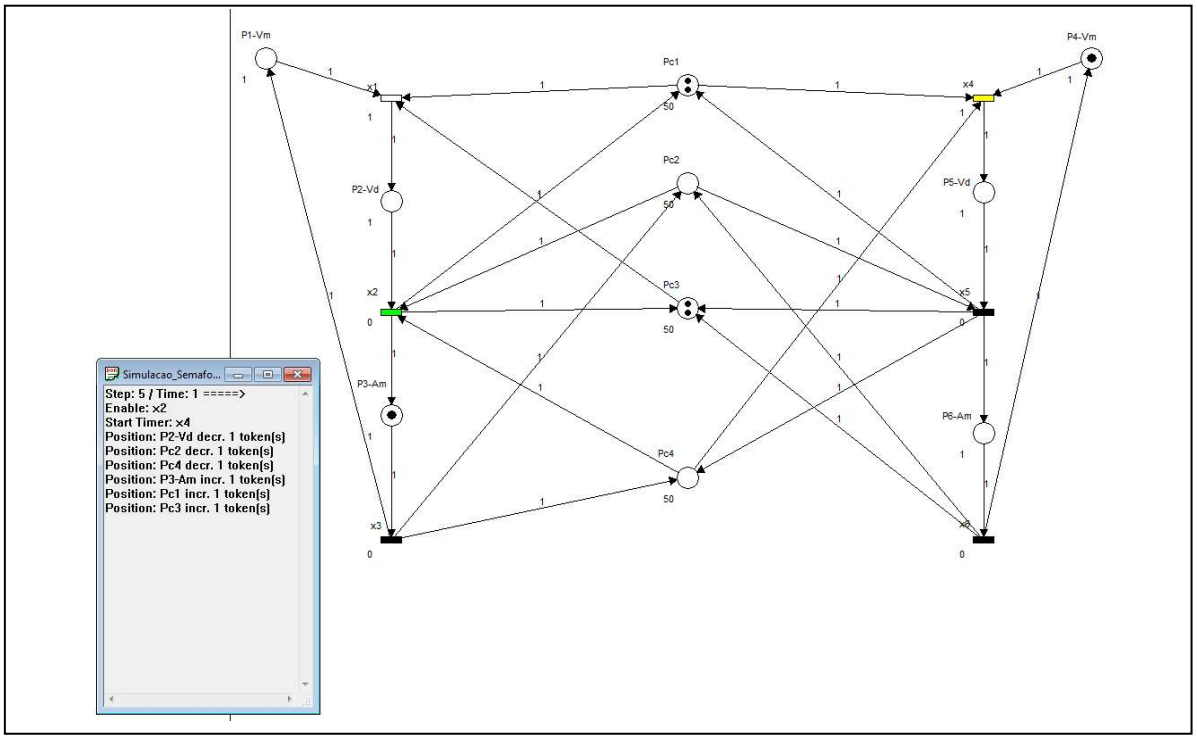
O controlador obtido, cujo funcionamento no HpSim é mostrado nas Figuras 16 a 23, apesar de ser eficiente pode ser melhor testado em outros simuladores computacionais os quais possam também incluir os tempos de cada sinalização dos semáforos.

