

REPRESENTAÇÃO EM PROGRAMAÇÃO LINEAR INTEIRA MISTA DO PROBLEMA DE MINIMIZAÇÃO DE SUPERVISORES

FRANCISCO DE ASSIS CARVALHO DA SILVA NETO*, MAX HERING DE QUEIROZ*, EDUARDO CAMPONOGARA*

**Departamento de Automação e Sistemas
Universidade Federal de Santa Catarina
Caixa Postal 476 Campus Trindade, CTC
88.040-900 Florianópolis, SC, Brasil*

Emails: assisneto@yahoo.com, max@das.ufsc.br, camponog@das.ufsc.br

Abstract— In supervisory control theory (SCT), the optimal supervisor (which can be represented as a generator) disables events on the plant according to its current state. Such supervisor typically has a big number of states. It is desirable to reduce the supervisor by finding a supervisor of minimal size, what is NP-hard. In this paper we propose a mixed-integer linear programming formulation of the supervisor reduction problem through a set cover over the optimal supervisor state set.

Keywords— Discrete Event System, Supervisory Control Theory, Optimization, Mixed-Integer Linear Programming

Resumo— Na Teoria de Controle Supervisório (TCS), busca-se encontrar o supervisor ótimo (que pode ser representado por um gerador), que desabilita eventos de uma planta de acordo com seu estado atual. Este supervisor possui tipicamente um grande número de estados. É desejável reduzir seu tamanho de modo a encontrar um supervisor mínimo, tarefa que é NP-difícil. Neste trabalho, propõe-se uma representação em programação matemática do problema de minimização de supervisores por uma cobertura do espaço de estados do supervisor ótimo.

Palavras-chave— Sistemas a eventos discretos, Teoria de controle supervisório, Otimização, Programação linear inteira mista

1 Introdução

Na Teoria de Controle Supervisório (TCS) (Wonham 2006), o problema associado a um sistema a eventos discretos (SED) é o de impor um comportamento controlável, não-bloqueante e minimamente restritivo a uma planta, de forma a respeitar uma dada linguagem de especificação. A solução ótima para tal problema é a máxima linguagem controlável, que pode ser representada por um gerador (supervisor supremo) que habilita (ou não) a ocorrência de eventos na planta, o qual possui um espaço de estados cujo tamanho é da ordem do produto entre o número de estados do gerador da planta pelo da especificação. Tal supervisor geralmente contém informações redundantes relativas às restrições que já são obedecidas implicitamente pela planta, ou seja, é possível reduzir o tamanho do espaço de estados do supervisor supremo sem que sua ação de controle sobre a planta seja modificada. É desejável reduzir supervisores por ser mais fácil e econômico implementar supervisores com pequeno número de estados. Além disso, supervisores reduzidos são mais fáceis de serem interpretados por especialistas. Especialmente na utilização da Teoria de Controle Supervisório Modular Local (de Queiroz e Cury 2002), a redução de supervisores evidencia tais vantagens.

O problema da minimização de supervisores foi pioneiramente estudado por Vaz e Wonham (1986), que demonstraram a existência de um su-

pervisor mínimo (não único), que é obtido por meio de agrupamentos de estados (cobertura) do supervisor supremo. Além disso, propuseram um algoritmo de complexidade exponencial com relação ao número de estados do supervisor supremo para se minimizar um supervisor. Posteriormente, Su e Wonham (2003) provaram que o problema de minimização de supervisores é NP-difícil (Cormen et al. 2001) e propuseram um algoritmo para se encontrar uma cota inferior para o número de estados dos supervisores mínimos. Além disso, apresentaram um algoritmo de complexidade polinomial para se reduzir supervisores por meio de partições no espaço de estados do supervisor supremo. Quando o número de estados obtidos por este algoritmo de redução coincide com o obtido pelo de cálculo da cota inferior, o supervisor reduzido encontrado é mínimo. Embora o algoritmo proposto por Su e Wonham possua complexidade polinomial, ele não é aplicável para supervisores muito grandes. Por exemplo, dificilmente consegue-se reduzir supervisores com mais de 2000 estados com as ferramentas computacionais disponíveis atualmente (e.g. utilizando-se a ferramenta CTCT (Wonham 2006)).

