Joaquim José da Cunha Júnior

Programação de Produção em Máquinas CNC para o curto prazo

Belo Horizonte
2012

Joaquim José da Cunha Júnior

Programação de Produção em Máquinas CNC para o curto prazo

Dissertação apresentada ao Departamento de Engenharia de Produção da Escola de Engenharia da Universidade Federal de Minas Gerais, para a obtenção de Título de Mestre em Engenharia de Produção, na Área de Concentração produção e logística.

Orientador: Prof. Dr. Mauricio Cardoso de Souza

Belo Horizonte
2012

Aos meus pais, Ju, Ana e Clara.

Agradecimentos

Agradeço, em primeiro lugar, ao professor e orientador Mauricio de Souza que, além de acreditar e apostar no nosso trabalho, teve toda a paciência e sabedoria inestimáveis e motivadoras sem as quais nada teria acontecido. Ao professor Horacio Yanasse pelas discussões e importantes colaborações que tanto contribuiram e engrandeceram nossa pesquisa. A toda a equipe (professores, demais servidores e alunos) do Programa de Pós-graduação em Engenharia de Produção da UFMG pela acolhida, amizade e contribuição, em especial aos professores Ricardo Camargo e Gilberto Miranda. Aos meus familiares, em especial, pelo apoio e paciência nos momentos de dificuldade quando tudo parecia impossível. Vocês fizeram possível.

RESUMO

Esse trabalho trata da aplicação prática de sequenciamento de tarefas em máquinas flexíveis. A empresa em questão é um indústria metal-mecânica onde uma máquina flexível é utilizada para processar peças de precisão. A máquina é assistida por um *software* CAM *(Computer Aided Manufacturing)* e executa trocas automáticas de ferramentas durante o processamento de um determinado *job*. A máquina precisa ser interrompida e há intervenção do operador quando um *job* requer uma ferramenta que não está disponível na caixa de ferramentas. Assim, o problema de sequenciamento dos *jobs* e minimização de trocas de ferrametas para minimizar custos com horas extras e com atrasos nos prazos de entrega consiste em decidir os *jobs* a serem processados em cada dia do horizonte de planejamento, junto com sua ordem de processamento, dado que as ferramentas necessárias para iniciar um *job* devem estar numa caixa de ferramentas com capacidade limitada. Modelos de programação inteira são propostos para tratar o problema e os restultados numéricos demosntram o potencial do uso de um modelo de otimização no contexto real e incentivam o desenvolvimento de métodos para solucioná-lo.

Abstract

This study treat a practical application of scheduling jobs on flexible machines. The company in analysis is a metal working industry where a single flexible machine is used to manufacture precision parts. The machine is assisted by computer aided manufacturing softwares and allows automatic changing when processing a jobs. The machine has to be stopped and human intervention occurs when a job requires at least one tool which is not already in the tool magazine. Thus, the job scheduling and tool switching problem to minimize overtime and weighted tardiness costs consists of deciding the jobs to be processed in each day of the horizon, along with their processing order, given that the required tools to start a job must be on the limited capacity tool magazine. Integer programming models for the problem are proposed and numerical results on real data instances, and significant gains were obtained with respect to the firm's practice. Results show the potential to the use of the optimization model in the real-life setting and encourage to the development of methods for solving it.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Problemas de sequenciamento de produção são amplamente abordados na literatura. Sua especificidade permite um vasto horizonte de pesquisa de acordo com os mais diversos processos de produção (Pinedo (2008), Błazewics et al. (1996)). Nesse estudo, serão abordados problemas de sequenciamento em uma máquina que utiliza um conjunto de ferramentas agrupado de formas distintas para executar um conjunto de tarefas, ou *jobs*, como usualmente denominado na literatura. Um *job* é um conjunto de operações realizadas de forma repetitiva por uma máquina. Assim, no caso de um processo de puncionamento, um *job* envolve a execução de um conjunto de estampos em chapas metálicas. Geralmente, esse conjunto de estampos é realizado de maneira repetitiva sobre várias chapas. O processamento de todas essas chapas (iguais) representa um *job*. Normalmente, máquinas controladas por comando numérico computadorizado (CNC), como tornos, puncionadeiras, fresas e etc. utilizam um conjunto de ferramentas necessárias à realização de suas atividades. Essas ferramentas são acomodadas em uma caixa de ferramentas com capacidade limitada. Durante o processamento das tarefas, as trocas entre as ferramentas que se encontram na caixa de ferramentas ocorre de forma automatizada. É comum que capacidade de alocação de ferramentas nessas máquinas seja inferior ao número total de ferramentas requeridas para o processamento de um conjunto de tarefas e, consequentemente, é necessário que ocorram trocas de ferramentas. O *setup* nesse tipo de máquina, geralmente, envolve sua parada para remoção da caixa de ferramentas (*multi-tool*), substituição das ferramentas presentes na caixa de ferramentas através da retirada das ferramentas que não serão utilizadas (sempre que for necessário liberar espaço na caixa de ferramentas) e inserção das ferramentas necessárias e recolocação da caixa de ferramentas na máquina. As atividades de *setup* são frequentes ao longo da produção para que a execução de todas as atividades seja possível e, consequentemente, a máquina precisa ficar parada para que ocorram as trocas de ferramentas. Na medida em que a diferença entre o número total de ferramentas requeridas e a capacidade da caixa de ferramentas da máquina aumenta, o número de trocas necessárias também cresce e, consequentemente, os tempos gastos com operações de *setup*.