Figura 16 – Simulação: S1 verde e S2 vermelho



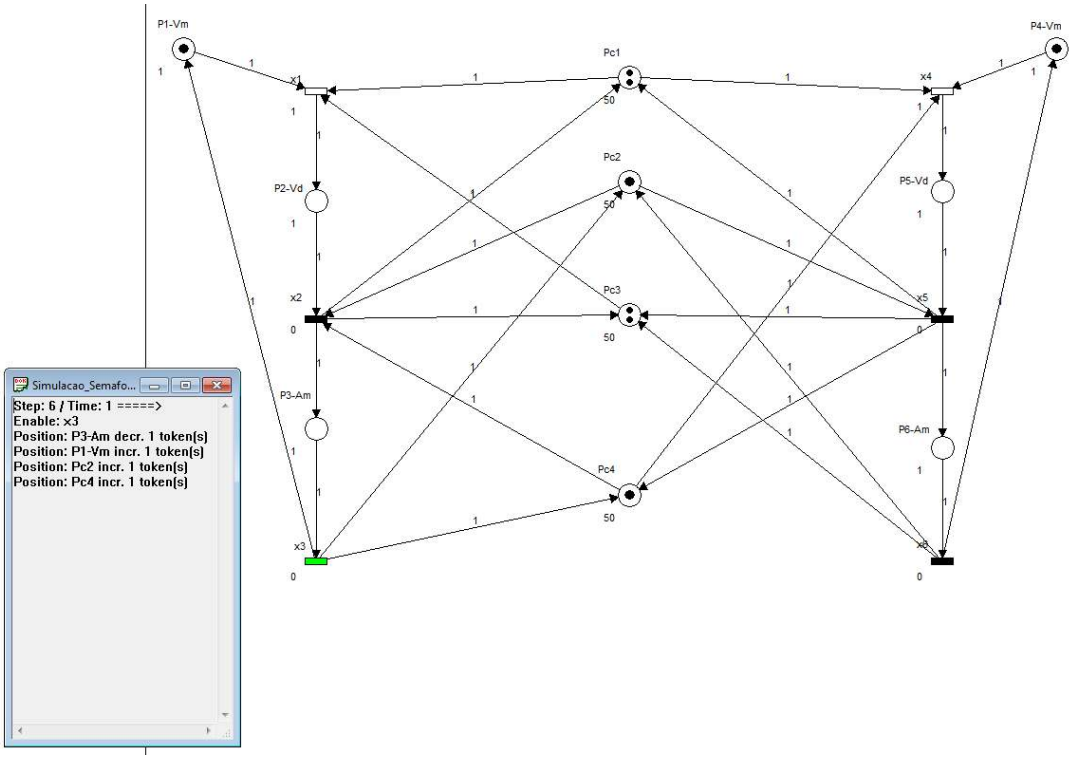
Fonte: Autores (2023).

Figura 17 – Simulação: S1 amarelo e S2 vermelho



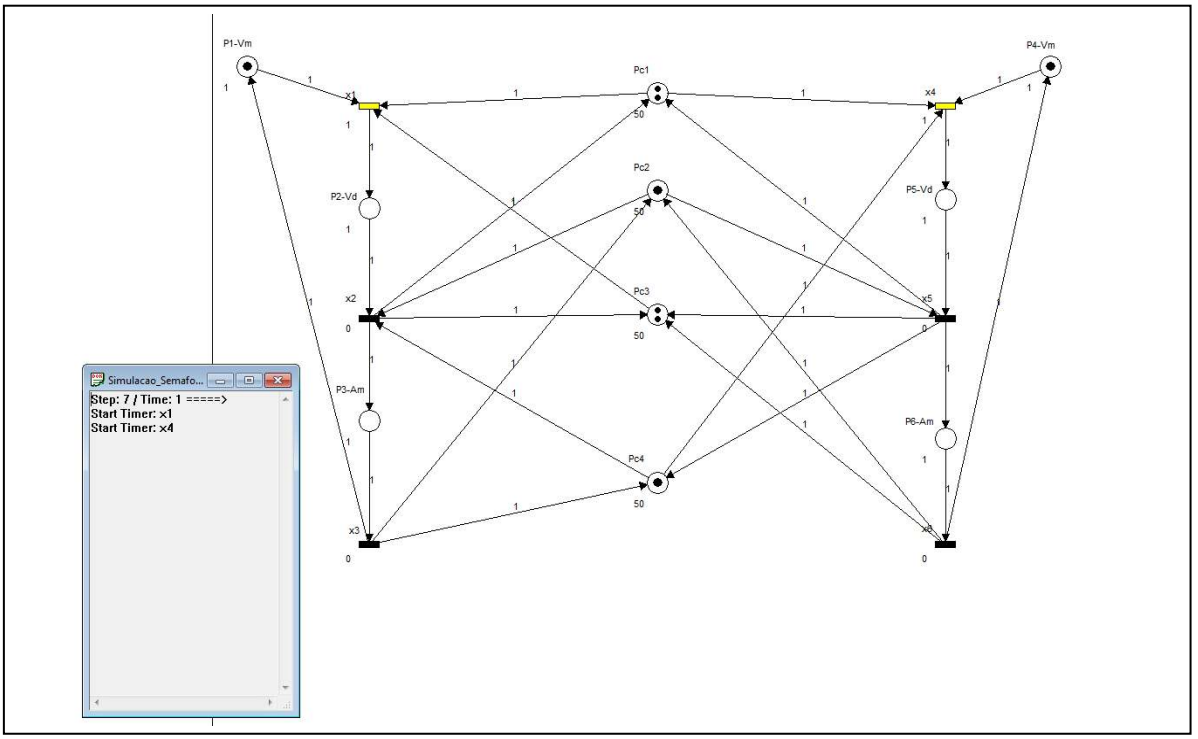
Fonte: Autores (2023).

