

MINIMIZAÇÃO DE MAKESPAN E CONSUMO ENERGÉTICO EM SISTEMAS A EVENTOS DISCRETOS

GUSTAVO CAETANO RAFAEL*, LUCAS VINÍCIUS RIBEIRO ALVES†, LUCAS DE SOUZA BATISTA‡, PATRICIA N. PENA§

*Programa de Pós-Graduação em Engenharia Elétrica - Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brasil

†Colégio Técnico - Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brasil

‡Departamento de Engenharia Elétrica, Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brasil

§Departamento de Engenharia Eletrônica, Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brasil

Abstract— It is common to observe in an industrial production line a great heterogeneity among the equipment used. Often, the differences among the type of machines allocated to a task can have a major impact on energy consumption and production time (makespan). In this sense, the Supervisory Control Theory (SCT) proves to be very useful, since it allows modeling the production system respecting its constraints and ensuring the safe execution on the real plant. In this work, the production time (Makespan) and the energy consumption minimization problem in Discrete Event Systems (DES) with multiple production routes is addressed as a multiobjective problem. For this, a model was developed, which takes into account the information regarding the power demanded by the system in each state of the supervisor. The multiobjective modeling proposed in the present work is an extension to the Supervisory Control and Optimization (SCO) approach, which is applied to mono-objective problems. Promising results were found regarding the resources allocation and the objective minimization.

Keywords— Discrete Event Systems, Multiobjective Optimization, Production Planning.

Resumo— É comum observar em uma linha de produção industrial uma grande heterogeneidade entre os equipamentos utilizados. Muitas vezes, as diferenças entre os tipos de máquinas alocadas para uma tarefa pode causar um grande impacto no consumo energético e no tempo de produção (*makespan*). Nesse sentido, a Teoria de Controle Supervisório (TCS) se mostra bastante útil, pois ela possibilita modelar o sistema de produção respeitando suas restrições e garantindo execução segura na planta real. Neste trabalho é tratado o problema de minimização do tempo de produção (*makespan*) e do consumo energético em Sistemas a Eventos Discretos (SED) com múltiplas rotas de produção como um problema multiobjetivo. Para tal, foi desenvolvida uma modelagem, que leva em consideração as informações relativas à potência demandada pelo sistema em cada estado do supervisor. A modelagem multiobjetivo proposta no presente trabalho é uma extensão à abordagem Controle Supervisório e Otimização (CSO) aplicada em problemas mono-objetivo. Os resultados foram promissores quanto a alocação de recursos e a minimização dos objetivos.

Palavras-chave— Sistemas a Eventos Discretos, Otimização Multiobjetivo, Planejamento da Produção.

1 Introdução

A produtividade é um dos aspectos mais importantes no contexto da automação industrial e o tempo é usualmente o recurso mais valioso. Nesse cenário, a pesquisa em técnicas de escalonamento e planejamento da produção são a chave para responder à demanda por eficiência (Wang et al., 2008). Porém, no caso de sistemas de produção flexível, com múltiplas rotas de produção onde cada rota apresenta características diferentes, pode ser interessante analisar a melhor forma de se distribuir a produção entre essas rotas. Nesse sentido, outro objetivo a ser analisado é o consumo energético, visto que as diferentes formas de produzir um mesmo produto, devido à heterogeneidade das máquinas, podem apresentar consumos energéticos totais diferentes. Além disso, sistemas industriais estão submetidos à limitações de potência contratada e a ultrapassagem desse valor pode levar a cobranças adicionais (ANEEL, 2010) que, em geral, são indesejadas.

O escalonamento de tarefas é a alocação, no tempo, de recursos finitos a tarefas, utilizando para tal um critério de otimização (Pinedo, 2012). Quando um problema de escalonamento não leva em consideração restrições quanto aos recursos que estão sendo utilizados e apenas usa restrições temporais e de precedência, ele pode ser tratado como um problema de planejamento (scheduling) (Ghallab et al., 2004). Este tipo de problema pertence à classe de otimização combinatória, na qual problemas mono-objetivo são NP-completos, ou seja, eles não podem ser resolvidos através de algoritmos de tempo polinomial (Garey & Johnson, 1980). Existem algoritmos eficientes (tempo polinomial) apenas para os problemas mais simples de planejamento, mas a maioria dos problemas multiobjetivo são NP-difíceis (Pinedo, 2012). Com surgimento de métodos heurísticos e metaheurísticos, houve o interesse de pesquisadores em utilizá-los para resolver problemas NP-difíceis de otimização combinatória multiobjetivo (Ehrgott & Gandibleux, 2000). Assim,

trabalhos como (Coelho et al., 2012) e (Gomes et al., 2014) avaliam o desempenho e complexidade de heurísticas como Multiobjective Variable Neighborhood Search (MOVNS), Multiobjective GRASP (MOG) entre outros, a fim de selecionar o mais adequado ao seu problema de planejamento.

No contexto de Sistemas a Eventos Discretos (SED), os problemas são modelados por meio de autômatos e linguagens, associados com a Teoria de Controle Supervisório (TCS) para obter um supervisor (Ramadge & Wonham, 1989) que garante a operação segura do sistema. O supervisor fornece as restrições de precedência de eventos, permitindo que o problema de sequenciamento possa ser tratado como um problema de planejamento de produção (Su et al., 2012; Pena et al., 2016).

Dentre os diversos formalismos para tratar o problema de planejamento da produção em SEDs, pode-se observar, na literatura, diversas técnicas que buscam minimizar o tempo de produção (*makespan*) ou maximizar o *throughput* na produção contínua, como no contexto de programação matemática (Schrijver, 1986), redes de Petri (López-Mellado et al., 2005), Autômatos Temporizados (Abdeddaïm et al., 2006), Verificação Formal (Herzig et al., 2014), entre outros. Entretanto, a literatura não apresenta ferramentas multiobjetivo para o tratamento do problema de planejamento da produção no contexto de Sistemas a Eventos Discretos, principalmente com o objetivo de minimizar o consumo energético e *makespan*.

