

CONTROLE SUPERVISÓRIO E OTIMIZAÇÃO: ABORDAGEM VNS-2OPT E ROBUSTEZ A PERTURBAÇÕES

TATIANA ALVES COSTA*, PATRÍCIA P. PENAF, RICARDO H. C. TAKAHASHI‡

*Programa de Pós-Graduação em Engenharia Elétrica - Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brazil
Universidade Federal de Ouro Preto - Departamento de Ciências Exatas e Aplicadas
João Monlevade, MG, Brasil

†Universidade Federal de Minas Gerais
LACSED (Laboratório de Análise e Controle de Sistemas a Eventos Discretos)
Departamento de Engenharia Eletrônica
Belo Horizonte, MG, Brasil

‡Universidade Federal de Minas Gerais
Departamento de Matemática
Belo Horizonte, Minas Gerais, Brasil

Emails: tatiana@ufop.br, ppena@ufmg.br, taka@mat.ufmg.br

Resumo— Este artigo apresenta novas contribuições em relação aos trabalhos reportados em Silva et al. (2011) e Costa et al. (2012), nos quais uma nova metodologia para tratar problemas de sequenciamento de tarefas em células flexíveis de manufatura é proposta, a metodologia *CSO* (Controle Supervisório e Otimização). Tal metodologia é definida como uma junção entre técnicas de controle supervisório de sistemas dinâmicos a eventos discretos (*CSSED*) e métodos de otimização. Especificamente no caso dos trabalhos citados, foram empregados respectivamente: (i) um algoritmo Clonal e (ii) um método híbrido, baseado em uma combinação entre um algoritmo de Colônia de Formigas e uma Busca Local 2-opt. Este trabalho propõe o uso de uma variante desta metodologia, que emprega um método baseado na metaheurística Variable Neighborhood Search (*VNS*) na otimização das sequências produzidas pelas técnicas *CSSED*. Além disso, neste trabalho é apresentada a característica da robustez inerente da metodologia *CSO* no tratamento de incertezas decorrentes da modelagem da planta ou de variações nos tempos de duração das tarefas. Os resultados dos testes realizados indicam que melhorias no tempo de operação da planta foram observadas ao se utilizar o *VNS* como método de otimização, bem como ilustram a robustez da metodologia no que diz respeito ao tratamento de incertezas.

Palavras-chave— Sequenciamento de Tarefas, Controle Supervisório, Otimização, Metaheurísticas.

1 Introdução

Problemas de sequenciamento de tarefas em células flexíveis de manufatura possuem grande importância industrial, uma vez que a resolução dos mesmos implica tanto na operação livre de falhas de plantas industriais de elevado custo quanto na utilização com máxima eficiência dos recursos disponíveis nessas plantas. Por essa razão tem havido, nos últimos anos, um grande esforço de pesquisa no sentido de se desenvolver ferramentas específicas e eficientes para o tratamento deste tipo de problema. Duas abordagens têm-se destacado na literatura, sendo elas: (i) técnicas baseadas em controle supervisório de sistemas dinâmicos a eventos discretos (*CSSED*) (Ramadge and Wonham, 1989) e (ii) métodos de otimização, tais como heurísticas e metaheurísticas (Zhang and Wu, 2011; Perez-Gonzalez and Framinan, 2010; Labadie et al., 2012). As abordagens *CSSED* não têm como objetivo realizar a busca pelo sequenciamento ótimo da produção. Por outro lado, incertezas na operação da planta podem limitar o uso de métodos de otimização em problemas de sequenciamento, uma vez que seria impraticável executar novamente o algoritmo de otimização, em tempo de execução, a fim de gerar

um novo sequenciamento que atenda às perturbações ocorridas durante a operação da planta.