Figura 18 – Simulação: S1 vermelho e S2 vermelho rápido para transição



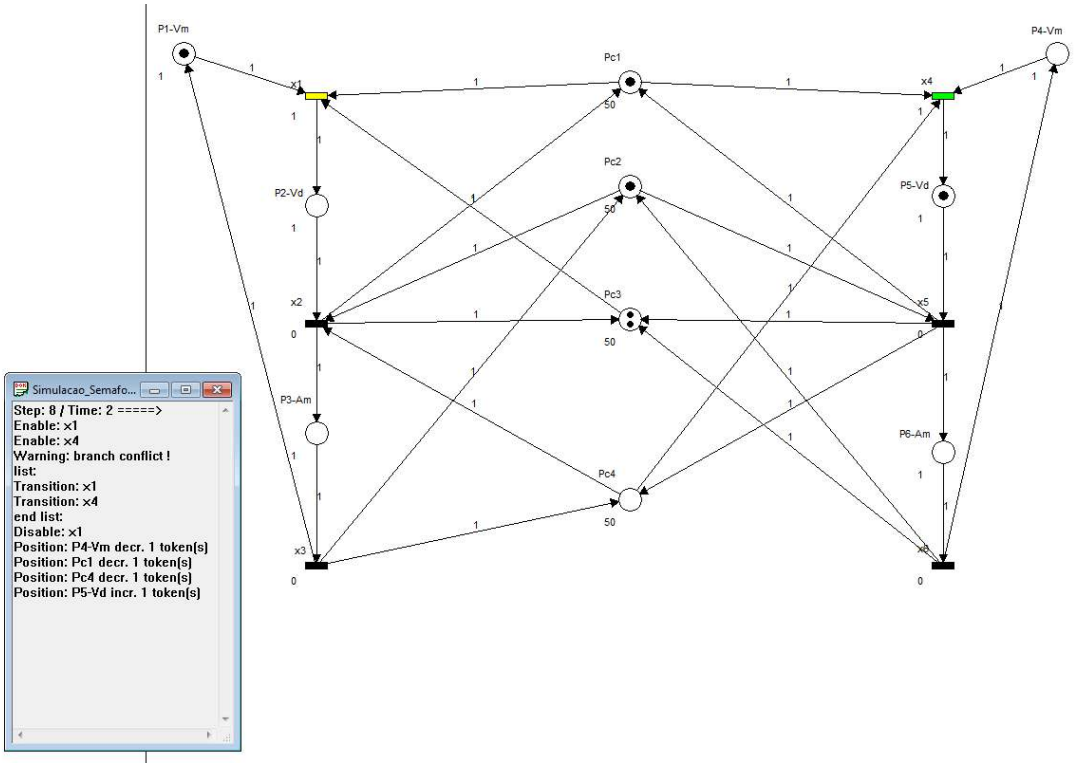
Fonte: Autores (2023).