Uma das formas de contornar a solução ótima de problemas NP-difíceis (Cormen et al. 2001) com grande espaço de soluções é por meio da solução de problemas de otimização (Wolsey 1998), que fazem uso de métodos iterativos envolvendo diversas etapas de tentativa e erro e do fato

de existir um ótimo global que possa ser alcançado utilizando-se condições de convergência pré-definidas. Para se utilizar tais técnicas, primeiramente é necessário que se consiga modelar o problema a ser resolvido em programação matemática (Wolsey 1998, Fourer et al. 1990), que é expressa em notação matemática e compreende basicamente uma função objetivo, variáveis de decisão e restrições.

Propõe-se neste trabalho uma representação em programação linear inteira mista do supervisor supremo, as possíveis coberturas em seu espaço de estados, bem como equações lineares de restrições que garantem que somente coberturas válidas, do ponto de vista da TCS, sejam obtidas. Além disso, apresenta-se uma função objetivo que, ao ser satisfeita, encontra um supervisor reduzido com número mínimo de estados. Expõem-se também resultados práticos de minimização de supervisores com a representação proposta utilizando-se o aplicativo CPLEX (ILOG 2003).

As técnicas de representação matemática e otimização utilizadas neste trabalho talvez não utilizem de maneira ótima as informações da estrutura do problema de minimização de supervisores. Não é pretensão deste trabalho propor um método de minimização de supervisores que resolva problemas de grande porte, mas apenas demonstrar que é possível tratar este tipo de problema utilizando-se ferramentas de otimização.

Na Seção 2 são apresentados conceitos básicos da TCS, redução de supervisores e programação linear inteira mista. Em seguida, na Seção 3, apresenta-se a representação do problema de minimização em programação linear inteira mista. Na Seção 4, apresentam-se um procedimento para solução do problema de minimização e seus resultados. Finalmente, na Seção 5, expõem-se as conclusões e perspectivas deste trabalho.

2 Preliminares

2.1 Teoria de Controle Supervisório

Na Teoria de Controle Supervisório (Wonham 2006), deseja-se que uma planta, representando o comportamento em malha aberta de um SED, respeite uma especificação de forma minimamente restritiva. A solução ótima para tal problema pode ser calculada automaticamente por um algoritmo de complexidade polinomial e ser representada pelo supervisor supremo $SupC$. A planta G é modelada como um gerador, da seguinte forma:

$$G = (\Sigma, Q, \delta, q_0, Q_m)$$

onde Q é o conjunto de estados, Σ é o conjunto de eventos (alfabeto), o qual é particionado em dois subconjuntos: o conjunto de eventos controláveis Σ_c e o conjunto de eventos não controláveis Σ_u , $\delta : \Sigma \times Q \rightarrow Q$ (função parcial) é a função de

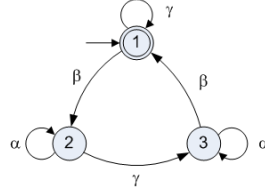


Figura 1: Planta G .

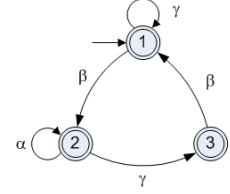


Figura 2: Gerador S .

	α	β	γ
1	dc	1	1
2	1	dc	1
3	0	1	dc

Tabela 1: Ψ de $SupC$

transição, $q_0 \in Q$ é o estado inicial e $Q_m \subseteq Q$ é o subconjunto de estados que são marcados. Por exemplo, a Figura 1 representa uma planta com $X = \{1, 2, 3\}$, $\Sigma = \{\alpha, \beta, \gamma\}$, $q_0 = 1$, $Q_m = \{1\}$ e δ definida de acordo com as transições do grafo.