O problema de minimização de tempo de produção é, no caso geral, não polinomial (Garey & Johnson, 1980), tornando boa parte dos problemas industriais computacionalmente intratáveis. Desse modo, usualmente são aplicadas heurísticas e metaheurísticas na solução desses problemas (Pena et al., 2016).

Neste trabalho, estendemos o trabalho anterior de (Costa et al., 2014), o qual introduz a metodologia Controle Supervisório e Otimização (CSO) aplicada em problemas mono-objetivo, para o universo de problemas multiobjetivo. A metodologia CSO original combina a teoria de controle supervisório e técnicas de otimização para resolver problemas de planejamento mono-objetivo. Neste contexto, propõe-se uma versão dedicada do VNS (*Variable Neighborhood Search*) multiobjetivo para realizar simultaneamente a minimização de tempo de produção (*makespan*) e consumo energético em sistemas de produção com múltiplas rotas. Além disso, são realizados experimentos para avaliar os impactos da otimização, quando toda produção é submetida a uma restrição de potência (quantidade máxima de potências demandada pelo sistema) e os efeitos da produção de um mesmo produto por rotas distintas.

Este artigo está organizado da seguinte forma. Na Seção 2 conceitos preliminares sobre sistemas a eventos discretos são apresentados. Na Seção 3 são apresentadas as novas definições e processo de contagem de tempo. O abordagem de otimização utilizada e o problema proposto são discutidos na

Seção 4. Os resultados são discutidos na Seção 5, e as conclusões são apresentadas na Seção 6.

2 Preliminares

Nessa seção são apresentadas as principais definições de suporte ao desenvolvimento deste trabalho. Para explicações mais detalhadas, o leitor deve dirigir-se à (Cassandras & Lafortune, 2008).

Seja Σ um conjunto finito não vazio de eventos, chamado de alfabeto. O comportamento de um SED é modelado como cadeias sobre Σ . O Fechamento Kleene Σ^* é o conjunto de todas as cadeias que podem ser construídas utilizando os símbolos do alfabeto Σ , incluindo a cadeia vazia ϵ . Um subconjunto $L \subseteq \Sigma^*$ é chamado linguagem. A concatenação de cadeias $s, u \in \Sigma^*$ é representada como su .

Uma cadeia $s \in \Sigma^*$ é chamada *prefixo* de $t \in \Sigma^*$, escrito $s \leq t$, se existe $u \in \Sigma^*$ tal que $su = t$. O prefixo fechamento \bar{L} de uma linguagem $L \subseteq \Sigma^*$ é o conjunto de todos os prefixos de cadeias em L , isto é, $\bar{L} = \{s \in \Sigma^* \mid s \leq t \text{ para algum } t \in L\}$.

Definição 1 *Um autômato finito determinístico é uma 5-tupla $G = (Q, \Sigma, \delta, q_0, Q_m)$ onde Q é um conjunto finito de estados, Σ é um conjunto finito de eventos (alfabeto), $\delta : Q \times \Sigma \rightarrow Q$ é a função de transição, $q_0 \in Q$ é o estado inicial e $Q_m \subseteq Q$ é o conjunto de estados marcados.*

A função de transição pode ser estendida para reconhecer cadeias sobre Σ^* como $\delta(q, \sigma s) = q'$ se $\delta(q, \sigma) = x$ e $\delta(x, s) = q'$.

As linguagens gerada e marcada são, respectivamente, $\mathcal{L}(G) = \{s \in \Sigma^* \mid \delta(q_0, s) = q' \wedge q' \in Q\}$ e $\mathcal{L}_m(G) = \{s \in \Sigma^* \mid \delta(q_0, s) = q' \wedge q' \in Q_m\}$.

A função de eventos ativos, definida por $\Gamma : Q \rightarrow 2^\Sigma$, é, dado um estado q , o conjunto de eventos $\sigma \in \Sigma$ para os quais $\delta(q, \sigma)$ é definido.

Definição 2 *Seja $G_1 = (Q_1, \Sigma_1, \delta_1, q_{01}, Q_{m1})$ e $G_2 = (Q_2, \Sigma_2, \delta_2, q_{02}, Q_{m2})$. O produto síncrono de G_1 e G_2 é:*

$$G_{12} = (Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, \delta_{12}, (q_{01}, q_{02}), Q_{m1} \times Q_{m2})$$

onde

$$\delta((q_1, q_2), e) = \begin{cases} (\delta_1(q_1, e), \delta_2(q_2, e)), & \text{se } e \in \Gamma_1(q_1) \cap \Gamma_2(q_2) \\ (\delta_1(q_1, e), q_2), & \text{se } e \in \Gamma_1(q_1) \setminus \Sigma_2 \\ (q_1, \delta_2(q_2, e)), & \text{se } e \in \Gamma_2(q_2) \setminus \Sigma_1 \\ \text{indefinido}, & \text{caso contrário.} \end{cases}$$

$$e \in \Gamma_{1||2}(q_1, q_2) = [\Gamma_1(q_1) \cap \Gamma_2(q_2)] \cup [\Gamma_1(q_1) \setminus \Sigma_2] \cup [\Gamma_2(q_2) \setminus \Sigma_1].$$

Exemplo 1 *Na Figura 1, pode-se observar o autômato G que é proveniente do produto síncrono entre os autômatos G_1 e G_2 .*