Figura 19 – Simulação: Mudança para que S2 alterne as cores dos semáforos



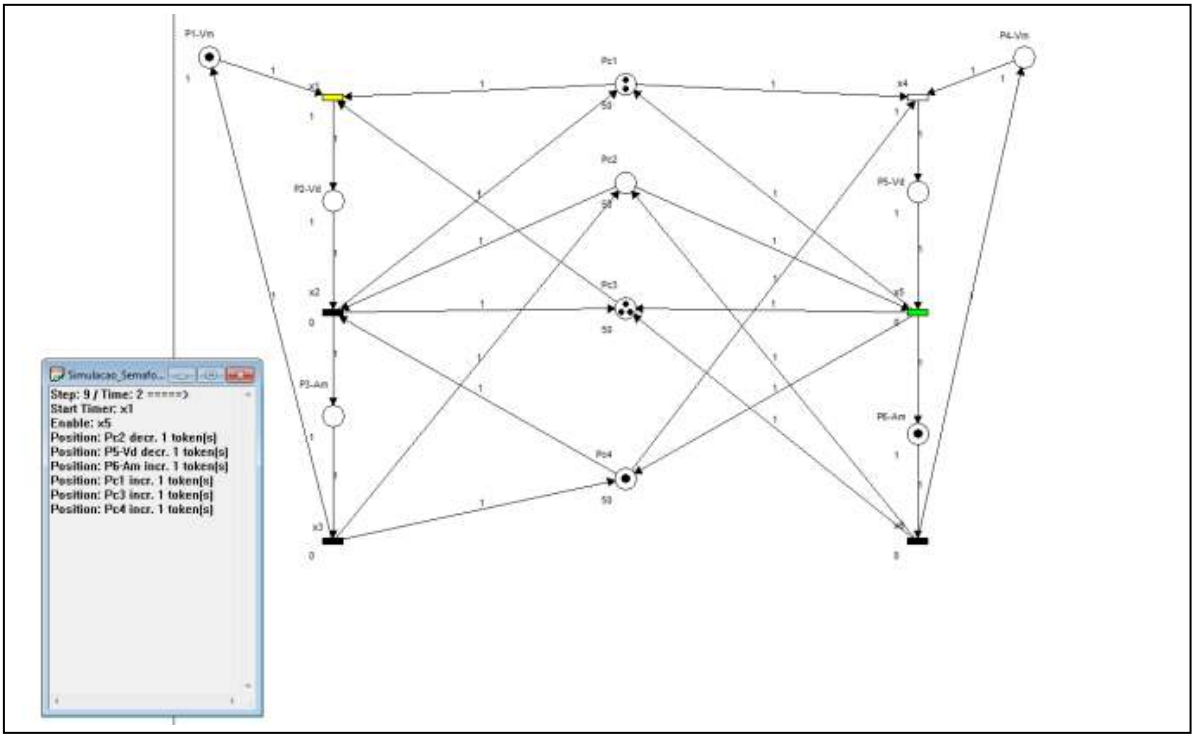
Fonte: Autores (2023).