O supervisor supremo para G que obedece de forma minimamente restritiva à especificação $K \in \Sigma^*$, onde Σ^* representa o conjunto de todas as possíveis cadeias finitas de um conjunto de eventos Σ , é um par $SupC = (S, \Psi)$, onde:

$$S = (\Sigma, X, \xi, x_0, X_m)$$

é um gerador, enquanto que:

$$\Psi : \Sigma \times X \rightarrow \{0, 1, dc\}$$

é uma lei de controle tal que:

$$\begin{aligned} \Psi(\sigma, x) &= 1, & \sigma \in \Sigma_u, x \in X \\ \Psi(\sigma, x) &\in \{0, 1, dc\}, & \sigma \in \Sigma_c, x \in X \end{aligned}$$

S é considerado como sendo dirigido externamente por uma cadeia $s \in \Sigma^*$ de eventos $\sigma \in \Sigma$ gerados pela planta G . Em contrapartida, enquanto S está no estado $x \in X$ e G no estado $q \in Q$, se a transição $\delta(\sigma, q)$ está definida ($\delta(\sigma, q)!$), então ela está sujeita à ação de controle $\Psi(\sigma, x)$. Se $\Psi(\sigma, x) = 0$ (desabilitado) então a transição $\delta(\sigma, q)$ está proibida de ocorrer, se $\Psi(\sigma, x) = 1$ (habilitado) então a transição $\delta(\sigma, q)$ está permitida, mas não forçada, a ocorrer. Para os casos em que $\Psi(\sigma, x) = dc$ (*don't care*) então o fato de se atribuir $\Psi(\sigma, x) = 0$ ou $\Psi(\sigma, x) = 1$ não faz diferença no comportamento de G , pois a transição $\delta(\sigma, q)$ não está definida ($-\delta(\sigma, q)!$). Um exemplo de gerador S de um supervisor supremo para a planta G da Figura 1 é mostrado na Figura 2. Neste caso, tem-se que a lei de controle Ψ é dada de acordo com a Tabela 1, desabilitando a ocorrência do evento α no estado 3 da planta.

2.2 Redução de supervisores

Dado o supervisor supremo $SupC = (S, \Psi)$, onde S é o gerador da máxima linguagem controlável

da especificação K sobre a planta G é possível encontrar um supervisor equivalente R , no que diz respeito à ação de controle sobre a planta G , que possui menor número de estados (Vaz e Wonham 1986). Tal redução pode ser obtida por meio de coberturas no conjunto de estados de S . Uma cobertura sobre um conjunto A qualquer é um conjunto de subconjuntos de A tais que sua união resulte em A e que nenhum deles seja vazio (uma partição, por exemplo, é um caso especial de cobertura na qual os subconjuntos possuem interseção vazia). Para que uma cobertura seja considerada válida para se conseguir um supervisor reduzido, ela deve respeitar algumas propriedades.

Uma cobertura sobre o conjunto de estados do gerador S do supervisor supremo $SupC = (S, \Psi)$ é uma *cobertura de controle* $C = \{X_i, i \in I\}$, tal que $X_i \subseteq X$ e I é um conjunto de índices, desde que as seguintes propriedades sejam respeitadas.

1. $\forall (i \in I) X_i \neq \emptyset$.
2. $\bigcup_{i \in I} X_i = X$ (*Cobertura completa*).
3. $\exists (I_m \subseteq I) : [(X_m = \bigcup_{i \in I_m} X_i) \wedge (X - X_m = \bigcup_{i \in (I - I_m)} X_i)]$ (*Consistência da marcação*).
4. $\forall (i \in I, \sigma \in \Sigma) \exists y \in X_i : \xi(\sigma, y)! \rightarrow \exists j \in I : \forall x \in X_i (\xi(\sigma, x)! \rightarrow \xi(\sigma, x) \in X_j)$ (*Determinismo*).
5. $\forall (i \in I, \sigma \in \Sigma) \forall (x, y \in X_i) (\Psi(\sigma, x) \neq dc \neq \Psi(\sigma, y)) \rightarrow (\Psi(\sigma, x) = \Psi(\sigma, y))$ (*Consistência da ação de controle*).