O desenvolvimento cada vez mais veloz de novas teconologias, o apoio estatal através de linhas de financiamento e fomento ao desenvolvimento industrial e o mercado globalizado, cada vez mais competitivo, tem aumentado consideravelmente o acesso de pequenas e médias empresas a maquinários flexíveis com tecnologia CNC. O salto tecnológico proposto por máquinas desse tipo pode proporcionar altos ganhos em qualidade e produtividade e torna as empresas mais competitvas. Contudo, tais equipamentos incorrem em custos mais elevados (mão de obra, manutenção, ferramentas e etc.) e, consequentemente, requerem técnicas de sequenciamento menos intuitivas. Na medida em que as máquinas se tornam cada vez mais flexíveis (através do desenvolvimento e uso de novas ferramentas) e processam um número maior de atividades, a complexidade do sequenciamento focado na redução de trocas e os custos aumenta.

O tempo de processamento das atividades em máquinas CNC é difícil de ser melhorado. Essas máquinas, normalmente, têm baixa interação humana e trabalham programadas por softwares CAM (*computer aided manufacturing*). Esses fatores proporcionam bons níveis de produtividade. Por outro lado, normalmente, as atividades de *setup* podem ser complexas e requerem tempos elevados para sua execução. As ferramentas apresentam custos elevados e as atividades de *setup* além de perturbar o processo com a parada da máquina, geram riscos a sua integridade, pois, uma troca de ferramenta má sucedida pode gerar danos que, normalmente, são irreversíveis.

O desenvolvimento de metodologias de sequenciamento com foco em redução dos tempos de *setup* pode ser uma forma extremamente conveniente para aumentar a produtividade dos equipamentos CNC. O tempo requerido para a execução do *setup* está diretamente ligado a quantidade de ferramentas que deverão ser substituídas. A redução do tempo de *setup* possibilita a redução de custos de produção.

O foco desse trabalho é desenvolver modelos matemáticos e métodos capazes de resolver o problema de sequenciamento de tarefas numa máquina CNC visando minimizar custos relacionados ao uso de horas extras e atrasos no término das tarefas. A implementação e os testes computacionais com um pacote de otimização sobre instâncias retiradas do ambiente fabril são fatores centrais na análise da viabilidade do estudo e desempenho dos modelos e métodos desenvolvidos.

Todo o texto que segue é baseado em algumas premissas que são padrão na literatura e que devem ser consideradas:

(i) em cada espaço da caixa de ferramentas cabe exatamente uma ferramenta;

(ii) não existem trocas simultâneas de ferramentas;

(iii) o tempo de troca é considerado o mesmo para qualquer que seja a ferramenta;

(iv) qualquer ferramenta pode ser colocada em qualquer posição da caixa de ferramentas sem que isso afete o tempo de processamento das atividades.

Após essa introdução, no Capítulo 2 será apresentado o exame de literatura acerca de problemas de minimização de trocas de ferramentas. No Capítulo 3 formulações matemáticas e suas principais características são apresentadas. No Capítulo 4 resultados obtidos a partir da implementação dos modelos discutidos no Capítulo 3 são demonstrados. Por fim, são feitas conclusões e propostas de oportunidades para a continuidade dos estudos são apresentados no Capítulo 5.

Capítulo 2

Exame da Literatura

O problema de minimização de trocas de ferramentas (PMTF), ao nosso conhecimento, começou a ser estudado de forma sistemática na literatura a partir dos trabalhos de Bard (1988) e Tang e Denardo (1988a). Embora Tang e Denardo (1988a) tenham afirmado empiricamente que o PMTF é NP-difícil, foram Crama et al. (1994) que formalmente, através de dois teoremas, apresentaram a prova. No primeiro teorema, comprovou-se que uma versão restrita do PMTF possui analogia precisa ao problema de permutação de matrizes que é comprovadamente NP-difícil através de vários estudos presentes na literatura. No segundo teorema, comprovou-se que o caso geral do PMTF considerando equipamentos com capacidade fixa e maior ou igual a dois corresponde ao problema do caixeiro viajante. Esse último também é conhecidamente NP-difícil.