A Teoria de Controle Supervisório é um método formal, baseado na teoria de linguagens e autômatos, para o cálculo sistemático de supervisores. O

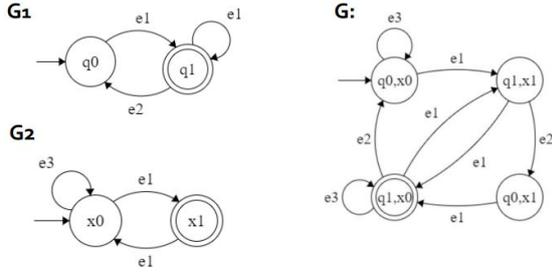


Figura 1: Produto síncrono entre os autômatos G_1 e G_2 resultando no autômato G

sistema a ser controlado é chamado *planta*, o controlador é o agente chamado *supervisor* e o problema de controle é encontrar um supervisor que garanta especificações de desempenho e segurança da maneira minimamente restritiva.

A planta é modelada por um autômato $G = (Q, \Sigma, \delta, q_0, Q_m)$ e $\Sigma = \Sigma_c \cup \Sigma_{nc}$ onde Σ_c é o conjunto de eventos controláveis, que podem ser desabilitados por um agente externo, e Σ_{nc} é o conjunto de eventos não controláveis, que não podem ser desabilitados por um agente externo, como o supervisor. A planta representa o modelo lógico de um SED, o comportamento do sistema sem nenhuma ação de controle. O papel de um supervisor S é regular o comportamento da planta e alcançar um comportamento K desabilitando eventos controláveis.

Seja E um autômato que representa especificações impostas à planta G . Dizemos que $K = \mathcal{L}_m(G||E) \subseteq \mathcal{L}_m(G)$ é controlável com respeito a G se $K\Sigma_{nc} \cap \mathcal{L}(G) \subseteq \bar{K}$. Existe um supervisor não bloqueante V para G tal que $\mathcal{L}_m(V/G) = K$ se, e somente se, K é controlável com respeito à G . Se K não satisfaz a condição, então a suprema sub linguagem controlável e não bloqueante $Sup\mathcal{C}(K, G)$ pode ser sintetizada, representando o supervisor não bloqueante minimamente restritivo. Para G e K , um supervisor monolítico, representado por um autômato S , pode ser computado para representar $Sup\mathcal{C}(K, G)$ de modo que $\mathcal{L}_m(S) = Sup\mathcal{C}(K, G) \subseteq K$ (Ramadge & Wonham, 1989).

As linguagens gerada e marcada de uma planta G sobre a ação de um supervisor S são, respectivamente, $\mathcal{L}(S/G)$ e $\mathcal{L}_m(S/G) \subseteq \mathcal{L}(S/G)$. Um supervisor S é dito não bloqueante quando $\mathcal{L}_m(S/G) = \mathcal{L}(S/G)$.

A partir deste ponto, chama-se supervisor um autômato S tal que $S = S||G$, ou seja, a dinâmica do supervisor engloba a dinâmica da planta sob controle.

3 Desenvolvimento

A ideia de potência é introduzida como uma propriedade de um estado de um autômato finito determinístico. Nesse contexto, um determinado estado pode estar associado uma potência relativa

ao trabalho em execução.

Uma maneira de representar a potência em um estado é por meio da definição de uma função, chamada função de potência:

Definição 3 Seja $G = (Q, _, _, _, _)$ um autômato. A função potência associada à G , definida como $f_p : Q \rightarrow \mathbb{R}^*$, é tal que, para cada estado $q \in Q$, associa um número real não negativo.

Usualmente é mais simples estabelecer a potência nos autômatos das plantas, ao invés do autômato do supervisor, tornando então necessária a definição de uma nova função de potência para a composição das plantas.

Definição 4 Sejam $G_1 = (Q_1, _, _, _, _)$ e $G_2 = (Q_2, _, _, _, _)$ autômatos e sejam f_{p_1} e f_{p_2} as funções de potência associadas, respectivamente, à G_1 e G_2 . A função de potência do autômato $G_{1||2} = (Q_1 \times Q_2, _, _, _, _)$ é definida como:

$$f_{p_{1||2}}((q_1, q_2)) = f_{p_1}(q_1) + f_{p_2}(q_2)$$

onde $q_1 \in Q_1$ e $q_2 \in Q_2$.

Exemplo 2 Na Figura 2, o autômato G é o resultado da composição paralela dos autômatos G_1 e G_2 . Assim, o cálculo das potências nos estados de G é feito pela soma de duas funções de potência. Por exemplo no estado q_0, x_0 , a potência total será a soma de $f_{G_1}(q_0) + f_{G_2}(x_0)$.

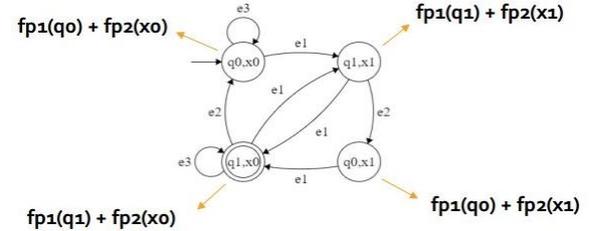


Figura 2: Representação do cálculo da potência nos estados do supervisor.

Especificações não executam tarefas, visto que são apenas restrições de comportamento, e dessa forma, para todas as especificações, a função de potência é sempre zero para todos os estados. Esse mesmo procedimento pode ser utilizado quando determinada planta não for de interesse no processo de otimização.

Para adicionar informações temporais à otimização, faz-se necessário avaliar o tempo até que determinado evento ocorra em um supervisor dada a sequência já executada até então.

