

TEORIA DE CONTROLE SUPERVISÓRIO E CONTROLE DIRIGIDO ÓTIMO: SOLUÇÃO DE UM PROBLEMA DE SEQUENCIAMENTO

RAMON G. DURÃES*, GUSTAVO L. VIEIRA*, PATRÍCIA N. PENNA*, BRUNO V. ADORNO†

**Laboratório de Análise e Controle de Sistemas a Eventos Discretos (LACSED)*
Universidade Federal de Minas Gerais
Av. Antônio Carlos, 6627, Pampulha - Belo Horizonte - MG, 31270-901

†*Departamento de Engenharia Elétrica (DEE), Universidade Federal de Minas Gerais*
Av. Antônio Carlos, 6627, Pampulha - Belo Horizonte - MG, 31270-901

Emails: ramonduraes@ufmg.br, gustavolvieira@ufmg.br, ppena@ufmg.br, adorno@ufmg.br

Abstract— This paper presents a new variation of the tower of Hanoi puzzle and, for this problem, develops a method of automatic modelling and optimal solving. The method utilizes the Supervisory Control Theory of discrete event systems combined with the Optimal Directed Control. A case study validates the presented method.

Keywords— Discrete event systems, Supervisory Control Theory, Optimal Directed Control, sequencing.

Resumo— Esse trabalho propõe uma nova variação do problema das torres de Hanói e, para ele, desenvolve um método automático de modelagem e solução ótima. Para isso, utiliza-se a Teoria de Controle Supervisório (TCS) de sistemas a eventos discretos, aliada ao Controle Dirigido Ótimo. Um estudo de caso valida o método apresentado.

Palavras-chave— Sistemas a eventos discretos, Teoria de Controle Supervisório, Controle Dirigido Ótimo, sequenciamento.

1 Introdução

Um modelo é uma representação teórica da realidade que nos permite solucionar problemas utilizando técnicas de análise e controle amplamente estudadas. Um Sistema a Eventos Discretos (SED) pode ser modelado por um autômato que relaciona os possíveis estados do sistema uns aos outros, por meio de transições causadas por eventos.

A precisão de um modelo também afeta diretamente os resultados das técnicas de análise e controle aplicadas sobre ele. Em um autômato é comum a utilização do controle supervisório monolítico, proposto por Ramadge & Wonham (1989), que é baseado no projeto de supervisores que limitam o comportamento da planta sob seu controle desabilitando um conjunto de eventos controláveis em cada estado.

Mas nem sempre a não violação das especificações é suficiente. Uma ação tomada pode afetar significativamente o desempenho do sistema, para melhor ou pior. Portanto é necessário que, dentre o conjunto de eventos factíveis fornecidos pelo supervisor, o melhor deles seja escolhido. Para dar suporte a essa decisão, pode-se utilizar o controle dirigido (Huang & Kumar, 2005), posteriormente expandido para controle dirigido ótimo por Huang & Kumar (2008), que limita o supervisor ao permitir no máximo um evento controlável por estado, minimizando uma função de custo. Este método é particularmente interessante por fornecer uma solução em tempo polinomial em relação ao número de estados do supervisor monolítico,

que minimiza o número de eventos executados até que se alcance o estado marcado.

Essa base teórica é suficiente para resolver diversos problemas em SEDs: projeta-se o modelo, obtém-se seu supervisor e aplicam-se métodos de suporte à escolha da melhor sequência de eventos quando necessário. Porém, há casos nos quais uma simples mudança na condição inicial do sistema, ou no seu objetivo final, gera uma solução bastante diferente da anterior. Além disso, essas ligeiras alterações requerem que os modelos já construídos sejam alterados, processo que em geral leva bastante tempo.

A inviabilidade da realização manual desse meticuloso processo de projeto e solução do problema de sequenciamento motivou este trabalho, que visa tratar de um problema de sequenciamento baseado nas torres de Hanói.

O clássico problema das torres de Hanói, em sua versão tradicional, consiste em mover uma pilha de discos de um pino para outro, um disco por vez, sem colocar um disco maior sobre um menor. Sua primeira referência conhecida é Lucas (1893) e ele ainda hoje é amplamente estudado tanto em sua versão original (Havur et al., 2013; Ahrabian et al., 2011), quanto em versões modificadas (Stockmeyer & Lunnon, 2010).

1.1 Contribuições

Este artigo propõe uma nova variação do problema das torres de Hanói: dadas pilhas de blocos coloridos, a tarefa é reorganizá-los para montar uma nova pilha com blocos numa ordem de co-

res determinada. Em geral há múltiplas soluções para um dado problema. A cada uma é associado como custo o número de movimentos realizados para atingir o estado final desejado. Naturalmente, é desejável encontrar no conjunto de soluções aquela com menor custo.

Para solucionar o problema, desenvolveu-se um método de modelagem automática que requer como entrada apenas a condição inicial do sistema, da qual todos os parâmetros necessários são extraídos. O método proposto gera automaticamente autômatos que modelam a planta e uma especificação que faz com que a sequência desejada seja obedecida. Compõe-se o modelo da planta com a especificação e calcula-se o supervisor para o sistema. De posse do supervisor, utiliza-se o controle dirigido ótimo para obter a sequência ótima de movimentos que soluciona o problema com sucesso e custo mínimo.