Em contrapartida, uma vez que a sequência de jobs a ser processada está definida, é possível minimizar o número de trocas de ferramentas em tempo polinomial. Esse resultado pode ser obtido adotando como critério manter na caixa de ferramentas as ferramentas que serão utilizadas primeiro. Assim, sempre que é necessário liberar espaço na caixa de ferramentas retiram-se primeiramente as ferramentas que serão utilizadas por último. Essa política *Keep Tool Needed Soonest (KTNS)* ou “mantenha na caixa de ferramentas aquelas que serão utilizadas mais cedo”, garante que a minimização do número de trocas para uma dada sequência de *jobs* é obtida em tempo polinomial segundo Bard (1988) e Tang e Denardo (1988a).

Bard (1988) formulou o PMTF como um problema de programação não linear inteira. O autor demonstrou que é possível reduzir o tamanho do problema através do pré-processamento dos dados de entrada. Se um determinado *job* requer exatamente as mesmas ferramentas ou um conjunto menor de ferramentas iguais ao de outro *job* o processamento sequencial de ambos (um *job* imediatamente após o outro) não requererá trocas. Assim, tais *jobs* podem ser agrupados em um único *job*, o que reduz o tamanho do problema. A técnica de solução do problema não linear utilizada por Bard (1988) foi a relaxação Lagrangeana. Essa técnica aliada ao *branch-and-bound* é comprovadamente eficiente na solução de vários problemas de sequenciamento. Através da relaxação de uma das restrições do modelo, o problema de sequenciamento de *jobs* é desacoplado do problema de trocas de ferramentas.

Para otimização local do problema o autor propôs o uso de uma heurística de relaxação baseada em informação dual capaz de encontrar soluções de boa qualidade rapidamente. Essa heurística é baseada em quatro passos. No primeiro deles o problema primal relaxado é obtido através de relaxação Lagrangeana. Ele é resolvido para se obter sequências de *jobs* candidatas. Para cada sequência é aplicada a política *KTNS* para se obter a número mínimo de trocas equivalente. A sequência capaz de produzir o menor valor na função objetivo do problema original é então escolhida. No segundo passo, a vizinhança da sequência escolhida é examinada. Nessa etapa a posição de alguns *jobs* é alterada e, através da aplicação da política *KTNS*, verifica-se o número de trocas requeridas para as novas sequências. No passo três, as soluções obtidas no passo dois são confrontadas com a sequência que gera o menor valor da função objetivo até o momento. Caso alguma delas apresente valor de função objetivo menor que a sequência atual, a mesma é escolhida e retorna-se ao passo dois, caso todas as demais soluções sejam analisadas e nenhuma tenha resultado melhor que a solução atual avança-se para o passo 4. O quarto passo interrompe a heurística, pois, a obteve-se uma solução local.

O autor avaliou a eficiência da sua heurística através de testes computacionais realizados em instâncias geradas aleatoriamente. A aplicação do pré-processamento dos dados, quando os *jobs* são agrupados conforme critério descrito anteriormenente, reduziu consideravalmente o tempo gasto para a execução completa da heurística. O resultados obtidos demonstraram a capacidade da heurística de fornecer resultados de boa qualidade em tempo computacional relativamente baixo, o que suporta a aplicação prática do método.

Tang e Denardo (1988a) propuseram um modelo de programação inteira para minimizar o número de trocas de ferramentas sobre o qual dedicaremos uma seção a seguir. Adicionalmente, Tang e Denardo (1988b) propuseram a aplicação de um esquema *branch-and-bound* baseado em um problema não linear capaz de produzir resultado ótimo para o problema de minimização de número de instantes de troca de ferramentas. Primeiramente os autores descrevem o problema de agrupamento de *jobs* e demonstram que ele é equivalente ao problema de sequenciamento de *jobs*. O problema de agrupamento de *jobs* em conjuntos que requeiram no máximo C ferramentas (onde C é a capacidade da caixa de ferramentas) é um caso particular que foi utilizado como base para a criação do esquema de *branch-and-bound* que combina dois procedimentos: a busca de limitantes superiores e de limitantes inferiores a paritr do valor ótimo da função objetivo do problema de agrupamento de *jobs*. Para avaliar a efeciência do procedimento de *branch-and-bound* ao problema, os autores o implementaram computacionalmente em instâncias geradas aleatoriamente e obtiveram resultados encorajadores para instâncias com números razoável de *jobs*.