As propriedades 1 e 2 garantem que uma *cobertura de controle* seja realmente uma cobertura. A Propriedade 3 (*consistência da marcação*), impõe que não se agrupem, em um mesmo elemento da cobertura, estados do $SupC$ que possuam marcações diferentes. Esta propriedade pode ser relaxada, conforme exposto por Su e Wonham (2003) e, posteriormente, por Sivollela et al. (2006). No que diz respeito à implementação de supervisores, a marcação muitas vezes perde seu sentido, apesar de ser fundamental para se garantir não-bloqueio na síntese de $SupC$. Neste trabalho esta propriedade é ignorada para efeitos de redução de supervisores, considerando-se que a marcação do sistema controlado é estabelecida apenas pela planta. Isso reduz a complexidade dos algoritmos e pode melhorar o resultado da minimização.

A Propriedade 4 (*determinismo*) garante que não sejam agrupados estados que, ao se construir o supervisor reduzido, impliquem que a ocorrência de um evento a partir de um estado leve a múltiplos estados. A Propriedade 5 (*consistência da ação de controle*) assegura que todos os estados agrupados possuam ação de controle consistente, isto é, que não habilitem e desabilitem simultaneamente o mesmo evento.

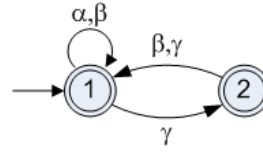


Figura 3: \bar{S}

	α	β	γ
1	1	1	1
2	0	1	1

Tabela 2: $\bar{\Psi}$

A partir de uma *cobertura de controle* C tal que $|C| < |Q|$, onde $|\bullet|$ representa o número de elementos de seu argumento, é possível se obter um supervisor reduzido $R = (\bar{S}, \bar{\Psi})$ da seguinte forma:

- $\bar{S} := (\Sigma, I, \bar{\xi}, i_0, I_m)$;
- seleciona-se $i_0 \in I$ tal que $x_0 \in X_{i_0}$;
- define-se $\bar{\xi} : \Sigma \times I \rightarrow I$ (função parcial) como: $\forall (i \in I, \sigma \in \Sigma)$, se $\xi(\sigma, x)!$ para algum $x \in X_i$, seleciona-se $j \in I$ tal que $\xi(\sigma, x) \in X_j$ para tal x . Define-se então $\bar{\xi}(\sigma, i) := j$;
- $I_m \subseteq I$ é o conjunto de todo $i \in I$ tal que para algum $x \in X_m$ tenha-se que $x \in X_i$.
- $\bar{\Psi} : \Sigma \times I \rightarrow \{0, 1, dc\}$ é definida como: para $\sigma \in \Sigma, i \in I$, se existe $x \in X_i$ tal que $\Psi(\sigma, x) \neq dc$ então $\bar{\Psi}(\sigma, i) := \Psi(\sigma, x)$, caso contrário $\bar{\Psi}(\sigma, i) := dc$;

Percebe-se pela definição acima que existem etapas na construção do supervisor reduzido nas quais há mais de uma possibilidade de escolha a ser tomada, por exemplo o estado inicial $i_0 \in I$ pode ser qualquer um desde que $x_0 \in X_{i_0}$. Isto demonstra que uma mesma cobertura de controle pode gerar diferentes supervisores, no que diz respeito a $i_0, \bar{\xi}$ e $\bar{\Psi}$, mas não ao número de estados que será $|C|$. Para os exemplos de planta e supervisor supremo já citados, pode-se construir um supervisor reduzido conforme a Figura 3 e a Tabela 2 a partir da cobertura de controle $C = \{\{1, 2\}, \{1, 3\}\}$.

2.3 Programação Linear Inteira Mista

O termo otimização (Wolsey 1998) é utilizado na matemática para se referir ao estudo de problemas nos quais se deseja minimizar (ou maximizar) uma função real $f : A \rightarrow \mathbb{R}$, sendo o domínio $A \subseteq \mathbb{R}^n$, chamado de *espaço de busca*, um conjunto especificado por um conjunto de restrições, que tipicamente são equações e inequações, que os elementos de A (conhecidos como *soluções factíveis*) devem satisfazer. No caso específico em que as restrições que delimitam o domínio $A \subseteq \mathbb{R}^n$ são expressas através de equações e inequações lineares e que a função objetivo é linear, garante-se que tanto o espaço de busca A quanto a função objetivo f são convexos, logo qualquer mínimo (máximo) local também será o global. Problemas desta classe

são conhecidos como problemas de *Programação Linear*, que geralmente podem ser resolvidos eficientemente mesmo no pior caso.