1.2 Organização do Artigo

O artigo está organizado da seguinte maneira: na seção 2 são apresentados os conceitos básicos relacionados a autômatos, controle supervisorio e controle dirigido ótimo. A seção 3 apresenta uma descrição detalhada do problema de sequenciamento de pilhas. A seção 4 é dedicada ao modelo desenvolvido e à sua construção e resolução algorítmica. A seção 5 resolve um problema exemplo, passo a passo, utilizando a metodologia proposta. A seção 6 apresenta as conclusões.

2 Noções Preliminares

Neste trabalho, aliada à Teoria de Controle Supervisorio de SED (TCS), propõe-se a utilização da teoria de Controle Dirigido Ótimo. Ambas teorias são desenvolvidas sobre o formalismo de linguagens e autômatos.

2.1 Linguagens e Autômatos

Seja Σ um conjunto finito e não vazio de símbolos, uma cadeia s é formada por uma sequência de eventos de Σ . A cadeia ε é composta por nenhum evento e é chamada de cadeia vazia. Σ^* é o conjunto de todas as cadeias possíveis de serem formadas com os símbolos de Σ e uma linguagem L sobre um conjunto de eventos Σ é um subconjunto de Σ^* .

Sejam s , t e u cadeias, t é prefixo de s se existe u tal que $s = tu$. A cadeia tu denota a concatenação de t e u . Seja uma linguagem $L \in \Sigma^*$, então o prefixo-fechamento de L , denotado por \bar{L} , é definido por $\bar{L} := \{s \in \Sigma^* | \exists t \in \Sigma^*, st \in L\}$. Por exemplo, seja $L = \{\alpha, \beta\gamma\sigma\}$ uma linguagem definida sobre o alfabeto $\Sigma = \{\alpha, \beta, \gamma, \sigma\}$, o prefixo-fechamento de L é dado por $\bar{L} = \{\varepsilon, \alpha, \beta, \beta\gamma, \beta\gamma\sigma\}$.

Um SED pode ser modelado por um autômato determinístico $G = (Q, \Sigma, \hat{\delta}, q_o, F)$, em que Q é o conjunto de estados, Σ é o conjunto de eventos, $\hat{\delta} : Q \times \Sigma \rightarrow Q$ é a função de transição, $q_o \in Q$ é o estado inicial e $F \subseteq Q$ é o conjunto de estados marcados.

A função $\hat{\delta}$ pode ser estendida para cadeias de eventos como a função $\delta : Q \times \Sigma^* \rightarrow Q$ tal que $\delta(q, \varepsilon) = q$; e $\delta(q, s\sigma) = \hat{\delta}(\delta(q, s), \sigma)$, sendo $q \in Q$, $s \in \Sigma^*$ e $\sigma \in \Sigma$. Em palavras, a função δ define para qual estado o sistema transita a partir de um estado de Q , dada a ocorrência da cadeia $s \in \Sigma^*$. Dessa forma, redefine-se o autômato determinístico como $G = (Q, \Sigma, \delta, q_o, F)$.

A linguagem gerada por um autômato G é definida como o conjunto de cadeias para as quais a função de transição estendida δ está definida a partir do estado inicial, ou $\mathcal{L}(G) = \{s \in \Sigma^* | \delta(q_o, s)!\}$, em que $!$ indica “está definido”. A linguagem marcada desse autômato é um subconjunto de $\mathcal{L}(G)$ que leva o autômato do estado inicial a um estado marcado, $\mathcal{L}_m(G) = \{s \in \Sigma^* | \delta(q_o, s) \in F\}$.

Um estado $q \in Q$ é dito acessível se $\exists s \in \Sigma^*$ tal que $\delta(q_o, s) = q$ e dito coacessível se $\exists s \in \Sigma^*$ tal que $\delta(q, s) \in F$. Um autômato G é dito acessível se todos os seus estados $q \in Q$ são acessíveis e coacessível se todos os estados forem coacessíveis. A parte acessível de G é denotada por $Ac(G)$ e a parte coacessível, por $CoAc(G)$.

Caso todos os estados de um autômato sejam acessíveis e coacessíveis, o autômato é dito aparado.

2.2 Composição Síncrona de Autômatos

Sejam dois autômatos $G_1 = (Q_1, \Sigma_1, \delta_1, q_{o1}, F_1)$ e $G_2 = (Q_2, \Sigma_2, \delta_2, q_{o2}, F_2)$, a composição síncrona desses dois autômatos é definida como $G = G_1 || G_2 = Ac(Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, \delta, (q_{o1}, q_{o2}), F_1 \times F_2)$, onde:

$$\delta((q_1, q_2), \sigma) = \begin{cases} (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma)) & \text{se } \delta_1(q_1, \sigma)! \text{ e } \delta_2(q_2, \sigma)! \\ (\delta_1(q_1, \sigma), q_2) & \text{se } \delta_1(q_1, \sigma)! \text{ e } \sigma \notin \Sigma_2 \\ (q_1, \delta_2(q_2, \sigma)) & \text{se } \delta_2(q_2, \sigma)! \text{ e } \sigma \notin \Sigma_1 \\ \text{indefinida} & \text{senão} \end{cases}$$

Um estado só é marcado em G caso os estados correspondentes em G_1 e G_2 o sejam.

2.3 Teoria de Controle Supervisorio

Seja $G = (Q, \Sigma, \delta, q_o, F)$ o autômato que modela uma planta. Particiona-se Σ em $\Sigma = \Sigma_u \cup \Sigma_c$, sendo Σ_c o conjunto de eventos controláveis, que podem ser inibidos de ocorrer no sistema e Σ_u o conjunto de eventos não controláveis, que não podem ser inibidos de ocorrer no sistema.