No contexto de gerenciamento de memórias de computador, Belady (1966) estudou um problema semelhante no qual tarefas a serem processadas por um computador necessitam de informações disponíveis em sua memória, entretanto, comumente, o espaço é insuficiente e informações precisam ficar armazenadas em dispositivos de memória secundária para serem inseridas imediatamente antes de seu processamento. Para processar as tarefas, as informações da memória precisam ser trocadas, analogamente ao que ocorre no PMTF.

Crama et al. (1994) observaram que o PMTF pode ser transformado num problema equivalente ao problema do caixeiro viajante se a capacidade da caixa de ferramentas for fixa e maior ou igual a dois ($C \geq 2$). Considerando um grafo $G = (V,E,lb)$, onde V é o conjunto de *jobs*, E é o conjunto formado por todos os pares de *jobs* e lb são as arestas do grafo onde $lb_{(i,j)}$ é o comprimento do arco i,j dado por uma subestimativa do número trocas necessárias para processar o *job* j imediatamente após o *job* i. Essa estimativa é dada por: $lb_{(i,j)} = max(|T_i \cup T_j| - C, 0)$, onde T_k é o conjunto de ferramentas necessárias para processar o *job* $k(k = 1, 2, ..., V)$. Como o grafo G utiliza estimativas para o número de trocas requeridas no processamento sucessivo de dois *jobs* não pode-se afirmar que a solução obtida é ótima para o PMTF. Essa solução será um limitante superior para o problema.

Assim, cada caminho obtido para o problema do caixeiro viajante sobre o grafo G é uma solução para o problema de minimização de trocas de ferramentas. Como o problema é NP-difícil é muito dispendioso solucioná-lo de maneira ótima. Sendo assim, Crama et al. (1994) propuseram diversas heurísticas para a solução do problema. Quatro delas, bastante conhecidas na literatura, baseadas em algoritmos do caixeiro viajante são: *shortest edge heuristic*, heurística da vizinhança mais próxima, heurística da inserção mais distante e algoritmos de *branch and bound*.

Adicionalmente os autores apresentaram uma heurística de minimização de blocos, que também faz associação do problema de minimização de trocas de ferramentas ao problema do caixeiro viajante. Apresentaram também heurísticas gulosas aliadas a política *KTNS*, uma heurística de intervalo e estratégias $2 - opt$, nas quais, dada uma sequência de *jobs* a posição de dois deles é trocada com objetivo de produzir uma solução melhor. Outra estratégia adotada no tratamento do problema foi a heurística denominada *Load-and-optimize* onde, dada uma sequência de *jobs*, aplica-se a política *KTNS* para se obter em tempo polinomial o menor número de trocas daquela sequência. O resultado obtido é tratado como uma nova instância que é submetida ao método exato. Obviamente, qualquer solução com valor menor que o resultado inicial representa em uma sequência melhor.

Os experimentos computacionais apresentados por Crama et al. (1994) foram realizados em instâncias aleatórias dividias em 16 tipos de acordo com os tamanhos das matrizes *job*-ferramenta e pela capacidade da caixa de ferramentas. Os autores concluiram que, avaliando a qualidade das soluções obtidas com cada heurística, a heurística gulosa e a $2 - opt$ apresentaram os melhores resultados.

Ainda tratando o problema através da analogia ao problema do caixeiro viajante Hertz et al. (1998) propuseram novas estimativas ao número de trocas de ferramentas entre dois *jobs*. Essa estimativa, como já descrita, representa a distância entre dois *jobs* representados como nós em um grafo.

Essas estimativas foram utilizadas para decompor o problema de minimização de trocas de ferramentas em um problema de caixeiro viajante para então determinar uma sequência de realização dos *jobs*. A política *KTNS* é então aplicada para fornecer o número mínimo de trocas para aquela sequência. Os autores apresentaram três heurísticas para o PMTF fazendo uso de heurísticas refinadas para o problema do caixeiro viajante desenvolvidas por Gendreau et al. (1992).

Os testes computacionais foram realizados em instâncias de dezesseis tipos (como proposto por Crama et al. (1994)) e apontaram para uma relação diretamente proporcional entre a qualidade das soluções e o tempo computacional requerido. Insuma, de acordo com o contexto industrial em questão o melhor método pode variar de acordo com a escolha do critério de avaliação como qualidade da solução ou tempo computacional.