Definição 5 Seja $S = (_, \Sigma, _, _, _)$ um supervisor. A função temporal, $f_T : \Sigma^* \times \Sigma \rightarrow \mathbb{R}^*$,

de S é definida como:

$$f_T(s, \sigma) = \begin{cases} t & \text{se } \delta(s, \sigma)! \\ \infty & \text{caso contrário} \end{cases}$$

onde, para um evento $\sigma \in \Sigma$ e uma sequência $s \in \mathcal{L}(S/G)$, t é o tempo até que o evento σ ocorra, dado que a sequência s acabou de ser executada.

Outra informação temporal importante é quanto tempo é necessário para executar uma sequência de eventos. Para isso, podemos expandir a função temporal para retornar o tempo total de execução de uma sequência como:

Definição 6 Seja $S = (_, \Sigma, _, _, _)$ um supervisor. A função temporal expandida, $f_T : \Sigma^* \rightarrow \mathbb{R}^*$, de S é definida como:

$$\begin{cases} f_T(\epsilon) & = 0 \\ f_T(s\sigma) & = f_T(s) + f_T(s, \sigma) \end{cases}$$

Dessa forma, precisa-se encontrar uma sequência da linguagem marcada do supervisor que minimize o tempo de produção (*makespan*) e o consumo energético respeitando limites de potência estabelecidos. Para calcular a energia, por sua vez, faz-se necessário definir a função de energia do sistema.

Definição 7 Seja $S = (Q, \Sigma, \delta, _, _)$ um supervisor. A função de energia, $f_E : Q \times \Sigma^* \rightarrow \mathbb{R}^*$, de S é definida como:

$$\begin{cases} f_E(\epsilon) & = 0 \\ f_E(s\sigma) & = f_E(s) + f_T(s, \sigma) \times f_p(\delta(q_0, s)) \end{cases}$$

onde $\sigma s \in \Sigma^*$.

Utilizando as funções temporal e de energia do supervisor, podemos utilizá-las como objetivos para a otimização. Uma vez definidas as funções de tempo e energia utiliza-se um simulador (Cassandras & Lafortune, 2008) para avaliação das sequências geradas. Nesta simulação o agendador de eventos é um elemento muito importante, por que é por meio dele que são realizadas as simulações do comportamento do sistema flexível de manufatura e é possível o cálculo da energia e do *makespan*. Ele armazena as informações de quando os eventos devem ocorrer e a atualização dos estados do sistema, sendo que a cada novo evento a evolução do sistema é atualizada.

3.1 Definições para otimização multiobjetivo

Nesta seção são apresentados alguns conceitos importantes para o entendimento do algoritmo multiobjetivo utilizado (Gomes et al., 2014).

Definição 8 *Dominância Pareto* - Sejam s e s^* soluções factíveis e f_k a k -ésima função objetivo, então:

- Se $f_k(s^*) \leq f_k(s)$ para todo $k = 1, 2, \dots, l$ e $f_k(s^*) < f_k(s)$ para algum k , a solução s é dominada por s^* .
- Se $f_j(s) < f_j(s^*)$ para algum j e $f_i(s) > f_i(s^*)$ para algum i , as soluções s e s^* são ditas não dominadas entre si, ou não comparáveis.

Definição 9 *Pareto Ótimo* - Uma solução factível s é chamada de Pareto-ótima se não há outra solução factível s^* tal que s^* domine s , isto é, uma solução s^* tal como $f_k(s^*) \leq f_k(s)$ para todo $k = 1, 2, \dots, l$ e $f_j(s^*) < f_j(s)$ para algum j .

O conjunto de todas as soluções Pareto Ótimas é denominado Conjunto Pareto Ótimo. A imagem deste conjunto no espaço de objetivos é chamada Fronteira Pareto Ótima. O critério de Dominância Pareto apresentado foi utilizado na abordagem proposta, para avaliar as soluções geradas a cada iteração e para determinar o conjunto de soluções não dominadas, a ser retornado.

3.2 Definição do Problema

Seja S o supervisor de um sistema de produção $G = \parallel_{k=1}^N G_k$, onde G_k é uma planta que compõe o sistema e seja f_T a função temporal e f_E a função de energia do sistema sob supervisão S/G . Seja n o número de eventos necessários para a produção de um lote de produtos e seja o universo de busca o conjunto $L = \{s \in \mathcal{L}_m(S/G) : n = |s| \wedge f_T(s) \neq \infty \wedge f_P(\delta(q_0, s')) < P_{max} \forall s' \in \{s\}\}$, sendo P_{max} a potência máxima admitida em um estado qualquer pelo qual a sequência passe.

Neste caso, o problema de planejamento da produção pode ser definido como um problema de otimização multiobjetivo na forma:

$$s^* = \underset{s \in L}{\operatorname{argmin}} (f_E(s), f_T(s))$$

onde s^* é uma das sequências que minimizam o tempo de produção e o consumo energético.

Neste trabalho, é proposto um algoritmo de busca local para obter uma estimativa das soluções Pareto-ótimas (Pinedo, 2012), que minimizem *makespan* e consumo energético (Bui et al., 2008). É importante notar que esses objetivos apenas são conflitantes quando existem múltiplas rotas de produção heterogêneas, caso contrário, não existiria variação no consumo energético, visto que o conjunto de dispositivos necessário para produzir um produto, bem como os tempos que esses dispositivos ficam em operação seriam sempre iguais.