Existem problemas de programação linear nos quais algumas variáveis são restritas a assumirem somente valores inteiros. Tais problemas são denominados problemas de *Programação Linear Inteira Mista* e neste caso, diferentemente de sua versão mais genérica que faz uso de variáveis reais quaisquer, não necessariamente há um método que os resolva eficientemente, pois normalmente expressam problemas NP-difíceis.

Um problema de programação linear inteira pode ser expresso em programação matemática da seguinte forma:

$$\begin{aligned} \text{Minimize} \quad & c^T x \\ \text{Sujeito a:} \quad & Ax \geq b \\ & Cx = d \\ & x \in \mathbb{Z}_+^n \end{aligned}$$

onde $n \in \mathbb{Z}_+$ é o número de variáveis do problema, A é uma matriz $m \times n$, sendo $m \in \mathbb{Z}_+$ o número de inequações do problema, C é uma matriz $p \times n$, sendo $p \in \mathbb{Z}_+$ o número de equações do problema e b, c e d são vetores de ordem m, n e p , respectivamente.

Em um problema de programação linear inteira mista, não necessariamente $x \in \mathbb{Z}_+^n$, pois algumas componentes de x podem pertencer ao conjunto dos reais, porém pelo menos uma deve pertencer ao conjunto dos inteiros.

3 Programação matemática

Nesta seção, a descrição do problema de minimização de supervisores em *Programação Linear Inteira Mista* é apresentada. As restrições lineares às quais as variáveis do problema estão sujeitas implicam que somente coberturas válidas, de acordo com as propriedades apresentadas na Seção 2.2, sejam candidatas a soluções factíveis.

Embora a restrição de *consistência da marcação* tenha sido ignorada, ela pode ser facilmente inserida com formulação análoga à da restrição de *consistência da ação de controle*, que é explicitada nas próximas seções.

3.1 Dados do problema

Sem perda de generalidade, assume-se que $X = \{1, 2, \dots, |X|\}$, $\Sigma = \{1, 2, \dots, |\Sigma|\}$ e $I = \{1, 2, \dots, |I|\}$. Desta forma, definem-se os seguintes termos (considerados dados para o problema de minimização) baseados em $SupC = (S, \Psi)$:

$$T_{j,k,l} := \begin{cases} 1 & \text{se } \xi(k, j)! \wedge \xi(k, j) = l \\ 0 & \text{caso contrário} \end{cases}$$

o termo T mapeia cada tripla $(j \in X, k \in \Sigma, l \in X)$ a um valor binário. Ela assume valor igual

a 1 quando há uma transição partindo do estado j por meio da ocorrência do evento k com destino ao estado l , caso contrário seu valor é zero. Por exemplo, para o gerador do supervisor representado na Figura 2, $T_{1,3,1} = T_{1,2,2} = T_{2,1,2} = T_{2,3,3} = T_{3,2,1} = 1$ e $T_{j,k,l} = 0$ para todos os outros j, k, l (para todos os exemplos desta seção considera-se que os eventos de Σ são etiquetados da seguinte forma $\alpha = 1, \beta = 2$ e $\gamma = 3$).

$$H_{j,k} := \begin{cases} 1 & \text{se } \Psi(k, j) = 1 \\ 0 & \text{caso contrário} \end{cases}$$

H mapeia cada par $(j \in X, k \in \Sigma)$ a 1 caso k esteja explicitamente habilitado em j , ou a 0 caso contrário. Por exemplo, para a lei de controle representada na Tabela 1, $H_{1,2} = H_{1,3} = H_{2,1} = H_{2,3} = H_{3,2} = 1$ e $H_{j,k} = 0$ para todos os outros j, k .

$$D_{j,k} := \begin{cases} 1 & \text{se } \Psi(k, j) = 0 \\ 0 & \text{caso contrário} \end{cases}$$

D mapeia cada par $(j \in X, k \in \Sigma)$ a 1 caso k esteja explicitamente desabilitado em j , ou a 0 caso contrário. Por exemplo, para a lei de controle representada na Tabela 1, $D_{3,1} = 1$ e $D_{j,k} = 0$ para todos os outros j, k .