Figura 20 – Simulação: S1 vermelho e S2 verde



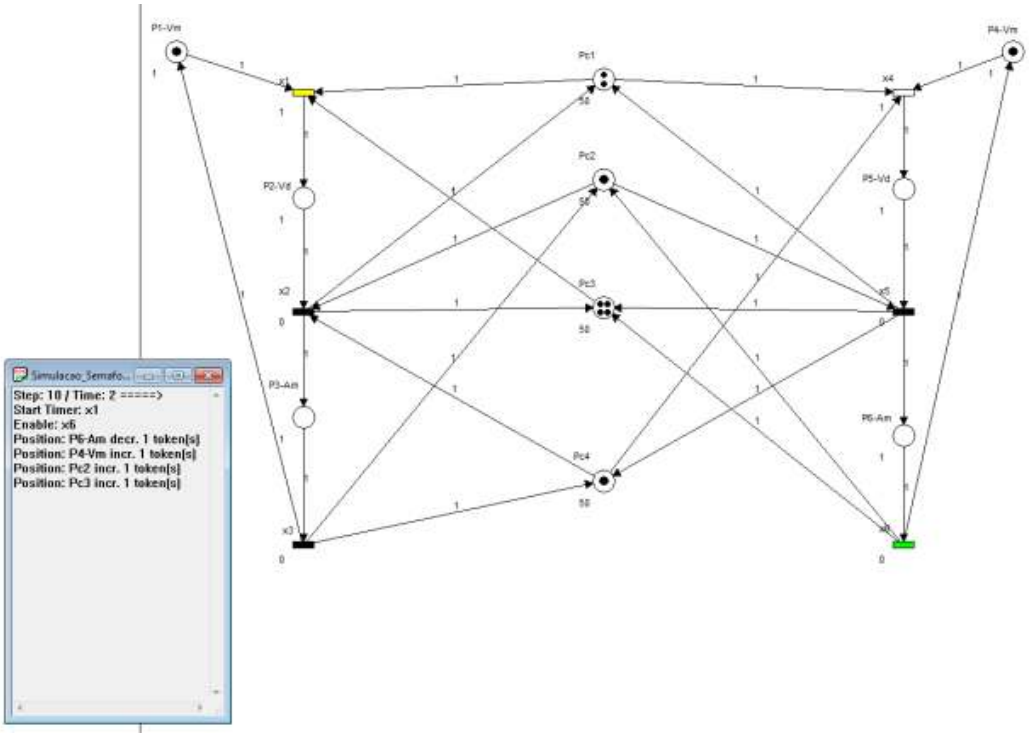
Fonte: Autores (2023).

Figura 21 – Simulação: S1 vermelho e S2 amarelo



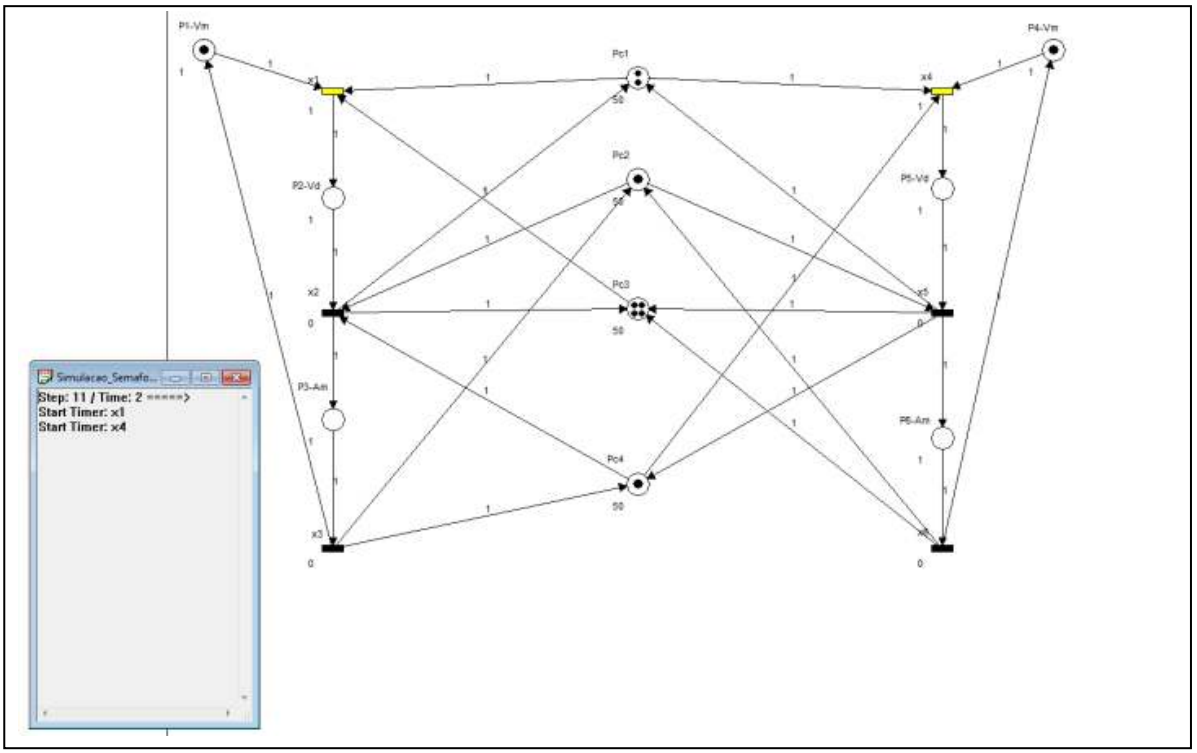
Fonte: Autores (2023).