Sobre Σ podem ser definidos subconjuntos de eventos γ , tais que se $\sigma \in \gamma$ então ele é permitido por γ ; senão, σ é desabilitado por γ . Esses subconjuntos são chamados de entradas de controle. Dessa forma, define-se o conjunto das entradas de controle Γ como $\Gamma = \{\gamma \in 2^\Sigma \mid \gamma \supseteq \Sigma_u\}$, onde 2^Σ denota o conjunto de todos os possíveis conjuntos de eventos do alfabeto Σ , e a condição $\gamma \supseteq \Sigma_u$ indica simplesmente que os eventos não controláveis não podem ser desabilitados. Por exemplo, sejam $\Sigma_c = \{\alpha, \beta\}$ e $\Sigma_u = \{\mu\}$, o conjunto de entradas de controle válidas é então $\Gamma = \{\{\mu\}, \{\mu, \alpha\}, \{\mu, \beta\}, \{\mu, \alpha, \beta\}\}$.

Um supervisor para G é uma função $V : \mathcal{L}(G) \rightarrow \Gamma$ que associa a cada cadeia possível $s \in \mathcal{L}(G)$ uma entrada de controle $\gamma = V(s) \in \Gamma$. O par (G, V) pode ser escrito como V/G para denotar G sob supervisão de V . O comportamento em malha fechada de V/G é definido como a linguagem $\mathcal{L}(V/G) \subseteq \mathcal{L}(G)$.

A linguagem marcada de V/G é $\mathcal{L}_m(V/G) = \mathcal{L}(V/G) \cap \mathcal{L}_m(G)$ e portanto consiste das cadeias marcadas de $\mathcal{L}_m(G)$ que sobrevivem à supervisão de V .

Seja K uma linguagem tal que $K \subseteq \Sigma^*$, e \bar{K} o prefixo-fechamento dessa linguagem. A linguagem K será controlável em relação a G se $\bar{K}\Sigma_u \cap \mathcal{L}(G) \subseteq \bar{K}$, ou seja, se e somente se nenhuma cadeia de $\mathcal{L}(G)$ que esteja no prefixo de K , quando seguida de um evento não controlável em G , deixa de ser prefixo em K .

A condição de controlabilidade de uma linguagem K garante que haverá controle supervisor não-bloqueante V para G tal que $\mathcal{L}_m(V/G) = K$. Este supervisor é classificado como supervisor marcador e não apenas controla a planta mas também possui ação de marcação.

Caso K não seja controlável, existe uma melhor e única aproximação de K chamada de máxima sublinguagem controlável contida em K em relação à planta G , denotada por $\text{SupC}(K, G)$. Neste trabalho, a entidade “supervisor” é representada por S , sendo $S = \text{SupC}(K, G)$ de forma que $s\sigma \in S \iff \sigma \in V(s)$. Com o desenvolvimento de extensões para a Teoria de Controle Supervisorio, o supervisor clássico passa a ser chamado de supervisor monolítico. No contexto desse trabalho, todos os eventos são controláveis, ou seja, $\Sigma = \Sigma_c$. Logo a linguagem K sempre será controlável em relação à planta G , e $S = \text{SupC}(K, G) = K$.

O supervisor S obtido será minimamente restritivo, ou seja, desabilita apenas eventos controláveis que podem levar o sistema a estados bloqueantes ou que violam as especificações. Assim, uma planta sob controle supervisorio representa todos os caminhos possíveis legais para o sistema.

Seja o sistema a ser controlado composto de subplantas G_i , com $i = 1, \dots, n$. Pela abordagem monolítica da teoria de controle supervisorio (Ramadge & Wonham, 1989), a planta G a ser contro-

lada deve ser única e pode ser obtida pela composição das subplantas G_i . Essa planta deve ser restringida por uma ou mais especificações E_j , com $j = 1, \dots, m$. De forma análoga, a especificação E deve ser única e pode ser obtida pela composição das especificações E_j para o cálculo do supervisor monolítico.

2.4 Controle Dirigido Ótimo

O método do controle supervisorio apresentado permite determinar todas as ações permitidas em um estado, mas isso não é o suficiente para solucionar o problema de sequenciamento em um Sistema a Eventos Discretos. É necessário determinar, dentre todas as opções, qual deve ser o comando de controle em cada estado.

A teoria do Controle Dirigido (Huang & Kumar, 2005) surge para refinar o Controle Supervisorio. Um controlador, chamado diretor, habilita no máximo um evento controlável para cada estado de uma planta (sem desabilitar eventos não-controláveis). Desse modo, uma planta sob controle dirigido já tem determinada, para cada estado, no máximo uma ação de controle.

Uma mesma planta pode possuir diversos diretores diferentes. Dentre eles, é desejável buscar aquele que leve a planta ao resultado desejado de maneira ótima, chamado Diretor Ótimo. O critério de otimalidade pode variar de acordo com cada problema, mas em geral envolve a minimização do gasto de algum recurso do sistema.

Huang & Kumar (2008) desenvolveram um método para obtenção do diretor ótimo em um autômato qualquer, com complexidade polinomial em relação ao número de estados. Esse método baseia-se na atribuição de um custo para cada evento, representando a quantidade de recursos dispendida com sua ocorrência.