4 Abordagem Proposta

Neste trabalho foi utilizada uma classe de métodos heurísticos baseada em Busca Local (Busca em Vizinhança). Esta abordagem inicia-se com uma solução completa do problema e constrói-se

a vizinhança desta solução que contém todas as soluções alcançáveis através de uma regra de movimento que modifica a solução inicial (Mladenović & Hansen, 1997). Apesar dos métodos heurísticos serem muito eficientes, não há garantia de se encontrar uma solução ótima, sendo seus resultados considerados ótimos locais. A versão utilizada foi proposta por (Geiger, 2008), a qual é conhecida como *Multi-Objective Variable Neighborhood Search* (MOVNS). Este método é apresentado no Algoritmo 1. Nele, a estrutura de vizinhança e a solução a ser explorada são selecionadas de forma aleatória a cada iteração (linhas de 4 a 14). Uma vez que uma solução s do conjunto de soluções não dominadas (D) é escolhida, esta solução é marcada como visitada. Dessa forma, ela não pode ser selecionada nas iterações seguintes, mesmo que ela permaneça no conjunto D (linhas 5 e 6). Caso todos os elementos do conjunto de soluções não dominadas já tenham sido visitadas e a condição de parada do algoritmo ainda não seja satisfeita, todas as soluções visitadas são desmarcadas (linha 12). Assim, a cada iteração do algoritmo, o conjunto D é atualizado com novas soluções vizinhas.

Algoritmo 1: Pseudocódigo do MOVNS implementado

```

1  $S \leftarrow$  Soluções iniciais;
2  $D \leftarrow$  Soluções não dominadas de  $S$ ;
3 enquanto Critério de Parada é Falso faça
4    $s \leftarrow$  solução não visitada de  $D$ ;
5   Marcar  $s$  como visitado;
6   Encontrar uma solução  $s'$  em  $N_k(s)$ ;
7   para 50 soluções  $s''$  em  $N_k(s')$  faça
8     Avaliar  $s''$ ;
9      $D \leftarrow$  Sol. não dom. de  $D \cup \{s''\}$ ;
10  fim
11  se Todo  $s$  em  $D$  foi visitado então
12    Desmarcar todos os visitados;
13  fim
14 fim

```

4.1 Estrutura de Vizinhança

A fim de evitar infactibilidades, foi necessário definir um mecanismo que garante encontrar vizinhos factíveis. As soluções vizinhas são obtidas a partir de trocas aleatórias nos eventos controláveis da sequência original. As trocas são realizadas em simulação tal que a cada evento controlável executado, existe a possibilidade desse evento ser trocado por outro evento controlável, até determinado limite. Tanto o evento removido quanto o evento colocado em seu lugar são armazenados em uma lista até que a troca possa ser desfeita.

Caso, devido às trocas, o sistema chegue a um bloqueio, ou seja, o evento previsto para ocorrer na sequência original não é factível e o limite de trocas foi alcançado, uma nova troca é feita para garantir a factibilidade da sequência. É importante notar que, trocando a posição apenas eventos controláveis, quando estes são factíveis, é sempre possível reconstruir uma sequência factível no

supervisor (Vilela & Pena, 2016).

4.2 Problema Teste

Como exemplo de aplicação, foi utilizado um sistema flexível de manufatura (SFM) (Ezpeleta et al., 1995), Figura 3, na qual as setas indicam o fluxo de peças no sistema. Nela pode-se observar a planta em questão, que é composta por três robôs (R_1, R_2, R_3), seis esteiras (E_1, E_2, \dots, E_6), quatro máquinas (M_1, M_2, M_3, M_4) e dez buffers unitários (B_1, B_2, \dots, B_{10}).

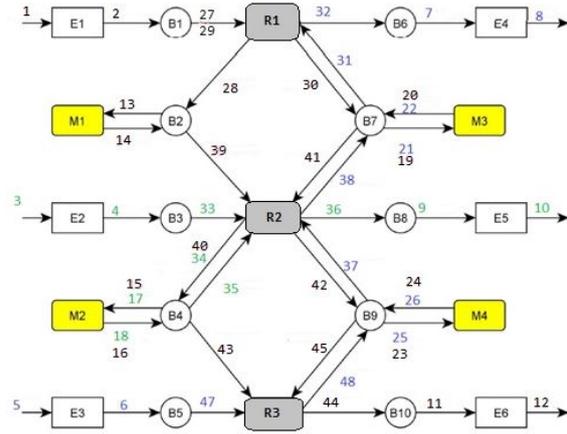


Figura 3: Sistema Flexível de Manufatura.

Uma característica especial desta planta é a sua capacidade de produzir três produtos diferentes (A, B, C) sendo que o produto A possui duas rotas alternativas A_1 e A_2 . Assim, podemos representar cada produto por meio de uma sequência de eventos controláveis (números ímpares), conforme:

- A_1 : 1, 29, 19, 41, 23, 45, 11;
- A_2 : 1, 27, 13, 39, 15, 43, 11;
- B : 3, 33, 17, 35, 9;
- C : 5, 47, 25, 37, 21, 31, 7.

Exemplo 3 Uma típica sequência de produção no sistema simplificado de manufatura seria a produção de 3 produtos ($A = B = C = 1$) na forma mais básica, que é a produção de um produto após o outro de forma sequencial. A seguir, as sequências são apresentadas:

- $Seq_1 = A_1 + B + C = [1, 29, 19, 41, 23, 45, 11] + [3, 33, 17, 35, 9] + [5, 47, 25, 37, 21, 31, 7]$
- $Seq_2 = A_2 + B + C = [1, 27, 13, 39, 15, 43, 11] + [3, 33, 17, 35, 9] + [5, 47, 25, 37, 21, 31, 7]$

Aqui, o símbolo “+” representa a concatenação das sequências.