Figura 22 – Simulação: S1 vermelho e S2 vermelho rápido para transição



Fonte: Autores (2023).

Figura 23 – Simulação: Mudança para reiniciar o sistema com S1 alternando cores



Fonte: Autores (2023).

5. Conclusão

Diante dos estudos apresentados, pode-se considerar que o presente artigo pode ser utilizado como um tutorial por outros pesquisadores interessados no assunto, onde foi explicitado com clareza os procedimentos necessários para elaborar o controle de semáforos, utilizando RP, atingindo assim ao objetivo geral proposto.

Os objetivos específicos também foram alcançados, visto que foram expostos os conceitos sobre a técnica escolhida, além da situação a ser considerada na modelagem e no controle ter sido detalhada. Por fim, simulou-se em ambiente computacional os resultados obtidos, evidenciando que por meio do controle através da invariância de lugar os semáforos do cruzamento podem ser sincronizados.

Dessa forma, a hipótese levantada foi confirmada. Por meio das pesquisas apresentadas conclui-se que tal técnica satisfaz a meta de se criar um controlador capaz de sincronizar os sinais de trânsito. Os cálculos foram feitos segundo o método escolhido, foi gerado um controlador e este foi integrado a RP dos semáforos.

A simulação evidenciou que o controlador é efetivo, apesar de não ser completo. Uma sugestão de continuidade desse estudo é simular o controle feito em outros ambientes computacionais, os quais podem ser incluídos os tempos de acionamento de cada lâmpada. Também pode-se ampliar o número de semáforos, criando sincronicidade em diferentes semáforos de vias, promovendo uma melhor mobilidade urbana.

Referências

ABREU, V. A. **Projeto de automação industrial têxtil baseado em Rede de Petri.** 2022. 67f. Monografia (Engenharia de Controle e Automação) – Universidade Federal de Ouro Preto. Ouro Preto, 2022. Disponível em: https://www.monografias.ufop.br/handle/35400000/4337. Acesso em: 17 fev. 2023.

ANDRADE, C. R. **Controle de sistemas Max-plus lineares sujeitos a restrições no estado:** aplicação a sistemas de transporte. 2008. 77f. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Minas Gerais. Belo Horizonte, 2008. Disponível em: http://www.bibliotecadigital.ufmg.br/dspace/bitstream/handle/

1843/BUOS-8CREUE/c_ntia_ ribeiro_andrade.pdf?sequence=1 Acesso em: 23 jan. 2023.

HPSIM. **Simulador.** 2023. Disponível em: http://www.winpesim.de. Acesso em: 02 fev. 2023.

OLIVEIRA, G. N. **Estudo da aplicação de Rede de Petri em um e-commerce**. 2022. 63f. Monografia (Engenharia Eletrônica e de Telecomunicações) – Universidade Federal de Uberlândia. Patos de Minas, 2022. Disponível em: https://repositorio.ufu.br/bitstream/123456789/35875/1/EstudoAplica%C3%A7%C3%A3oRede.pdf. Acesso em: 16 fev. 2023.

PONTES, B. S. *et al.* Análise da percepção da mobilidade urbana pelos usuários na cidade de Manaus. **Research, Society and Development,** v. 11, n. 17, p. 1-13, 2022.

REIS, J. F.; SILVA, G. G. Modelagem e controle de semáforos utilizando redes de petri. **Multices.** v. 4, n. 1, p. 182-195, 2016.

SILVA, G. G., **Notas das aulas de Tópicos III.** UFSJ: Ouro Branco, 2014.