Sendo assim, uma nova metodologia, proposta em Silva et al. (2011), aqui denotada por *CSO* (Controle Supervisório e Otimização), tem se mostrado bastante promissora, uma vez que é definida como uma combinação entre as duas abordagens mencionadas. A idéia básica é que um algoritmo de otimização seja responsável por realizar a busca pelo sequenciamento ótimo da produção, enquanto o controle supervisório tem por função codificar todas as restrições do problema, mantendo o sistema funcionando dentro de trajetórias admissíveis. Outro trabalho reportando o uso desta metodologia pode ser encontrado em Costa et al. (2012). Os dois trabalhos diferem entre si pelo método de otimização utilizado sendo, no primeiro caso, adotado um algoritmo Clonal e no segundo, um método híbrido, baseado em um algoritmo de Colônia de Formigas e uma Busca Local 2-opt. Uma característica importante dessa metodologia está relacionada ao fato de que, ao se manter o controle supervisório como um mecanismo de controle efetivo sobre a planta, entende-se que esta metodologia consiga atingir uma robustez mediante perturbações na operação da mesma. Isto

porque, uma vez definidos os controladores que implementam as restrições do problema, os mesmos codificam todas as soluções factíveis que podem ser usadas para lidar com qualquer mudança nos planos de produção que possam vir a ocorrer em tempo de execução.

A solução de Controle Supervisório (Ramadge and Wonham, 1989) tem como paradigma a não desabilitação de eventos não-controláveis (usualmente modelam as respostas da planta). Como consequência, os eventos não-controláveis uma vez habilitados, permanecem habilitados até que sejam executados. Portanto, se por alguma razão alguma operação é executada em um tempo diferente do planejado, o controle supervisório ainda terá um curso de ação que leva à execução correta de todas as tarefas.

O presente trabalho propõe uma variante da metodologia *CSO*, que utiliza um algoritmo baseado na metaheurística *VNS*, aqui denominada *VNS-CSO*, para buscar o melhor sequenciamento dentre os fornecidos pela abordagem *CSSED*. Os resultados obtidos demonstram que melhorias no tempo de operação da planta foram verificadas ao se utilizar o método *VNS* como técnica de otimização. A partir das soluções geradas pela otimização, analisa-se o comportamento destas soluções mediante incertezas nos tempos associados os eventos, comprovando-se a robustez da metodologia *CSO* ao se tratar distorções ocorridas nos tempos de duração das tarefas.

Este trabalho está estruturado como se segue. A seção 2 apresenta os conceitos necessários ao entendimento do trabalho. O *VNS-CSO* proposto é descrito na seção 3. Uma análise sobre a robustez da metodologia *CSO* no tratamento de incertezas é feita na seção 4. Os testes realizados e os resultados obtidos estão reportados na seção 5. A seção 6 encerra o trabalho com as conclusões sobre o mesmo.

2 Noções Preliminares

Esta seção introduz os conceitos preliminares necessários ao entendimento da metodologia *VNS-CSO* proposta.

2.1 Controle Supervisório

Foi utilizado no desenvolvimento desta metodologia o formalismo iniciado por Ramadge and Wonham (1989), no qual um Sistema a Eventos Discretos é modelado por um autômato $G = (Q, \Sigma, \delta, q_0, Q_m)$, onde Q é o conjunto de estados, Σ é o conjunto de eventos, δ é a função de transição, q_0 é o estado inicial e Q_m é o conjunto de estados marcados. Σ^* é o conjunto de todas as cadeias finitas de elementos em Σ , incluindo a cadeia vazia ϵ e uma linguagem é um subconjunto de Σ^* . O comportamento de um autômato G é modelado

por duas linguagens, sendo elas: $\mathcal{L}(G) \subseteq \Sigma^*$, definida como o conjunto de cadeias finitas geradas por G e $\mathcal{L}_m(G) \subseteq \mathcal{L}(G)$, representando o conjunto de cadeias que representam tarefas completas (ou seja, que terminam em estados marcados).