Espera-se que o algoritmo de otimização encontre sequências, nas quais os eventos controláveis são misturados de forma a se obter a minimização do *makespan* e da energia consumida. Considerando a planta apresentada na Figura 3, para

se produzir 6 produtos ($A = B = C = 2$) utilizando o MOVNS seriam geradas seqüências de eventos controláveis conforme as soluções 1 e 2:

- **Solução 1:** 5, 3, 1, 27, 1, 33, 3, 47, 5, 13, 17, 25, 35, 33, 9, 17, 35, 39, 9, 27, 13, 15, 37, 47, 21, 25, 13, 39, 31, 7, 15, 11, 37, 21, 13, 11, 31, 7
- **Solução 2:** 1, 3, 5, 29, 1, 33, 3, 19, 17, 41, 35, 29, 23, 9, 33, 17, 19, 45, 41, 23, 35, 11, 9, 45, 47, 25, 11, 37, 5, 47, 21, 25, 31, 7, 37, 21, 31, 7

Cada uma dessas soluções representa uma seqüência de eventos distinta, que são resultado do processo de otimização. Essas seqüências são factíveis e legais, ou seja, ao serem implementadas num sistema real suas ações não resultariam em danos a planta.

Na Tabela 1 estão relacionados os intervalos de tempo dos eventos de acordo com as máquinas. Os eventos de início de operação são controláveis (números ímpares) e os eventos de finalização são não controláveis (números pares). Apesar de aparecerem na Figura 3 os eventos não controláveis são ignorados. Na Tabela 1 pode-se perceber que cada máquina possui tempos distintos, que são referentes a sua operação.

Tabela 1: Intervalos de Tempo entre Eventos

Planta	Início	Fim	Tempo(min)
E1	1	2	25
E2	3	4	25
E3	5	6	38
E4	7	8	21
E5	9	10	19
E6	11	12	24
M1	13	14	100
M2	15	16	100
	17	18	70
M3	19	20	75
	21	22	80
M4	23	24	200
	25	26	160
R1	27	28	15
	29	30	15
	31	32	15
R2	33	34	20
	35	36	20
	37	38	20
	39	40	20
	41	42	20
R3	43	44	30
	45	44	30
	47	48	30

Já na Tabela 2 estão relacionados os valores de potência associados aos estados dos dispositivos (máquinas) da planta. Nela cada dispositivo possui entre dois e três estados, sendo o primeiro estado “Ocioso”, no qual a máquina tem potência zero, e nos demais estados “Trabalhando” ela tem potência referente ao evento que a levou àquele estado (o evento está entre parênteses). Assim, as esteiras $E1, E2, \dots, E6$ possuem apenas dois estados e as máquinas $M2, M3, M4$ têm três estados, que são referentes à sua operação.

Tabela 2: Potência nos estados dos dispositivos que compõem a planta

Dispositivo	Estado	Potência(Watts)
Esteira (E_i)	Ocioso	0
	Trabalhando	100
M1	Ocioso	0
	Trabalhando	500
M2	Ocioso	0
	Trabalhando (15)	500
	Trabalhando (17)	600
M3	Ocioso	0
	Trabalhando (19)	300
	Trabalhando (21)	400
M4	Ocioso	0
	Trabalhando (23)	350
	Trabalhando(25)	600
Robôs	Ocioso	0
	Trabalhando	200

5 Resultados

Realizou-se a síntese de um supervisor controlável e não bloqueante para o problema tratado, resultando em um autômato de 6.288.192 estados e 45.988.488 transições, cuja linguagem marcada é o espaço de busca do problema.

Uma seqüência arbitrária, que produza n_a produtos do tipo A, n_b produtos do tipo B e n_c produtos do tipo C é composta por $14n_a + 10n_b + 14n_c$ eventos, sendo 14, 10, 14 o número de eventos necessários para se produzir um produto A, B e C, respectivamente, e esse número é fixo.

Todos os teste apresentados foram realizado em um notebook com processador Intel Core I7-3537U 2.0 GHz e 8.0 GB de memória RAM. Além disso, foi utilizado a biblioteca UltraDES de (Martins et al., 2017) para computar o supervisor e na modelagem do problema.

Para cada problema foram realizadas 10 execuções e considerado um número de 1000 iterações do algoritmo. Ao final, as fronteiras Pareto aproximadas, foram escolhidas considerando aquela com o maior valor de hipervolume (Zitzler & Thiele, 1999). Além disso, como método de suporte a tomada de decisão foi aplicado o ELECTRE I (Collette & Siarry, 2013).

O primeiro problema abordado foi a produção de quantidades iguais nas três máquinas ($A = B = C = 100$), sem que houvesse limitação da potência máxima necessária durante o processo. Na Figura 4 é possível observar os dois conjuntos de soluções, um para valores de makespan menores que 460 horas e outro para valores acima. Uma investigação das seqüências de eventos presente no primeiro conjunto (makespan ≤ 460 horas) revelou a predominâncias de produtos feitos pela linha A_2 , que é a mais rápida e tem o maior consumo de energia. Já no segundo conjunto (makespan > 460 horas) inicialmente existe um equilíbrio entre a utilização das linhas A_1 e A_2 , mas à medida que o makespan aumenta, o mesmo é deslocado no sentido de privilegiar a linha de produção A_1 . O que resulta em um menor consumo de energia, mas, em contra-partida, há aumentos significati-

vos no tempo de produção. No encontro das linhas tracejadas está a solução sugerida pelo algoritmo de auxílio à tomada de decisão. O ELECTRE I foi capaz de encontrar uma solução que equilibra o uso das linhas A_1 e A_2 de forma atender os compromissos de minimização do consumo energético e makespan.