O algoritmo utilizado possui dois elementos principais: busca reversa a partir dos estados marcados terminais e refinamento sucessivo do espaço de busca. A lógica é explicada superficialmente abaixo, mas para entender passo a passo o método do controle dirigido recomenda-se consultar o trabalho original (Huang & Kumar, 2008).

A busca reversa é iniciada no supervisor monolítico a partir do conjunto de estados marcados terminais do sistema. Utiliza-se uma lógica gulosa para determinar o estado que, com um único evento, alcança esse conjunto com o menor custo. Para esse estado é então selecionada no máximo uma única ação de controle, desabilitando todos os outros eventos controláveis nesse estado. Ele é então adicionado ao conjunto considerado e o processo se repete até que o estado inicial do sistema seja alcançado. Nesse ponto, o autômato é aparado e o resultado obtido é um diretor.

Posteriormente, é realizado um refinamento do espaço de estados. Dentre os estados marca-

dos não terminais e o estado inicial, elimina-se do autômato aquele com maior custo. No novo autômato, a busca reversa é aplicada novamente. O novo diretor obtido é comparado com o anterior e aquele com menor custo é preservado. O processo de refinamento se repete até que a planta resultante seja vazia. Nesse ponto, o diretor obtido é garantidamente ótimo.

3 Descrição do Problema

A Torre de Hanói (Lucas, 1893) é um problema muito conhecido há gerações, tanto por cientistas da computação quanto por roboticistas (Havur et al., 2013). Em robótica, por exemplo, este problema é utilizado para estudo de algoritmos recursivos, reconhecimento de objetos, planejamento de tarefas, planejamento de movimentos, entre outros. O problema consiste em mover uma pilha de discos com o menor número de movimentos de um pino a outro. Deve-se obedecer às seguintes regras:

- Só um disco deve ser movido por vez;
- Um disco maior não pode ser colocado sobre um menor.

A Figura 1 ilustra uma situação típica encontrada no problema das torres de Hanói.

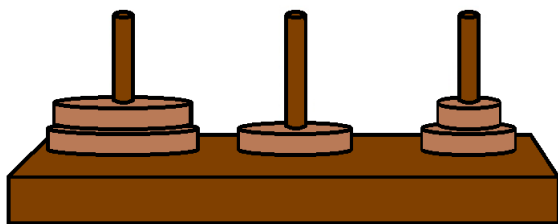


Figura 1: Torres de Hanói

Este artigo propõe uma variação do problema original: dadas diversas pilhas de blocos, com diferentes cores, deseja-se montar uma nova pilha com um número mínimo de movimentos. A sequência das cores de blocos desejada na nova pilha (pilha resultado $p0$) é uma entrada do sistema definida pelo usuário.

As seguintes regras devem ser seguidas:

- Apenas um bloco pode ser movido por vez;
- Os blocos retirados do topo de uma pilha só podem ser colocados no topo de outras pilhas já montadas, ou na pilha que se deseja montar.

A solução desse problema depende da configuração inicial das pilhas, que é outra entrada do sistema. A Figura 2 mostra um exemplo de situação inicial.

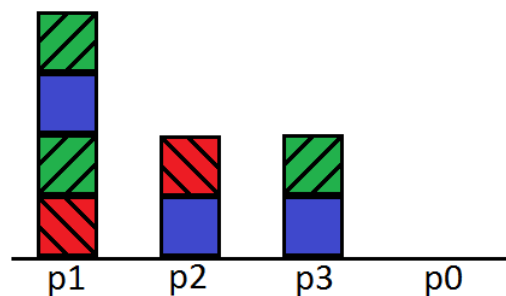


Figura 2: Exemplo de configuração inicial do sistema. Os blocos com linhas diagonais para a direita (\swarrow) representam a cor verde; os com linhas diagonais para a esquerda (\searrow) são vermelhos; e os que não possuem linhas são azuis.

4 Metodologia: Modelagem e Solucionamento

Para lidar com o problema, definem-se os parâmetros P para o número de pilhas (não inclui a pilha resultado $p0$), B para o número de blocos e C para o número de cores diferentes no sistema. Define-se ainda o tamanho T para as pilhas que não pode ser excedido. Dado que as pilhas têm tamanho fixo e os blocos só podem ser movidos para outras pilhas, este tamanho é definido em função do número de blocos e de pilhas na condição inicial do sistema de forma a garantir que todos os blocos possam ser acessados. Caso contrário, todas as pilhas poderiam começar cheias e não seria possível mover nenhum bloco. O parâmetro T será referido como “tamanho mínimo” neste trabalho apesar de representar também o tamanho máximo permitido para as pilhas.

Este trabalho impõe uma nova restrição ao problema: uma vez na pilha $p0$, o bloco não pode mais ser retirado, ou seja, ela não pode ser utilizada temporariamente para reordenar blocos.

Propõe-se solucionar o problema descrito utilizando a teoria de controle supervisão de SEDs, aliada ao controle dirigido ótimo.

4.1 Pilhas

A cada pilha i é associado um autômato $G_{pi} = (Q_{pi}, \Sigma_{pi}, \delta_{pi}, q_{opi}, F_{pi})$ que mostra todas as configurações possíveis para a pilha, ou seja, todas as possíveis combinações de cores em cada posição. Cada autômato possui um estado central que indica que a pilha está vazia. A partir desse estado, definem-se “níveis de estados” que indicam o tamanho atual das pilhas. Caso o estado não pertença ao último nível, ele está ligado a C outros estados de um nível superior, cada um associado a uma cor do sistema.