O conjunto de eventos usados para modelar a planta é dividido em dois subconjuntos: o conjunto de eventos controláveis e o conjunto de eventos não-controláveis, Σ_u . A ação do supervisor sobre a planta consiste em inibir a ocorrência de eventos controláveis de forma a alcançar o comportamento desejado. Às vezes não é possível implementar a linguagem desejada e a especificação (também chamada de linguagem desejada) é dita não-controlável. Formalmente, uma linguagem K é não-controlável em relação a uma dada linguagem $\mathcal{L}(G)$ se $\overline{K}\Sigma_u \cap \mathcal{L}(G) \not\subseteq \overline{K}$. Nestes casos, o supervisor implementa a máxima sublinguagem controlável da linguagem desejada, chamada $Sup\mathcal{C}(K, G)$, onde K , a linguagem desejada, é obtida pela composição da especificação genérica dada por E e $\mathcal{L}_m(G)$.

A projeção natural $P_i : \Sigma^* \rightarrow \Sigma_i^*$ é uma operação definida sobre dois conjuntos de eventos, Σ e Σ_i , onde $\Sigma_i \subseteq \Sigma$. Esta operação mapeia cadeias em Σ^* em cadeias em Σ_i^* apagando as ocorrências de eventos em $\Sigma \setminus \Sigma_i$. Esta operação é definida a seguir:

$$P_i(\epsilon) = \epsilon$$

$$P_i(s\sigma) = \begin{cases} P_i(s) & \text{se } s \in \Sigma^*, \sigma \notin \Sigma_i \\ P_i(s)\sigma & \text{se } s \in \Sigma^*, \sigma \in \Sigma_i. \end{cases}$$

O conceito de projeção natural pode ser estendido para linguagens regulares como $P_i(L) = \{u_i \in \Sigma_i^* \mid u_i = P_i(u) \text{ para algum } u \in L\}$.

2.2 Descrição da Metodologia *CSO*

O problema de otimização tratado consiste na minimização do tempo total gasto na produção de lotes de produtos, a serem manufaturados em uma célula flexível de manufatura. A metodologia *CSO* foi ilustrada utilizando-se o exemplo do sistema flexível de manufatura (*SFM*), citado pela primeira vez em Queiroz et al. (2005) e apresentado na Figura 1. Este *SFM* está apto à manufatura de dois tipos de produtos: produto A, composto por um bloco com um pino cônico no topo e produto B, formado por um bloco com um pino cilíndrico pintado. O *SFM* é composto por oito equipamentos, sendo eles: três esteiras (*C1*, *C2* e *C3*), uma fresa, um torno, um robô, uma máquina de pintura (*PD*) e uma máquina de montagem (*AM*). Estes equipamentos são conectados por meio de buffers B_i , $i = 1, \dots, 8$, cada qual capaz de armazenar uma unidade por vez. As especificações que devem ser atendidas por um controlador *CSSED* para este problema são: (i) evitar overflow e underflow de peças nos buffers; (ii) permitir operação simultânea da fresa e do torno; (iii) assegurar que ambos os produtos A e B sejam produzidos,

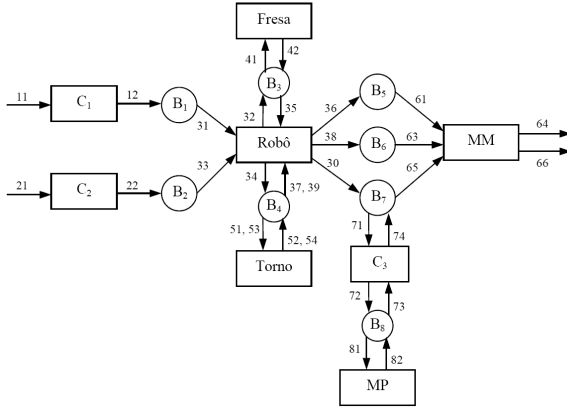


Figura 1: Sistema Flexível de Manufatura

finalizando-se todos os respectivos passos necessários à produção dos mesmos.