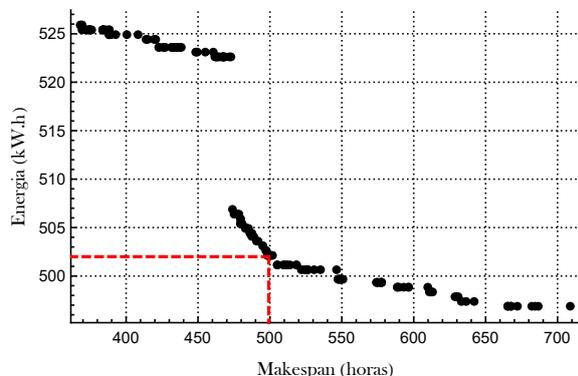


Figura 4: Problema 1: 300 produtos ($A = B = C = 100$) sem restrição de energia.

Quando o primeiro problema foi repetido, agora com limitação da potência máxima utilizada durante os estágios do processo (potência máxima foi fixada em $1500W$), percebe-se uma redução no número de soluções no conjunto Pareto estimado. Além disso, na Figura 5, pode se observar que as soluções com menor tempo de produção apenas surgem para valores de *makespan* maiores que 400 horas, enquanto no problema sem restrição (Figura 4) elas são observadas a partir de valores de *makespan* menores que 400 horas.

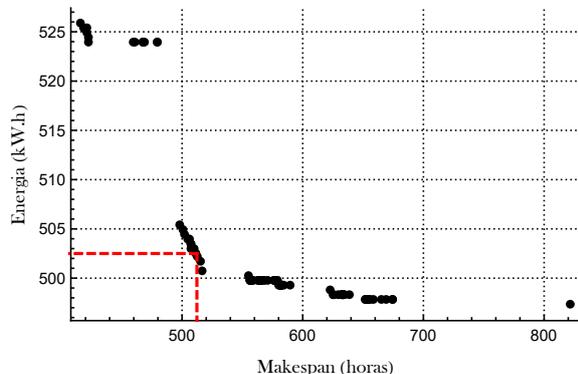


Figura 5: Problema 1: 300 produtos ($A = B = C = 100$) com restrição de energia.

Já no segundo problema a quantidade total produzida foi mantida, mas o número individual de peças foi alterado para $A = 150, B = C = 75$. Além disso, não foi limitada a potência máxima durante o processo. Uma característica desse problema foi deslocar o fluxo de produção em favor da maior quantidade do produto A . Na Figura 6 pode-se observar que houve uma redução do *makespan* máximo. Do mesmo modo, no intervalo de 370 a 460 horas do *makespan* existe uma predominância das soluções que priorizam o caminho

de menor tempo e maior consumo energético, que neste caso é a sequência A_2 .

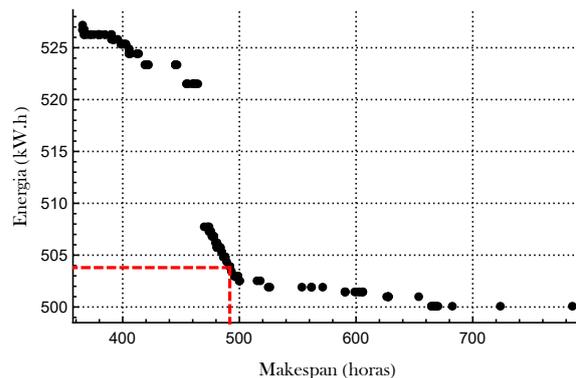


Figura 6: Problema 2: 300 produtos ($A = 150, B = C = 75$) sem restrição de energia.

Para o caso com restrição no segundo problema, Figura 7, observa-se que, novamente, há uma diminuição no número de soluções obtidas. É importante notar que a limitação de potência, também limita o número máximo de máquinas funcionando em paralelo, o que afeta diretamente o tempo de produção.

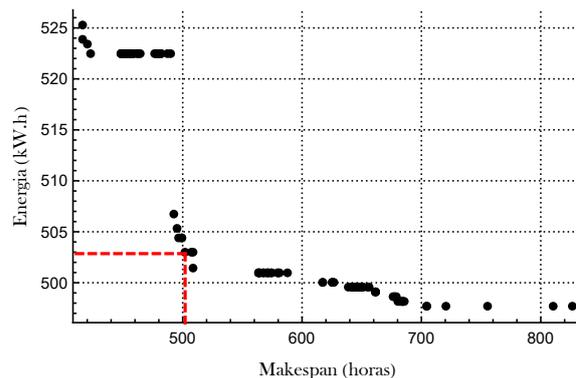


Figura 7: Problema 2: 300 produtos ($A = 150, B = C = 75$) com restrição de energia.

As soluções na fronteira Pareto estimada pelo MOVNS estão bem distribuída no espaço de soluções, porém em todos os casos apresentou um degrau que possivelmente está associado à estrutura do supervisor. É importante observar que sistemas sob supervisão são muito restritos e valores baixos de *makespan* usualmente estão associados ao uso de todos os recursos disponíveis, o que sugere a produção de produtos A não apenas pela rota mais rápida, mas por ambas as rotas em paralelo, e não apenas a que apresenta menor *makespan*.

Foram definidos pesos no algoritmo de tomada de decisão, o ELECTRE-I, que privilegiassem um menor consumo energético, mas sem prejudicar o *makespan*. Isto resultou, em todos os casos, na escolha de pontos após o degrau na fronteira Pareto estimada. Para uma melhor adequação da tomada de decisão, seria necessária uma

análise de custos, por exemplo, relacionando o lucro obtido com o tempo de produção e gasto de energia.

6 Conclusão

Este trabalho abordou o problema de planejamento de produção em um sistema com múltiplas rotas heterogêneas, no qual um produto é manufaturado por dois caminhos distintos. Esta tarefa foi tratada como um problema multiobjetivo cujos critérios de otimização foram *makespan* e consumo energético.