Neste trabalho, são utilizadas as letras “R”, “G” e “B”, para indicar as cores vermelho, verde e azul, respectivamente, para representar tanto es-

tados quanto eventos dos autômatos. Essas letras são maiúsculas quando se referem a estados e minúsculas para eventos. Nas ilustrações, os blocos com linhas diagonais para a direita (\swarrow) representam a cor verde; aqueles com linhas diagonais para a esquerda (\searrow) são vermelhos; e os blocos que não possuem linhas são azuis.

A Figura 3 mostra uma pilha $p1$ e o autômato associado G_{p1} . Nela, o estado 0 é o estado central. Como há duas cores de blocos no sistema, o estado 0 está ligado a dois outros estados: $G1$ e $R1$. Esses estados fazem parte do nível 1 e indicam que a pilha possui 1 bloco verde ou vermelho, respectivamente. Por não serem do último nível, também estão ligados a dois outros estados e esse raciocínio se repetiria caso houvesse mais níveis.

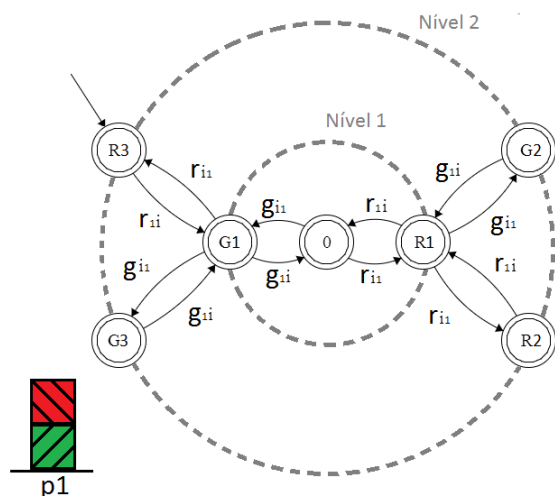


Figura 3: Exemplo de modelagem de uma pilha com $T = 2$.

Os eventos desses autômatos têm a forma x_{ij} , o que significa “mover o bloco de cor x da pilha i para a pilha j ”. Foi definido que $i \neq j$, caso contrário o modelo incluiria a situação em que um bloco é retirado e adicionado à mesma pilha, o que apenas aumentaria o modelo inutilmente.

Caso um bloco seja adicionado à pilha em questão, o estado atual se afasta do central (vai para o nível imediatamente superior), pela execução do evento que depende da cor do bloco. Caso um bloco seja retirado da pilha, o estado atual se aproxima do central (muda para o nível imediatamente inferior).

O estado inicial fornece toda a informação necessária acerca de uma determinada pilha. Ele é definido observando o tamanho atual e a sequência de cores dos blocos na pilha. Seja t o tamanho atual da pilha, o estado inicial pertencerá ao nível t do autômato. A sequência de cores dos blocos define a sequência de eventos pela qual este estado é alcançado a partir do estado central: o primeiro evento a partir do estado central indica a cor do primeiro bloco empilhado e assim sucessivamente.

No exemplo da Figura 3, o estado inicial é $R3$. Ele pertence ao nível 2 por haver 2 blocos na pilha. A partir do estado central 0, ele é alcançado pela sequência $g_{i1}r_{i1}$, de modo que g_{i1} e r_{i1} significam “mover um bloco verde ou vermelho, respectivamente, de uma pilha qualquer i para a pilha 1”. A sequência *verde*>*vermelho* refere-se à ordem de empilhamento da pilha $p1$. Caso a pilha $p1$ estivesse na ordem contrária (*vermelho*>*verde*), o estado inicial seria $G2$.

Denota-se $|Q_{G_{pi}}|$ o número de estados de um autômato G_{pi} . Cada nível desse autômato possui C^t estados, em que C é o número de cores diferentes no sistema e t é o nível atual, que é igual ao tamanho atual da pilha. Dessa forma, o modelo de cada pilha possui $|Q_{G_{pi}}| = \sum_{t=0}^T C^t$ estados, em que T é o tamanho mínimo definido para as pilhas no sistema. Cada termo deste somatório indica um nível, sendo o nível $t = 0$ relacionado ao estado central (pilha vazia).

Seja $|\rightarrow_{G_{pi}}|$ o número de transições associadas a esse autômato. Calcula-se $|\rightarrow_{G_{pi}}|$ por meio da expressão

$$|\rightarrow_{G_{pi}}| = \left[\sum_{t=0}^{(T-1)} (C^t \times 2 \times C) \right] \times [P + (P - 1)]/2$$

$$= (2P - 1) \times \sum_{t=1}^T C^t.$$

O somatório indica que percorre-se o autômato desde o nível 0 (tamanho $t = 0$) até o penúltimo nível (tamanho $t = T - 1$) somando o número de transições associada a cada um deles. Assim, multiplica-se o número de estados no nível atual (C^t) pelo número de estados aos quais eles estão ligados (C). Essas ligações são indicadas por meio de duas arestas orientadas (ou setas), por isso multiplica-se o termo do somatório por 2. O segundo termo indica que metade dessas setas estão associadas a $P - 1$ eventos: retirar um bloco das outras $P - 1$ pilhas e colocar na atual, ficando a outra metade associada a P eventos: retirar um bloco da pilha modelada e colocá-lo em uma das outras $P - 1$ pilhas, acrescidas da pilha resultado $p0$.