O controle supervisorio para o *SFM* foi obtido utilizando-se técnicas de Controle Modular Local, conforme descrito em Pena et al. (2010). Os supervisores atuam por meio apenas da desabilitação de eventos controláveis e garantem, de forma minimamente restritiva, o atendimento das especificações. A partir daí, a metodologia *CSO* consiste em aplicar o processo de otimização sobre o sistema sob o controle exercido pelos supervisores.

Para ser consistente com a teoria de controle utilizada, estabelece-se que a solução da otimização tem que prover uma sequência em que apenas os eventos controláveis devem ser manipulados. Nada pode ser especificado em relação aos eventos não-controláveis. Portanto, uma solução para o problema é representada por vetores de eventos controláveis, conforme descrito a seguir: (i) A produção de uma base e um pino para um produto A é representada, respectivamente, pelos vetores $a_0 = 11-31-41-35-61$ e $a_1 = 21-33-51-37-63$; (ii) Os vetores $b_0 = 11-31-41-35-61$ e $b_1 = 21-33-53-39-71-81-73-65$ representam, nesta ordem, a produção da base e do pino para um produto do tipo B. Para garantir o sequenciamento dentro das subsequências (produção de bases e pinos) e evitar que um grande número de soluções infactíveis sejam geradas durante o processo de otimização, uma solução para o problema tratado foi formulada como uma sequência de a_0 , a_1 , b_0 e b_1 . Para exemplificar, considere a sequência de eventos 11-31-41-35-61-21-33-51-37-63-11-31-41-35-61-21-33-53-39-71-81-73-65, que representa a produção de um produto do tipo A e outro do tipo B. De acordo com a codificação adotada, tal sequência é representada como: $a_0a_0a_0a_0a_0a_1a_1a_1a_1a_1b_0b_0b_0b_0b_1b_1b_1b_1b_1b_1b_1$. Maiores detalhes sobre essa representação podem ser encontrados em Silva et al. (2011) e Costa et al. (2012).

A simulação do *SFM* é feita através de um software computacional, aqui denotado por *Simu-*

lador, que tem seu funcionamento norteado pelo proposto em Cassandras and Lafortune (2008). Para realizar essa simulação, 1 unidade de tempo (u.t.) foi associada a todos os eventos reportados como não-controláveis e, diferentes valores (de 15 u.t. a 38 u.t.), foram associados aos eventos controláveis. Esses valores são apresentados na Tabela 1. A interpretação a ser dada ao tempo associado a um evento é a de modelar o tempo de permanência mínimo no estado alcançado pelo evento. Por exemplo, seja o evento 11, que liga a esteira C_1 . O tempo de 25 u.t. é interpretado como o tempo em que esta esteira permanece no estado ligado.

Tabela 1: Tempos de Simulação.

Subsistemas	Evento	u.t.	Evento	u.t.
C_1	11	25	12	1
C_2	21	25	22	1
Robô	31	21	32	1
	33	19	34	1
	35	16	36	1
	37	24	38	1
	39	20	30	1
Fresa	41	30	42	1
	51	38	52	1
Torno	53	32	54	1
	61	15	64	1
MM	63	25	66	1
	65	25		
C_3	71	25	72	1
	73	25	74	1
MP	81	24	82	1

O software de simulação é utilizado para avaliar uma solução e retornar seu valor de função objetivo. Sendo assim, toda solução gerada pelo algoritmo de otimização é repassada ao *Simulador*, para ser avaliada por ele. Caso se trate de uma solução factível, o *Simulador* calcula e retorna seu custo final, obtido com base nos tempos de execução dos eventos, apresentados na Tabela 1. No caso de não ser factível, a sequência é tratada visando-se restaurar sua factibilidade.

Os passos a seguir resumem a metodologia *CSO*: (i) O comportamento do *SFM* deve ser modelado e o controle supervisorio para o mesmo obtido; (ii) Um software que simula o comportamento do sistema (planta + supervisores) deve ser implementado; (iii) Um algoritmo de otimização, que interage com o software de simulação, deve ser executado para se obter uma solução possivelmente ótima para o problema, conforme descrito em Silva et al. (2011) e Costa et al. (2012).