3.2 Variáveis do problema

As variáveis que devem ter seus valores ajustados a cada iteração de forma a encontrar a solução ótima são definidas a seguir:

1. $c_{i,j} \in \mathbb{Z}_+$, representa uma cobertura. Sendo que se um elemento $i \in I$ da cobertura contém o estado $j \in X$ então $c(i, j) = 1$ e 0 caso contrário.
2. $t_{i,k,l} \in \mathbb{Z}_+$, representa as transições dos elementos $i \in I$ da cobertura. Assumindo valor 1 quando existe pelo menos um estado $j \in X$ pertencente ao elemento $i \in I$ da cobertura que, através do símbolo $k \in \Sigma$, atinge o estado $l \in X$.
3. $z_{i,k,v} \in \mathbb{Z}_+$, é a variável que auxilia na análise de determinismo de uma cobertura de controle, assumindo valor 1 caso todos t de $i \in I, k \in \Sigma$ levem a um subconjunto do elemento da cobertura $v \in I$. Caso contrário, assume valor zero.
4. $r_i \in \mathbb{Z}_+$ deve assumir valor 1 quando o elemento $i \in I$ da cobertura contém um ou mais estados e valor 0 caso contrário.
5. $h_{i,k} \in \mathbb{Z}_+$ deve assumir valor 1 quando o elemento $i \in I$ da cobertura contém um ou mais estados que habilitam o evento $k \in \Sigma$. Assume o valor 0 caso contrário.
6. $d_{i,k} \in \mathbb{Z}_+$ deve assumir valor 1 quando o elemento $i \in I$ da cobertura contém um ou mais

estados que desabilitam o evento $k \in \Sigma$. Assume o valor 0 caso contrário.

3.3 Problema de minimização

Dados T, H, D e sendo $i, v \in I$, $j, l \in X$, $k \in \Sigma$, o problema de minimização de supervisores em programação linear inteira mista é definido como:

$$\text{Minimize: } \sum_{i \in I} r_i$$

Sujeito a:

$$\forall j \quad \sum_{i \in I} c_{i,j} \geq 1 \quad (1)$$

$$\forall i \quad r_i \leq 1 \quad (2)$$

$$\forall i \quad r_i \leq \sum_{j \in X} c_{ij} \quad (3)$$

$$\forall i, j \quad r_i \geq c_{i,j} \quad (4)$$

$$\forall i, k, l \quad t_{i,k,l} \leq 1 \quad (5)$$

$$\forall i, k, l \quad t_{i,k,l} \leq \sum_{j \in X} (c_{i,j} \times T_{j,k,l}) \quad (6)$$

$$\forall i, k, l, j \quad t_{i,k,l} \geq c_{i,j} \times T_{j,k,l} \quad (7)$$

$$\forall i, k \quad h_{i,k} \leq 1 \quad (8)$$

$$\forall i, k \quad h_{i,k} \leq \sum_{j \in X} (c_{i,j} \times H_{j,k}) \quad (9)$$

$$\forall i, k, j \quad h_{i,k} \geq c_{i,j} \times H_{j,k} \quad (10)$$

$$\forall i, k \quad d_{i,k} \leq 1 \quad (11)$$

$$\forall i, k \quad d_{i,k} \leq \sum_{j \in X} (c_{i,j} \times D_{j,k}) \quad (12)$$