Para tanto foi desenvolvida uma modelagem em Sistemas a Eventos Discretos, na qual as informações relativas à potência demandada e ao consumo energético foram incorporadas aos estados dos autómatos do modelo. Esta abordagem não só permite a avaliação do consumo energético em cadeias do supervisor, mas também define quais estados não devem ser visitados a fim de se respeitar as restrições de potência.

Com a metodologia proposta foram obtidos resultados promissores no que tange a geração da fronteira de Pareto estimada.

Em trabalhos futuros, pretende-se aplicar outros algoritmos de otimização multiobjetivo em problemas similares, além de analisar problemas com um número maior de rotas de produção. Um outro ponto a ser investigado é a aplicação de restrição de potência na síntese de supervisores, de forma que seja possível garantir a controlabilidade e não bloqueio, mesmo em um contexto onde alguns estados não podem ser visitados por extrapolar o limite de potência.

Agradecimentos

O presente trabalho foi realizado com o apoio financeiro das agências de fomento brasileiras CAPES, CNPq e FAPEMIG.

Referências

- Abdeddaïm, Y., Asarin, E. & Maler, O. (2006). Scheduling with Timed Automata, *Theoretical Computer Science* **354**(2): 272 – 300.
- ANEEL, Agência Nacional de Energia Elétrica. (2010). Resolução normativa nº 414, de 9 de setembro de 2010. Art. 93.
- Bui, L. T., Alam, S., Bui, L. T. & Alam, S. (2008). *Multi-Objective Optimization in Computational Intelligence: Theory and Practice (Premier Reference Source)*, 1 edn, IGI Global, Hershey, PA, USA.
- Cassandras, C. & Lafortune, S. (2008). *Introduction to Discrete Event Systems*, 2 edn, Springer.
- Coelho, V. N., Souza, M. J. F., Coelho, I. M., Guimarães, F. G. & Lust, T. (2012). Algoritmos multiobjetivos para o problema de planejamento operacional de lava.
- Collette, Y. & Siarry, P. (2013). *Multiobjective optimization: principles and case studies*, Springer Science & Business Media.
- Costa, T., Paiva, L., Pena, P. & Takahashi, R. (2014). Abordagem cso: Extensão para tratar lotes maiores de produção, *Anais do XX Congresso Brasileiro de Automática, CBA* p. 26.
- Ehrgott, M. & Gandibleux, X. (2000). A survey and annotated bibliography of multiobjective combinatorial optimization, *OR-Spektrum* **22**(4): 425–460.
- Ezpeleta, J., Colom, J. M. & Martinez, J. (1995). A petri net based deadlock prevention policy for flexible manufacturing systems, *IEEE transactions on robotics and automation* **11**(2): 173–184.
- Garey, M. R. & Johnson, D. S. (1980). Computers and intractability: A guide to the theory of np -completeness, *Bulletin (New Series) of the American Mathematical Society* **3**(2): 898–904.
- Geiger, M. J. (2008). Randomised variable neighbourhood search for multi objective optimization, *arXiv preprint arXiv:0809.0271* pp. 1–8.
- Ghallab, M., Nau, D. & Traverso, P. (2004). *Automated Planning Theory and Practice*, Elsevier.
- Gomes, H. C., das Neves, F. d. A. & Souza, M. J. F. (2014). Multi-objective metaheuristic algorithms for the resource-constrained project scheduling problem with precedence relations, *Computers & Operations Research* **44**: 92–104.
- Herzig, A., de Menezes, M. V., de Barros, L. N. & Wassermann, R. (2014). On the revision of planning tasks, *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, pp. 435–440.
- López-Mellado, E., Villanueva-Paredes, N. & Almeyda-Canepa, H. (2005). Modelling of batch production systems using petri nets with dynamic tokens, *Mathematics and Computers in Simulation* **67**(6): 541 – 558.
- Martins, L. R. R., Alves, L. V. R. & Pena, P. N. (2017). Ultrades-a library for modeling, analysis and control of discrete event systems, *Proceedings of the 20th World Congress of the International Federation of Automatic Control* **50**(1): 5831–5836.
- Mladenović, N. & Hansen, P. (1997). Variable neighborhood search, *Computers & operations research* **24**(11): 1097–1100.
- Pena, P. N., Costa, T. A., Silva, R. S. & Takahashi, R. H. (2016). Control of flexible manufacturing systems under model uncertainty using supervisory control theory and evolutionary computation schedule synthesis, *Information Sciences* **329**: 491–502.
- Pinedo, M. L. (2012). *Scheduling: Theory, Algorithms, and Systems*, 3rd edn, Springer Publishing Company, Incorporated.
- Ramadge, P. J. & Wonham, W. M. (1989). The control of discrete event systems, *Proceedings of the IEEE* **77**(1): 81–98.
- Schrijver, A. (1986). *Theory of Linear and Integer Programming*, Vol. 20, John Wiley & Sons, Inc., New York, NY, USA.
- Su, R., Van Schuppen, J. H. & Rooda, J. E. (2012). The synthesis of time optimal supervisors by using heaps-of-pieces, *IEEE Transactions on Automatic Control* **57**(1): 105–118.
- Vilela, J. N. & Pena, P. N. (2016). Supervisor abstraction to deal with planning problems in manufacturing systems, *13th International Workshop on Discrete Event Systems, WODES 2016*, IEEE, pp. 117–122.
- Wang, W., Yuan, C. & Xiaobing, L. (2008). A fuzzy approach to multi-product mixed production job shop scheduling algorithm, *Proceedings - 5th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2008*, Vol. 1, pp. 95–99.
- Zitzler, E. & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach, *IEEE transactions on Evolutionary Computation* **3**(4): 257–271.