AlFawzan e AlSultan (2002) exploraram um algoritmo de busca tabu que, dada uma sequência de *jobs*, blocos de *jobs* são realocados na sequência a partir da análise das vizinhanças de trocas de *jobs*. Os autores empregaram memórias de curto e longo prazos para penalizar movimentos que tenham sido realizados mais frequentemente ao longo da busca.

Yanasse (2007) desenvolveu um procedimento de cálculo de limites inferiores para o PMTF baseado na resolução de subproblemas. O autor demonstrou que o valor ótimo do número mínimo de trocas de ferramentas para um subconjunto estrito do total de *jobs* a serem processados é menor ou igual ao valor ótimo do número mínimo de trocas necessário ao processamento de todos os *jobs*. Assim o procedimento seleciona um subconjunto de *jobs* e resolve de maneira ótima o subproblema que lhe é associado. Em seguida, é acrescentado um job por vez ao subproblema enquanto o esforço computacional para tal for considerado aceitável, ver Yanasse (2007). Yanasse et al. (2008) propuseram que para o cálculo de limites superiores ao valor ótimo do PMTF sejam mantidas apenas as melhores (s = 1, 2, 3, ...) sequências parciais em cada nível da árvore num esquema de enumeração.

Yanasse et al. (2009) apresentaram um esquema enumerativo para definir o sequenciamento ótimo de um conjunto de *jobs*. O método, inicialmente, define um subconjunto de *jobs* escolhidos combinando o menor número de *jobs* que juntos utilizam todas as ferramentas. O número de trocas necessárias para realizar esses *jobs* é um limitante inferior para o problema (ver Yanasse et al. (2009)). A partir desse subconjunto, os *jobs* são inseridos ao subconjunto nas posições do sequenciamento de forma e reduzir o número de trocas necessárias. Primeiramente os *jobs* que não requerem trocas para serem processados e assim por diante. Ao final, quando todos os *jobs* foram inseridos, temos uma solução viável ou limitante superior, cujo valor corresponde ao limitante inferior do problema, logo, tem-se uma solução ótima.

Cunha e Souza (2009) aplicaram uma heurísitca gulosa a um caso real. Baseando-se na heurística GRASP(ver Resende e Ribeiro (2003)) um procedimento de sequenciamento foi definido para obter soluções rápidas e de boa qualidade.

Nesse procedimento, a cada iteração a configuração da máquina (ou seja, as ferramentas disponíveis na caixa de ferramentas) é comparada com a lista de atividades a serem sequenciadas. Assim, é possível conhecer a quantidade de trocas necessárias para realizar cada uma das atividades. As tarefas são ordenadas de acordo com o número de trocas necessárias e, de acordo com o *gap* pré-estabelecido, entram em uma lista de candidatas a próxima tarefa no sequenciamento. Ou seja, todas as tarefas que requerem um número de trocas menor ou igual ao mínimo de trocas mais o *gap* são candidatas. Um dos *jobs* presentes na lista de candidatos é escolhido aleatoriamente para ser o próximo no sequenciamento. A caixa de ferramentas é então atualizada para processar esse *job*. Se todas as posições da caixa de ferramentas estiverem ocupadas, será necessário remover alguma ferramenta para liberar espaço. O critério de escolha da ferramenta que será retirada é uma adaptação do *KTNS - Keep tool needed soonestd* (ver Bard (1988), Tang e Denardo (1988a) e Crama et al. (1994)), no qual as ferramentas são mantidas na máquina de acordo com a quantidade de vezes que ainda serão utilizadas no sequenciamento das atividades futuras (aquelas menos utilizadas são as primeiras a serem retiradas).

Os resultados obtidos comprovam o potencial de ganho no desenvolvimento de metodologias para atacar o problema e podem servir como limitantes superiores para métodos exatos.

Cunha e Souza (2010) e Cunha et al. (2011) publicaram alguns dos modelos desenvolvidos durante o trabalho de mestrado. Inicialmente foi apresentado um modelo de minimização de custos relativos a horas extras. Esse modelo foi elaborado com base no modelo apresentado por Tang e Denardo (1988a) e acrescentou a possibilidade de discretizar o horizonte de planejamento em períodos (dias). Em cada um desses períodos existe uma carga possível de horas extras. Nesse modelo todos o *jobs* devem ser realizados o que dificulta a obtenção de soluções viáveis iniciais. Nos modelos propostos posteriormente não há essa restrição e *jobs* podem não ser alocados na sequência de produção dentro do horizonte de planejamento. O que significa assumir o custo da não produzí-los naquele planejamento.