$$\forall i, k, j \quad d_{i,k} \geq c_{i,j} \times D_{j,k} \quad (13)$$

$$\forall i, k \quad \sum_{v \in I} z_{i,k,v} = 1 \quad (14)$$

$$\forall i, k, v, l \quad z_{i,k,v} \leq c_{v,l} - t_{i,k,l} + 1 \quad (15)$$

$$\forall i, k \quad h_{i,k} + d_{i,k} \leq 1 \quad (16)$$

A Equação 1 garante que cada estado pertença a pelo menos um elemento da cobertura $c_{i,j}$. Dado que $r_i \in \mathbb{Z}_+$, a Equação 2 garante que $r_i \in \{0, 1\}$, a Equação 3 força $r_i = 0$ caso X_i seja um elemento vazio da cobertura e a Equação 4 garante $r_i \geq 1$ quando o elemento da cobertura X_i possui pelo menos um elemento. Ou seja, o conjunto de equações lineares 2, 3 e 4 fazem com que $r_i = 0$ se $\sum_{j \in X} c_{ij} = 0$ ou $r_i = 1$ se $\sum_{j \in X} c_{ij} \geq 1$. Os grupos de equações $\{5, 6, 7\}$, $\{8, 9, 10\}$ e $\{11, 12, 13\}$ realizam o mesmo trabalho para as variáveis t, h e d respectivamente. As equações 14 e 15 garantem que as condições de determinismo sejam satisfeitas. A condição de consistência da ação de controle é garantida pela Equação 16.

Pode-se demonstrar que, uma vez encontrada a solução ótima para uma dada instância do problema, obtém-se a cobertura de controle ótima pelo valor da variável c (que pode ser vista como uma matriz de dimensões $|I| \times |X|$), ignorando-se possíveis linhas formadas apenas por zeros. Por

exemplo, para o supervisor de exemplo apresentado na Seção 2, a solução ótima pode ser representada por c , tal que $c_{11} = c_{12} = c_{21} = c_{23} = 1$ e $c_{13} = c_{22} = 0$, ou seja $C = \{\{1, 2\}, \{1, 3\}\}$, que pode gerar um supervisor conforme representado pela Figura 3 e pela Tabela 2.

4 Solução e resultados

Por se tratar de um problema NP-difícil, o espaço de busca se torna imenso à medida em que o tamanho de uma instância do problema aumenta. Um alternativa para contornar este problema é a de diminuir o espaço de busca, reduzindo-se $|I|$, da seguinte forma.

1. Estima-se a cota inferior $l_b \in \mathbb{Z}_+$ do número de estados do supervisor mínimo (Su e Wonham 2003, Sivollela et al. 2006).
2. Executa-se o algoritmo de solução limitando-se a dimensão máxima da variável c a l_b .
3. Caso não haja solução factível e $l_b < |X|$, faz-se $l_b = l_b + l_b p$ (limitando-se l_b a $|X| - 1$) e retorna-se ao passo anterior, onde $p \in (0, 1] \in \mathbb{R}$ é um passo que incrementa a cota inferior a cada execução do algoritmo, aumentando seu espaço de busca.

No pior caso (quando não é possível reduzir uma dada instância do problema), a execução dos passos sugeridos anteriormente causa processamento desnecessário, uma vez que seria possível concluir isso executando-se o algoritmo uma única vez com $|I| = |X|$. Porém, para casos em que a taxa de redução do supervisor mínimo em relação ao supervisor supremo é grande, os passos acima resultam em grandes ganhos de eficiência.

A representação do problema de minimização de supervisores em programação linear inteira mista foi implementada utilizando-se a linguagem de programação matemática AMPL (Fourer et al. 1990). A ferramenta utilizada para solucionar instâncias de problemas de minimização foi o aplicativo CPLEX (ILOG 2003) executando em um Intel Pentium 4 2.66 GHz com 1 Gb de memória RAM e GNU/Linux Mandrake 9.2.

Exemplos de diferentes tamanhos foram minimizados para que se pudesse avaliar a eficiência da solução do problema de minimização de supervisores por métodos de otimização e, também, compará-lo a métodos tradicionais. A Tabela 3 mostra comparações entre o método para se solucionar o problema de programação linear inteira mista (PLIM) e o utilizado pelo aplicativo CTCT (no mesmo computador), quanto ao número de estados obtidos pela redução e o tempo de execução em segundos. Em todas as soluções por PLIM apresentadas nesta tabela, limitou-se inicialmente o tamanho da cobertura de controle à cota inferior

SupC	Nº estados		T. exec. (s)	
	CTCT	PLIM	CTCT	PLIM
3	3	2	< 1	0,010
6	3	2	< 1	0,020
12	3	2	< 1	0,060
24	3	2	< 1	0,140
48	3	2	< 1	0,530
96	3	2	< 1	2,030
192	3	2	≈ 19	9,040
384	3	2	≈ 1230	51,570
768	-	-	> 1500	f.m.