O autômato $G_{p1} = (Q_{p1}, \Sigma, \delta_{p1}, q_{op1}, F_{p1})$ da Figura 3 possui $|Q_{G_{p1}}| = \sum_{t=0}^2 2^t = 7$ estados. O número de transições $|\rightarrow_{G_{p1}}|$ depende do número de pilhas no sistema. Supondo que a pilha $p1$ esteja num sistema com 4 pilhas ($P = 4$), calcula-se o número de transições por $|\rightarrow_{G_{p1}}| = \left[\sum_{t=0}^1 (2^t \times 2 \times 2) \right] \times [4 + (4 - 1)]/2 = 42$.

Todos os estados dos autômatos G_{pi} são marcados pois deseja-se que apenas a especificação E determine qual estado de K será considerado uma tarefa completa.

A Figura 4 mostra um segundo exemplo de modelo de pilha. As setas cinza indicam como diferentes parâmetros afetam o tamanho do autô-

mato: cresce radialmente com o aumento do tamanho das pilhas (T), e aumenta um “ramo” para cada cor de bloco diferente (C) no sistema. Neste caso, $T = 2$, $C = 3$ e o estado inicial ($R3$) revela que o autômato está modelando a pilha $p2$ da Figura 2.

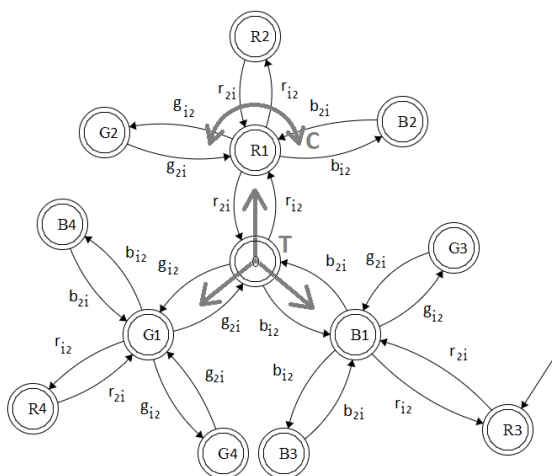


Figura 4: Autômato G_{p2} : modelo associado à pilha $p2$ da Figura 2.

4.2 Planta

Ao modelar o comportamento global da planta, deve-se garantir que qualquer bloco do sistema possa ser alcançado. Para isso, a condição $B \leq T(P - 1) + 1$ deve ser satisfeita. Dessa expressão, calcula-se o tamanho mínimo T das pilhas: $T \geq (B - 1)/(P - 1)$. Define-se que todos os autômatos de pilhas do sistema devem permitir que elas possuam T blocos, mesmo que na situação inicial isso não seja verdade. Expande-se portanto o número de estados ($|Q|$) das pilhas, mantendo o estado inicial q_0 .

O sistema será modelado pelo autômato $G = \parallel_{i=1}^P G_{pi}$, sendo G_{pi} os modelos associados a cada pilha.

O número de eventos em G , denotado por $|\Sigma|$, pode ser calculado por

$$\begin{aligned} |\Sigma| &= (C \times P) + (C \times A_{P,2}) \\ &= (C \times P) + (C \times \frac{P!}{(P-2)!}). \end{aligned}$$

O primeiro termo, $(C \times P)$, é o número de eventos que levam um bloco de cor e pilha quaisquer à pilha $p0$. O termo $(C \times A_{P,2})$ é o número de eventos que levam o bloco de uma pilha à outra, exceto à pilha resultado.

4.3 Especificação

A especificação do modelo indica a sequência de cores de blocos a ser obtida na pilha resultado $p0$.

Sejam B_{p0} o número de blocos na pilha $p0$ e $|Q_E|$ o número de estados da especificação, tem-se a relação $|Q_E| = B_{p0} + 1$. Às transições entre estados são associados os eventos das cores desejadas na pilha resultado. A ordem é intuitiva: a partir do estado inicial, o primeiro evento indica o primeiro bloco a ser depositado em $p0$. O estado final é o único marcado e nenhum outro evento é permitido, uma vez que o resultado desejado já foi obtido.

Com exceção do último, cada estado possui um auto-laço permitindo todos os eventos do sistema, exceto aqueles relacionados à pilha desejada. Isso indica que o modelo permite que os blocos sejam movidos de pilha em pilha para que outros sejam alcançados. A ausência desse auto-laço no estado marcado impede os movimentos desnecessários de blocos em outras pilhas.

A Figura 5 mostra um exemplo de especificação. Nele, o nome dos estados indica o número de blocos na pilha $p0$ e a sequência da pilha a ser montada é *azul > vermelho > vermelho > verde*, representada pelos conjuntos de eventos b_{i0} , r_{i0} , r_{i0} , g_{i0} , respectivamente, com $i \in \{1, \dots, P\}$.

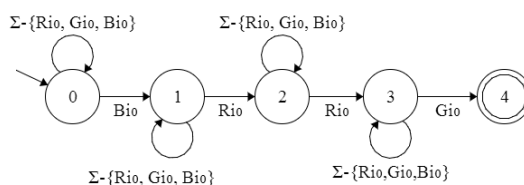


Figura 5: Exemplo de especificação.