Os testes computacionais foram realizados com instâncias obtidas no ambiente fabril e apresentaram resultados bons quando comparados aos resultados obtidos com as metodologias intuitivas utilizadas na prática.

2.1. Modelo proposto por Tang e Denardo (1988a)

Tang e Denardo (1988a) propuseram um modelo de programação inteira baseado em variáveis binárias que relacionam *jobs* assim como ferramentas carregadas na máquina a posições na ordem de sequenciamento.

Dados um conjunto J de *jobs*, um conjunto M de ferramentas ($|M| > C$, onde C é a capacidade da caixa de ferramentas), variáveis binárias x_n^j são iguais a 1 se o *job* $j \in J$ é processado na posição $n \in N = \{1,\ldots,n\}$. Variáveis binárias w_n^m são iguais a 1 se a ferramenta $m \in M$ está presente na caixa de ferramentas enquanto o job está sendo processado na posição n. Por fim, as variáveis binárias p_n^m são iguais a 1 se a ferramenta está presente na caixa de ferramentas na posição n, mas, não estava na posição $n-1$, logo, $p_n^m = 1$ representa uma troca de ferramentas. A matriz binária a_j^m indica com o valor 1 se ferramenta m é requerida para o processamento do *job* j e com o valor zero caso contrário. A formulação é a seguinte:

$$Minimizar \quad \sum_{m \in M} \sum_{n \in N} p_n^m \tag{2.1}$$

$$Sujeito\ a:$$

$$\sum_{j \in J} x_n^j = 1 \quad \forall n \in N, \tag{2.2}$$

$$\sum_{n \in N} x_n^j = 1 \quad \forall j \in J, \tag{2.3}$$

$$x_n^j a_j^m \leq w_n^m \qquad \forall n \in N, \forall m \in M, \forall j \in J, \tag{2.4}$$

$$\sum_{m \in M} w_n^m \leq C \qquad \forall n \in N, \tag{2.5}$$

$$w_n^m - w_{n-1}^m \leq p_n^m \quad \forall n \in N/\{1\}, \forall m \in M, \tag{2.6}$$

$$x_n^j \in \{0,1\} \qquad \forall n \in N, \forall j \in J, \tag{2.7}$$

$$w_n^m \in \{0,1\} \qquad \forall n \in N, \forall m \in M, \tag{2.8}$$

$$p_n^m \in \{0,1\} \qquad \forall n \in N, \forall m \in M, \tag{2.9}$$

A função objetivo direciona a solução do problema no sentido de minimizar o total de trocas de ferramentas. A restrições (2.2) e (2.3) são restrições de atribuição que garantem que cada *job* será executado exatamente uma vez (restrição (2.2)) e que, em cada posição do sequenciamento, apenas um *job* será alocado (restrição (2.3)). As restrições (2.4) garantem que o job só será processado se as ferramentas requeridas para o seu processamento estiverem disponíveis na caixa de ferramentas. A capacidade da caixa de ferramentas é garantida pelas restrições (2.5) que impedem que um número de ferramentas maior que a capacidade seja alocado na caixa. As restrições (2.6) obrigam que, se um ferramenta está na caixa na posição n e não estava na posição $n-1$, é preciso ser inserida através de uma troca. As restrições (2.7), (2.8) e (2.9) definem que as variáveis x_n^j, w_n^m e p_n^m são binárias.

A formulação é simples do ponto de vista de implementação computacional, porém é fraca em termos de relaxação linear, visto que esta é sempre zero a menos que os jobs sejam fixados. Após alguns testes utilizando o modelo proposto ((2.1) a(2.9)) em algumas instâncias de tamanho pequeno , os experimentos conduzidos por Tang e Denardo (1988a) não obtiveram bons resultados. Diante disso os autores apresentaram uma heurística baseada no fato de que o problema de trocas de ferramentas é um caso especial do problema de sequenciamento de *jobs*.

Essa heurística consiste em três etapas. A primeira obtem uma solução de boa qualidade através do caminho Hamiltoniano em um grafo onde os *jobs* representam os nós e o tamanho das arestas é dado pelo número de trocas entre dois jobs quando processados em sequência. A partir da sequência obtida no primeiro passo, o segundo passo consite em empregar a política *KTNS* para obter o menor número de trocas para aquela sequência. O terceiro passo caracteriza-se pela aplicação de perturbações a sequência obtida no intuito de obter resultados melhores.

Esse procedimento foi testado computacionalmente e forneceu respostas de boa qualidade para instâncias com 10, 20 e 30 jobs. Para instâncias com números de *jobs* maiores ou iguais a 40 o procedimento se mostrou menos eficiente, o que pode ter sido causado pelo fato de o tempo computacional estipulado não ser suficiente para a obtenção de bons resultados.