Por meio desta metodologia, pode-se determinar um sequenciamento de tarefas para um lote pré-determinado de diferentes produtos. De posse desse sequenciamento, a operação da planta deve ser realizada considerando-se o controle supervisorio como um controle principal, enquanto a

sequência de operações definida pelo algoritmo de otimização deve ser usada como uma referência de alto nível a ser seguida pelo controle supervisorio.

3 CSO-VNS

Este trabalho propõe a utilização de um algoritmo baseado na metaheurística *VNS*, para otimizar as sequências produzidas pela abordagem *CSSED*. O *VNS* (Maldenovic and Hansen, 1997) é um método de busca local que explora o espaço de soluções através de mudanças sistemáticas de estruturas de vizinhança, explorando vizinhanças gradativamente mais “distantes” da solução corrente e focando a busca em torno de uma nova solução somente se um movimento de melhora é executado.

No *VNS* implementado, uma solução inicial para o problema é gerada de forma autônoma pelo software de simulação. Os vizinhos de uma solução são gerados utilizando-se dois tipos de movimentos, a saber: (i) Movimento de troca, no qual dois elementos de uma solução são selecionados de forma aleatória e têm suas posições trocadas; (ii) Movimento de troca de blocos, onde dois blocos de eventos, de mesmo tamanho, selecionados aleatoriamente dentro de uma solução, têm suas posições trocadas. Seis estruturas de vizinhanças são obtidas variando-se a intensidade de cada um desses movimentos: (i) $N^{Troca-20}(s)$, $N^{Troca-30}(s)$ e $N^{Troca-40}(s)$, definidas como sendo vizinhanças formadas pelo conjunto de vizinhos obtidos a partir de uma solução s , aplicando-se sobre a mesma um número n de movimentos de troca; (ii) $N^{TrocaBloco-20}(s)$, $N^{TrocaBloco-30}(s)$ e $N^{TrocaBloco-40}(s)$, representando os vizinhos obtidos a partir de uma solução s ao se aplicar sobre ela um número n de movimentos de troca de blocos. Nas estruturas citadas, o valor de n é definido como sendo, respectivamente, 20%, 30% e 40% do tamanho da solução considerada.

Todo vizinho gerado pelo algoritmo *VNS* é passado ao *Simulador*, para ser avaliado por ele. No caso de um vizinho infactível, o *Simulador* trata a infactibilidade conforme descrito em Oliveira et al. (2013).

O *VNS* conta ainda com um procedimento de busca local 2-opt (Helsgaun, 2000), que é executado sobre a solução corrente, a cada iteração do método. Os passos que resumem o procedimento da busca 2-opt são listados a seguir: (i) Uma posição dentro da sequência é fixada; (ii) Mantendo-se fixa esta posição, todas as possibilidades de mudanças são analisadas, gerando-se todos os possíveis vizinhos da solução corrente; (iii) Caso o valor de função objetivo do melhor vizinho seja menor que o valor de função objetivo da corrente, este torna-se corrente e o processo é reiniciado; (iv) O algoritmo termina quando o valor de função objetivo do melhor vizinho é pior que o valor

da solução corrente.

O *VNS* implementado tem seu processamento interrompido quando um total de 20 iterações consecutivas sem melhora no valor de função objetivo da melhor solução global é atingido.

4 Tratamento de Incertezas

Uma solução completa para o problema de sequenciamento de tarefas em sistemas de manufatura consiste numa sequência de eventos que, se executada naquela ordem, minimiza uma função objetivo. No caso deste trabalho, focado em sistemas de manufatura, pretende-se minimizar o tempo de produção de um lote de produtos. Esta sequência completa é composta de eventos controláveis (o comando de ligar cada uma das operações) e não-controláveis (neste caso, a sinalização do final de cada operação). No entanto, a tentativa de ordenar os eventos não-controláveis contraria a filosofia do controle supervisorio, que baseia-se na execução da lei de controle pela desabilitação de eventos controláveis.