4.4 Obtenção da Solução Ótima

Após a síntese do supervisor, basta submetê-lo ao algoritmo do diretor ótimo proposto por Huang & Kumar (2008). O diretor obtido como resultado representa uma única cadeia s_o , que é garantidamente ótima no que se refere a alcançar o estado marcado com o mínimo de movimentos possível.

Sabe-se que o método de busca do diretor ótimo não lida apropriadamente com processos nos quais há compartilhamento de recursos, conforme demonstrado por Vieira & Pena (2013). No entanto, no problema há apenas um agente que realiza uma ação de cada vez. Logo o método de controle dirigido é adequado para este problema.

4.5 Resumo dos Procedimentos

Sintetiza-se nessa seção a metodologia proposta como segue:

1. obter o número de pilhas (P), o número de cores (C) e o número de blocos (B) a partir da condição inicial das pilhas;
2. calcular o tamanho mínimo (T) das pilhas, garantindo que $B \leq T(P - 1) + 1$;

- para cada pilha i , criar um autômato G_{pi} conforme descrito na seção 4.1;
- definir os estados iniciais dos modelos de cada uma das pilhas observando a sequência de blocos, a partir da configuração inicial;
- obter G por meio da composição dos autômatos G_{pi} ;
- gerar a especificação E de acordo com a ordem desejada da pilha resultado $p0$;
- obter $K = E || \mathcal{L}_m(G)$ e o supervisor $S = \text{SupC}(K, G)$ implementará a linguagem K uma vez que todos os eventos são controláveis;
- submeter o autômato que reconheça K ao algoritmo do diretor ótimo. A saída será um autômato que implementa uma única cadeia s_o , que é garantidamente ótima com relação ao número de movimentos efetuados.
- O autômato que modela o comportamento global da planta, G , é obtido fazendo $G = G_{p1} || G_{p2} || G_{p3}$. G possui $|Q| = 29.564$, $|\Sigma| = 27$ e $|\rightarrow| = 202.254$.
- Como o objetivo é montar uma pilha na ordem *azul > vermelho > vermelho > verde*, a especificação E é semelhante à da Figura 5. Obtém-se um autômato com $|Q_E| = 5$, $|\Sigma_E| = 27$ e $|\rightarrow_E| = 84$.
- Calcula-se o supervisor $S = \text{SupC}(K, G) = E || \mathcal{L}_m(G)$. O supervisor calculado possui $|Q_S| = 13.590$ estados, $|\Sigma_S| = 27$ eventos e $|\rightarrow_S| = 68.232$ transições.
- O supervisor S é submetido ao algoritmo de cálculo do controle dirigido ótimo. A cadeia ótima $s_o = g_{13}b_{10}g_{13}r_{10}r_{20}g_{30}$ é obtida, onde g_{13} representa que um bloco verde deve ser movido da $p1$ para a pilha $p3$ e assim por diante.

5 Estudo de Caso

Considera-se a configuração inicial da Figura 2 e pretende-se montar a pilha $p0$ como especificado no autômato da Figura 5, ou seja, uma pilha formada pelos blocos *azul > vermelho > vermelho > verde*. Para isso, seguem-se os passos definidos na seção 4.5.

- Identifica-se que há 3 pilhas ($P = 3$) e 8 blocos ($B = 8$) de 3 cores diferentes ($C = 3$).
- Dados B e C , calcula-se o tamanho mínimo das pilhas por $8 \leq T(3 - 1) + 1$, logo $T \geq 3, 5$. Sendo assim, define-se $T = 4$.
- Às pilhas $p1$, $p2$ e $p3$ são associados os autômatos G_{p1} , G_{p2} e G_{p3} respectivamente, projetados da maneira descrita na seção 4.1. Cada um deles no formato do exemplo da Figura 4, com o mesmo número de cores mas com tamanho $T = 4$. Estes autômatos possuem $|Q| = 121$ estados, $|\Sigma| = 15$ eventos e $|\rightarrow| = 600$ transições, mas devido à estrutura conhecida podem ser obtidos automaticamente. Para isso, o software MATLAB^{®1} foi utilizado.
- Os estados iniciais foram definidos para as pilhas modeladas por G_{p1} , G_{p2} e G_{p3} . No caso de G_{p3} , o estado inicial é um estado do 2º nível, que pode ser alcançado a partir do estado central por meio de qualquer cadeia $b_{i1}g_{i1}$, ou seja, qualquer cadeia que indique que um bloco azul deve ser movido de uma pilha qualquer i (exceto a pilha resultado $p0$) para a pilha 1, seguido de um bloco verde.

Para a criação dos autômatos do modelo a partir da configuração inicial das pilhas, foi desenvolvido um algoritmo em MATLAB[®]. Para obter o supervisor, utilizou-se o software TCT criado por Feng & Wonham (2006). O algoritmo do diretor ótimo utilizado é aquele implementado em Vieira & Pena (2013), também no MATLAB.

A Tabela 1 reúne as informações sobre o tamanho de cada autômato gerado nos passos descritos.

Tabela 1: Tamanho dos autômatos calculados.

Autômato(s)	$ Q $	$ \Sigma $	$ \rightarrow $
G_{p1}, G_{p2}, G_{p3}	121	15	600
G	29.564	27	202.254
E	5	27	84
S	13.590	27	68.232

A Figura 6 ilustra a configuração das pilhas durante a execução passo a passo da solução obtida após a realização dos procedimentos descritos. No canto superior esquerdo de cada imagem do sistema está o nome do último evento ocorrido. Na figura, a sequência de imagens segue a sequência de eventos da cadeia s_o , da esquerda pra direita, de cima para baixo.