2.2. Modelo proposto por Laporte et al. (2003)

Alternativamente, Laporte et al. (2003) propuseram uma formulação baseada no modelo de Dantzig et al. (1954) para o problema do caixeiro viajante. Esse modelo utiliza-se do fato de que o problema em questão corresponde ao problema do caixeiro viajante se um nó fictício representando o início e o fim das operações for introduzido.

Introduzindo variáveis binárias para garantir que uma ferramenta necessária ao processamento de um *job* esteja carregada na máquina ao início de sua execução. O nó fictício que caracteriza o início e o término das operações é representado por zero, variáveis binárias x^i_j são iguais a 1 se o *job* i e processado imediatamente antes do *job* j na sequência (i,$j \in J \cup \{0\}$). Variáveis binárias y^i_m são iguais a 1 se a ferramentas m está na caixa de ferramentas durante o processamento do job i ($i \in J, m \in M$). O conjunto de ferramentas requeridas pelo *job* i é dado por M_i. Por fim, as variáveis binárias z^i_m são iguais a 1 se a ferramenta m é inserida na caixa de ferramentas para processar o job i, ou seja, representa que houve uma troca. O modelo é o seguinte:

$$Minimizar \quad \sum_{i \in J} \sum_{m \in M} z^i_m, \tag{2.10}$$

$Sujeito\ a:$

$$\sum_{j \in J \cup \{0\}/\{i\}} x^i_j = 1 \qquad \forall i \in J \cup \{0\}, \tag{2.11}$$

$$\sum_{i \in J \cup \{0\}/\{i\}} x^i_j = 1 \qquad \forall j \in J \cup \{0\}, \tag{2.12}$$

$$\sum_{i,j \in S} x^i_j \leq |S| - 1 \quad \forall S \subset J \cup \{0\}, 2 \leq |S| \leq |J| - 1, \tag{2.13}$$

$$\sum_{m \in M} y_m^i \leq C \qquad \forall j \in J, \tag{2.14}$$

$$x_j^i + y_m^j - y_m^i \leq z_m^j + 1 \qquad \forall i \in J \cup \{0\}, \forall j \in J, \forall m \in M, \tag{2.15}$$

$$y_m^i = 1 \qquad \forall i \in J, m \in M_i, \tag{2.16}$$

$$z_m^i = 0 \qquad \forall i \in J, \forall m \in M/M_i, \tag{2.17}$$

$$x_j^i \in \{0,1\} \qquad \forall i,j \in J \cup \{0\}, \tag{2.18}$$

$$y_m^i \in \{0,1\} \qquad \forall i \in J, m \in M, \tag{2.19}$$

$$z_m^i \in \{0,1\} \qquad \forall i \in J, m \in M, \tag{2.20}$$

A exemplo do modelo proposto por Tang e Denardo (1988a), a função objetivo minimiza o total de trocas de ferramentas, porém, através da soma das variáveis z_m^i. A restrições de atribuição (2.11) e (2.12) garantem que cada *job* é precedido exatamente por um outro *job* (inclusive o *job* fictício denotado por zero e que representa a origem) e que cada *job* sucedido por exatamente um *job* (inclusive o *job* fictício de destino). As restrições (2.13) previnem a formação de sub-rotas na solução do problema. Uma sub-rota ocorre quando o número total de jobs no incluídos na sequência é menor que o número total de jobs a serem sequenciados. Se a soma de x_j^i é igual a $|S|$ para todas as conexões (I,J) em um subtour, então é possível eliminar soluções desse tipo impondo que o somatório de x_j^i em todas as conexões de um sub-conjunto de S seja menor ou igual a $|S| - 1$.

As restrições (2.14) garantem que a capacidade da caixa de ferramentas é respeitada. As restrições (2.15) obrigam que, caso uma ferramenta m nessária ao processamento de um *job* j ($y_m^j = 1$) não esteja na caixa de ferramentas durante o processamento de i ($y_m^i = 0$) e o *job* i precede o *job* j no sequenciamento ($x_j^i = 1$), então ela precisa ser inserida ($z_m^i = 1$), ou seja, uma troca é realizada. As restrições (2.16) obrigam que as ferramentas necessárias a fabricação do *job* i (M_i) esteja na caixa de ferramentas enquanto o mesmo é processado. As restrições (2.17) garantem que uma troca só é realizada quando um ferramenta é necessária para realizar o próximo job. As restrições (2.18), (2.19) e (2.20) definem que as variáveis x_j^i, y_m^i e z_m^i são binárias.