Ao utilizar o controle supervisorio como base para a otimização, a filosofia do controle supervisorio deve ser respeitada. Desta forma, não se considera parte da solução a ordenação dos eventos não-controláveis. Este fato justifica a utilização apenas de eventos controláveis no processo de otimização conforme mostrado na Seção 2.2. Esta solução implementa, na prática, todas as soluções que intercalam eventos não-controláveis entre os eventos da sequência.

Formalmente, seja uma solução $w \in \Sigma_c^*$. Na prática, esta solução implementa um conjunto de cadeias em $W \subseteq \Sigma^*$ tal que: $W = \{w' \in \Sigma^* | w' \in P^{-1}w \cap \mathcal{L}_m(G)\}$, onde $P : \Sigma^* \rightarrow \Sigma_c^*$ é a projeção de cadeias em Σ para cadeias em Σ_c . Pode-se afirmar que $P(W) = w$, ou seja, todas as cadeias de W projetam em w . Isto significa que para cada solução w existe um conjunto W de cadeias formadas por eventos controláveis e não-controláveis, que são implementadas.

Esta forma de tratar o problema é bastante vantajosa quando comparada aos métodos de otimização, tais como as heurísticas e metaheurísticas. Isto porque, qualquer alteração ocorrida nos tempos de duração de cada operação (representadas pelos tempos associados aos eventos controláveis) podem causar inversões na ordem de execução dos eventos não-controláveis. Estas inversões geram mudanças na sequência efetivamente executada, mas de acordo com essa abordagem, sem sair de W , ou seja, qualquer que seja a sequência executada ela ainda possui a mesma sequência de eventos controláveis, portanto continua sendo modelada por w . Já para o caso dos métodos de otimização, qualquer mudança nos tempos de duração das operações implica em uma nova execução do algoritmo de otimização, de forma a gerar

um novo sequenciamento, que atenda às modificações ocorridas.

5 Resultados

Esta seção apresenta os resultados dos testes realizados afim de se analisar o desempenho da metodologia proposta. Para tal, foi utilizado um conjunto de 14 instâncias, que proporcionam a produção de até quatro produtos dos tipos A e B definidos.

Primeiramente, os testes realizados tiveram por objetivo comparar os resultados obtidos pelo *VNS* proposto aos alcançados pelo *Clonal* (Silva et al., 2011) e pelo algoritmo de Colônia de Formigas (Costa et al., 2012). Os resultados obtidos são apresentados na Tabela 2, na qual: (i) a primeira coluna indica o número de produtos A e B a serem manufaturados; (ii) as quatro colunas abaixo do *Clonal-CSO*, do *CF+2opt-CSO* e do *VNS-CSO* representam, nessa ordem, o melhor resultado, a média, o desvio padrão (%) e a média do número de avaliações de função objetivo das soluções encontradas por cada uma das abordagens consideradas, em um total de 30 execuções das mesmas.

Os dados apresentados na Tabela 2 demonstram que apesar de realizar um maior número de avaliações de função objetivo e, por causa disso ser um pouco mais lento, o *VNS* proposto consegue chegar aos melhores resultados para todas as instâncias consideradas. Ainda para o caso das instâncias (2,1), (2,2) e (1,3), o uso desta metaheurística implica uma redução no tempo de operação da planta, em relação às duas outras abordagens analisadas. O segundo melhor desempenho é verificado para a abordagem que utiliza o algoritmo de Colônia de Formigas, sendo a pior performance observada para o algoritmo *Clonal*. Ainda de acordo com esta tabela, pode-se constatar que o *VNS* mostra-se mais estável, uma vez que apresenta menores valores médios e desvios para as soluções encontradas.