Tabela 3: Resultados comparativos

SupC	Tamanho limite da cobertura de controle				
	2	3	5	6	10
3	0,01	0,02	-	-	-
6	0,02	0,11	0,31	0,72	-
12	0,06	0,50	3,44	10,15	100,95
24	0,14	4,66	62,47	93,39	> 600
48	0,53	28,98	161,67	> 600	-
96	2,03	164,93	> 600	-	-
192	9,04	> 600	-	-	-
384	51,57	> 600	-	-	-

Tabela 4: Resultados variando-se $|I|$

do tamanho do supervisor mínimo ($l_b = 2$). A última linha da tabela, indica que o problema não foi solucionado por PLIM em virtude de falta de memória (f.m.), que é uma questão crítica na minimização de supervisores por PLIM, pois o número de variáveis dinâmicas é da ordem $\Theta(|X| \times |\Sigma| \times |I|)$, enquanto que as equações e inequações de restrições é da ordem $\Theta(|X|^2 \times |\Sigma| \times |I|)$.

Conforme demonstrado na Tabela 3, a solução por PLIM mostrou-se eficiente para instâncias com até de 384 estados e que podem ser bastantes reduzidas. A Tabela 4 mostra o tempo de execução em segundos do algoritmo de solução por PLIM para cada instância da Tabela 3, porém variando-se a limitação inicial da cobertura de controle. Os resultados da Tabela 4 mostram que, caso se inicie a solução por PLIM limitando-se menos o tamanho da cobertura de controle, fato que seria inevitável em instâncias que possuem baixa taxa de minimização (e consequentemente uma alta cota inferior l_b), piora-se de maneira crítica a eficiência do algoritmo.

5 Conclusões e perspectivas

Este trabalho mostra que é possível representar o problema de minimização de supervisores utilizando-se programação linear inteira mista. Embora o algoritmo convirja sempre para uma solução ótima, o custo para tal convergência pode ser impraticável, dado que é um problema NP-difícil. Para estes casos, pode-se interromper o algoritmo durante sua execução e verificar se a atual solução subótima é aceitável com relação à cota inferior. Além disso, pode-se obter uma melhor cota inferior a partir da solução subótima do

problema dual (Wolsey 1998).

Apesar de ser inviável a redução de supervisores grandes (com mais de 1000 estados) nesta implementação, abre-se perspectivas para um estudo mais detalhado deste problema relacionado à área de programação linear inteira mista. Acredita-se que é possível melhorar a eficiência do algoritmo buscando cortes específicos que façam uso da estrutura do problema quando solucionado por algoritmos *branch and bound*, bem como do significado de relaxações lineares do problema. Atualmente os autores investigam a busca por soluções alternativas de otimização, como por exemplo, computação evolutiva (Holland 1975). Uma vez que o problema seja adequadamente formulado nesse contexto, espera-se obter reduções próximas da solução ótima para supervisores maiores.

Referências

- Cormen, T. H., Leiserson, C. E., Rivest, R. L. e Stein, C. (2001). *Introduction to Algorithms*, The MIT Press.
- de Queiroz, M. e Cury, J. (2002). Synthesis and implementation of local modular supervisory control for a manufacturing cell, *Proc. 6th International Workshop on Discrete Event Systems (WODES02)*.
- Fourer, R., Gay, D. M. e Kernighan, B. W. (1990). *A Modeling Language for Mathematical Programming*.
- Holland, J. (1975). *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.
- ILOG (2003). *ILOG CPLEX 9.0 User's Manual*.
- Sivollela, L. F., da Cunha, A. E. C. e Ades, R. (2006). Redução de supervisores como ferramenta para a implementação de supervisores em controladores discretos, *XVI Congresso Brasileiro de Automática*.
- Su, R. e Wonham, W. (2003). *Supervisor Reduction for Discrete-Event Systems*, PhD thesis, Systems Control Group, University of Toronto.
- Vaz, A. e Wonham, W. (1986). On supervisor reduction in discrete-event systems, *International Journal of Control*.
- Wolsey, L. A. (1998). *Integer Programming*, John Wiley.
- Wonham, W. M. (2006). *Supervisory Control of Discrete-Event Systems*, Systems Control Group, University of Toronto.