6 Conclusão

Neste trabalho propõe-se uma nova variação do problema das torres de Hanói: um sistema constituído por pilhas de blocos coloridos, no qual objetiva-se montar uma nova pilha numa sequência específica de cores de blocos por meio da reorganização dos mesmos, realizando o menor número possível de movimentos. Desenvolveu-se para este problema um método de modelagem e

¹ <http://www.mathworks.com/>

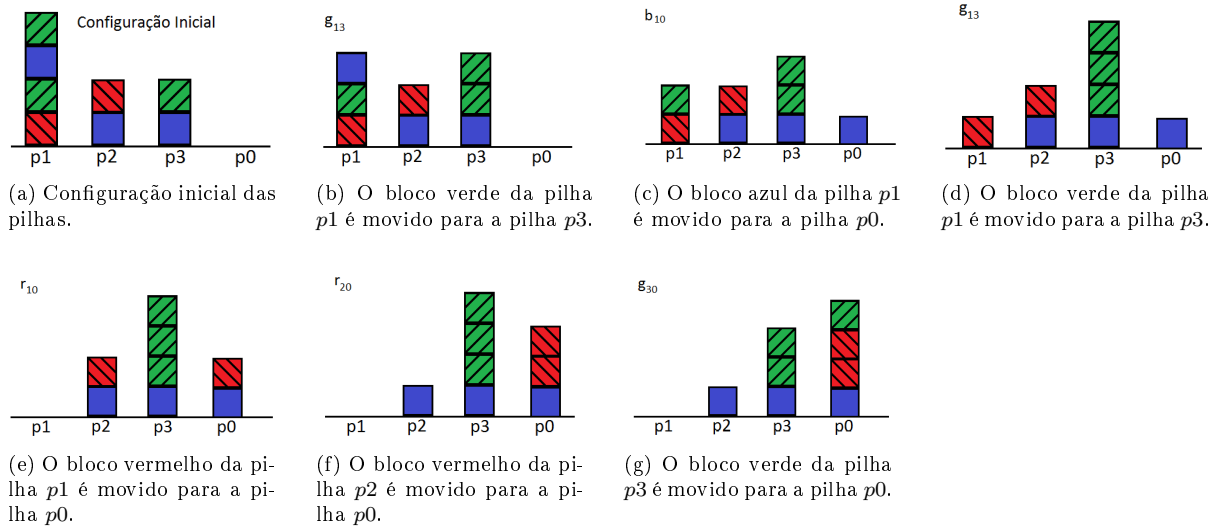


Figura 6: Passo a passo da solução gerada pelo diretor.

obtenção da solução ótima de forma automática dados a condição inicial do sistema e o objetivo a ser alcançado. Para isso, utilizaram-se as teorias do controle supervísório e do controle dirigido ótimo de sistemas a eventos discretos.

O potencial de generalização do problema de sequenciamento proposto aumenta a gama de trabalhos futuros e a aplicabilidade dos resultados. Pretende-se dar seqüência a este trabalho implementando a metodologia nele proposta em um sistema real, utilizando visão computacional para capturar as informações da configuração inicial e um braço robótico para mover os blocos.

O esforço computacional realizado para a construção do modelo é grande. O crescimento das pilhas, do número de pilhas e do número de cores pode tornar o problema insolúvel devido à complexidade computacional inerente ao processo de síntese do supervisor. Portanto, propõe-se para trabalhos futuros a busca de modelos de menor complexidade.

Agradecimentos

Este trabalho foi realizado com o apoio das agências FAPEMIG, CAPES e CNPq.

Referências

- Ahrabian, H., Badamchi, C., & Nowzari-Dalini, A. (2011). On the solution of the towers of hanoi problem. *5(1)*, 974 – 977.
- Feng, L. & Wonham, W. (2006). Tct: A computation tool for supervisory control synthesis. In

8th International Workshop on Discrete Event Systems (WODES'06) (pp. 388–389).

- Havur, G., Haspalamutgil, K., Palaz, C., Erdem, E., & Patoglu, V. (2013). A case study on the tower of hanoi challenge: Representation, reasoning and execution. In *IEEE International Conference on Robotics and Automation (ICRA '13)* (pp. 4552–4559).

- Huang, J. & Kumar, R. (2005). Nonblocking directed control of discrete event systems. In *44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC'05)* (pp. 7627–7632).

- Huang, J. & Kumar, R. (2008). Optimal non-blocking directed control of discrete event systems. *IEEE Transactions on Automatic Control*, *53(7)*, 1592–1603.

- Lucas, E. (1893). *Recreations Mathematiques*. Paris: Gauthier-Villars.

- Ramadge, P. & Wonham, W. (1989). The control of discrete event systems. *Proceedings of the IEEE; Special issue on Dynamics of Discrete Event Systems*, *77(1)(1)*, 81–98.

- Stockmeyer, P. & Lunnon, F. (2010). New variations on the tower of hanoi. *CONGRESSUS NUMERANTIUM*, *201*, 277–288.

- Vieira, G. L. & Pena, P. N. (2013). Aplicações e limitações do controle dirigido ótimo não-bloqueante de sistemas a eventos discretos. *XI Simpósio Brasileiro de Automação Inteligente (SBAI'13)*, *11*, 1–6.