Em um segundo momento, novos testes foram realizados, visando-se agora testar a robustez da metodologia proposta a perturbações ocorridas na duração das operações. Foram consideradas 4, das 14 instâncias definidas anteriormente, sendo elas: (1,2), (2,1), (0,3) e (3,0). As sequências otimizadas obtidas para estes lotes foram analisadas pelo *Simulador* considerando-se diferentes configurações para os tempos de duração dos eventos definidos na Tabela 1.

A cada simulação, alguns eventos foram selecionados de forma aleatória e tiveram seu tempo de duração levemente perturbados, assumindo novos valores definidos dentro de um intervalo pré-determinado. Três diferentes intervalos foram considerados, nos quais os tempos dos eventos podem variar aleatoriamente, com distribuição uniforme, em intervalos de $\pm 10\%$, $\pm 20\%$ e $\pm 30\%$ do

valor original apresentado na Tabela 1. Foram realizadas 30 simulações para cada instância, em cada um dos intervalos definidos.

Os resultados obtidos podem ser observados na Tabela 3, onde: (i) as duas primeiras colunas indicam respectivamente, o número de produtos A e B a serem manufaturados e o valor de função objetivo da melhor solução encontrada para o problema; (ii) as demais colunas indicam o valor mínimo, o valor máximo, a média e o desvio padrão (%) das soluções encontradas variando-se o tempo dos eventos dentro dos intervalos 10, 20 e 30%, nessa ordem.

Em todas as simulações realizadas, apesar das alterações nos tempos de execução dos eventos, a sequência obtida pela metodologia proposta permaneceu factível, comprovando que alterações na ordem dos eventos não-controláveis não tornam a solução obtida pela otimização inviável. A degradação no tempo final de produção não é demasiadamente significativa, quando consideradas pequenas perturbações nos tempos de execução dos eventos, como pode ser verificado analisando-se os valores médios das soluções obtidas variando-se os tempos dentro dos intervalos 10 e 20%. Obviamente esta degradação tende a aumentar à medida que as distorções nos tempos tornam-se muito maiores, conforme pode ser notado nos dados obtidos para o intervalo 30%.

6 Conclusões

Neste trabalho foi proposta uma variante da metodologia *CSO* (Silva et al., 2011), denominada *VNS-CSO*, a qual utiliza um algoritmo baseado na metaheurística *VNS*, para otimizar os sequenciamentos obtidos pela abordagem *CSSSED*. Os resultados obtidos demonstraram que melhorias no tempo de operação da planta foram obtidas ao se utilizar o método *VNS* como técnica de otimização. Além disso, a robustez inerente à metodologia *CSO*, no que se refere ao tratamento de distorções nos tempos de duração das tarefas, também foi apresentada neste trabalho, como uma vantagem adicional decorrente da técnica proposta.

Agradecimentos

O presente trabalho foi realizado com o apoio financeiro das agências CAPES, CNPq, e FAPESP, e do programa Procad-CAPES.

Referências

- Cassandras, C. G. and Lafortune, S. (2008). *Introduction to Discrete Event Systems*, Kluwer Academic Publishers.
- Costa, T. A., Oliveira, A. C., Pena, P. N. and Takahashi, R. H. C. (2012). An ant sys-

Tabela 2: Resultados para as metodologias analisadas.

(nA,nB)	Clonal-CSO				CF+2opt-CSO				VNS-CSO			
	Min	Med	Dev	Aval	Min	Med	Dev	Aval	Min	Med	Dev	Aval
(0,1)	204	204,03	0,18	750	204	204,06	0,25	1458	204	204,03	0,18	780
(1,0)	160	160,00	0,00	790	160	160,00	0,00	1129	160	160,00	0,00	430
(1,1)	250	258,20	4,83	5360	250	250,00	0,00	9171	250	250,20	0,40	34680
(0,2)	303	304,70	1,31	8070	303	303,00	0,00	12103	303	303,06	0,25	27140
(2,0)	254	254,86	0,73	6150	254	254,40	0,49	7808	254	254,03	0,18	12160
(1,2)	354	370,56	9,54	42220	349	349,00	0,00	38087	349	349,16	0,37	171300
(2,1)	336	356,76	8,71	31750	335	337,83	0,64	32011	334	334,53	1,27	129270
(2,2)	438	472,10	12,92	116040	430	430,00	0,00	91504	414	416,13	5,19	499180
(0,3)	406	409,10	1,47	10170	402	405,00	0,00	36654	402	402,13	0,34	134900
(3,0)	340	342,30	1,20	9040	339	339,13	0,43	24840	339	339,13	0,34	55290
(1,3)	475	516,43	16,01	43300	452	451,00	0,00	88869	448	464,96	8,55	441710
(3,1)	429	461,86	17,24	44970	419	431,33	12,55	74624	419	421,10	3,31	361810
(0,4)	507	510,56	1,59	17290	501	503,00	0,00	85627	501	501,00	0,30	419590
(4,0)	428	431,13	1,30	12540	424	424,06	0,25	59890	424	424,00	0,00	141790

Tabela 3: Robustez da Metodologia Proposta.

(nA,nB)	FO Orig.	10%				20%				30%			
		Min.	Max.	Med.	Des.	Min.	Max.	Med.	Des.	Min.	Max.	Med.	Des.
(1,2)	349	345	364	354,40	3,98	342	371	357,86	7,00	328	395	358,26	13,25
(2,1)	334	334	348	340,56	4,12	327	358	341,56	7,84	320	366	343,26	11,60
(0,3)	402	396	429	408,90	6,98	389	432	406,76	11,22	391	436	410,53	13,88
(3,0)	339	335	354	343,76	4,85	326	366	343,96	9,84	324	384	350,50	15,377

tem algorithm for task scheduling in a flexible manufacturing cell with supervisory control, *XIX Congresso Brasileiro de Automática*, pp. 2515–2522.

Helsgaun, K. (2000). An effective implementation of the Lin-Kernighan traveling salesman heuristic, *European Journal of Operational Research* **126**: 106–130.

Labadie, N., Mansini, R., Melechovsky, J. and Calvo, R. W. (2012). The team orienteering problem with time windows: An LP-based granular Variable Neighborhood Search, *European Journal of Operational Research* **220**(1): 15–27.

Maldenovic, N. and Hansen, P. (1997). Variable neighborhood search, *Computers and Operations Research* **24**: 1097–1100.

Oliveira, A. C., Costa, T. A., Pena, P. N. and Takahashi, R. H. C. (2013). Clonal Selection Algorithms for Task Scheduling in a Flexible Manufacturing Cell with Supervisory Control, *Proceedings of the IEEE Congress on Evolutionary Control - Cancun - Mexico*, pp. 982–988.

Pena, P. N., Cunha, A. E. C., Cury, J. E. R. and Lafortune, S. (2010). Metodologia e ferramenta de apoio ao teste de não conflito no

controle modular de sistemas a eventos discretos, *Revista Controle e Automação* **21**: 58–68.

Perez-Gonzalez, P. and Framinan, J. M. (2010). Setting a common due date in a constrained flowshop: A variable neighbourhood search approach, *Computers and Operations Research* **37**(10): 1740–1748.

Queiroz, M., Cury, J. and Wonham, W. (2005). Multitasking supervisory control of discrete-event systems, *Discrete Event Dynamic Systems: Theory and Applications* **15**: 375–395.

Ramadge, P. and Wonham, W. (1989). The Control of Discrete Event Systems, *Proceedings IEEE, Special Issue on Discrete Event Dynamic Systems*, Vol. 77, pp. 81–98.

Silva, R. S., Oliveira, A. C., Pena, P. N. and Takahashi, R. H. C. (2011). Algoritmo clonal para job shop scheduling com controle supervisorio, *X Simpósio Brasileiro de Automação Inteligente*, pp. 1376–1381.

Zhang, R. and Wu, C. (2011). A simulated annealing algorithm based on block properties for the job shop scheduling problem with total weighted tardiness objective, *Computers and Operations Research* **38**: 854–867.