A formulação de Laporte et al. (2003) domina em termos de relaxação linear a de Tang e Denardo (1988a), mas, a sua implementação computacional é mais complexa em razão da restrição (2.13) de eliminação de sub-rotas (ver Laporte (1992)). Adicionalmente Laporte et al. (2003) propuseram dois algoritmos para a solução do problema, o primeiro utilizando *branch-and-cut* e o segundo utilzando *branch-and-bound*. O algoritimo de *branch-and-cut* proposto por Laporte et al. (2003) utiliza como solução viável no nó raíz da árvore de busca uma solução obtida a partir da aplicação da heurística proposta por Gendreau et al. (1992). O algoritmo de *branch-and-bound* utiliza como limitante superior inicial uma solução obtida atravéz da aplicação de uma heurística gulosa onde o primeiro *job* é aquele requer o maior número de ferramentas (o critério de desempate é o número de ferramentas mais frequentemente utilizadas), a partir daí os próximos jobs são escolhidos de acordo com o número de ferramentas comuns com o último *job* escolhido. Dada uma sequência, o limitante inferior é obtido através da aplicação da política *KTNS*. As heurísticas foram testadas em instâncias geradas segundo os mesmos critérios propostos por Crama et al. (1994). A heurística baseada no algoritmo *branch-and-cut* apresentou bons resultados, porém, em instâncias muito pequenas (até 9 *jobs*). Para instâncias com número de *jobs* maior ou igual a 10 os resultados não foram satisfatórios. A heurística baseada em *branch-and-bound* obteve resultados mais encorajadores e foi capaz de resolver instâncias maiores com até 25 *jobs*. Os resultados demonstraram que o modelo proposto por Laporte et al. (2003) embora domine o modelo de Tang e Denardo (1988a) não é aplicável para solução de instâncias grandes. A heurística baseada em *branch-and-bound* apresentou resultados satisfatórios em instâncias de tamanhos que podem ser verificados no contexto prático o que confirma seu potencial de aplicação num contexto real.

Capítulo 3

Problema de troca de ferramenta no horizonte de programação da produção

3.1. Contexto industrial

Com base no modelo proposto por Tang e Denardo (1988a) alguns modelos são propostos com foco em adequar sua aplicação ao problema real estudado. O cenário utilizado para o desenvolvimento dos modelos e avaliação de sua eficácia é uma indústria de pequeno porte do ramo de mecânica de precisão. Nessa empresa, o processo produtivo central é o puncionamento realizado por uma máquina flexível com tecnologia CNC. A Puncionadeira CNC pode ser vista na Figura 3.1. As características do processo estão alinhadas àquelas descritas para o problema de minimização de trocas de ferramentas, uma vez que, a capacidade da caixa de ferramentas da máquina geralmente é inferior ao total de ferramentas requeridas para processar um conjunto de jobs. As figuras 3.2 e 3.3, respectivamente, mostram a caixa de ferramentas e algumas ferramentas utilizadas no processo. A máquina em questão é uma puncionadeira CNC (Figura 3.1) que possui capacidade para 10 ferramentas (Figura 3.2).

A programação de produção ocorre a partir de planejamentos semanais atualizados diariamente. Assim, é importante que o modelo empregado no sequenciamento seja capaz de gerar soluções que abranjam mais de um período. Normalmente, horas extras são empregadas para viabilizar a fabrica-

Figura 3.1: Puncionadeira CNC

ção de todas as atividades solicitadas pelos clientes. Eventualmente, pedidos deixam de ser atendidos (em prazo), pois, não é possível realizar horas extras, ou, não compensa, pois, o custo excede o custo de não atendimento do pedido. É importante notar também que, normalmente, *jobs* compõem produtos. Assim, o não atendimento do prazo de entrega de um determinado *job* pode comprometer a a entrega de um produto que é também composto por outros *jobs*.

Além da opção de realizar horas-extras, é importante notar que produtos que não são processados dentro do prazo acarretam em custos por descumprimento do prazo acordado com o cliente. Esse custo pode ser ocasionado por multas contratuais ou até mesmo pela insatisfação de um determinado cliente. De acordo com o estudo dos custos de *backlog* de cada grupo de clientes é possível avaliar a viabilidade de realizar horas-extras para realizar determinada atividade, pois, eventualmente, pode ser mais vantajoso assumir um custo de *backlog*.

Os modelos propostos a seguir incorporam essas características. Assim, caso não haja capacidade para processar todas as tarefas no prazo e dentro do horário regular de trabalho, a solução ótima será um compromisso entre realizar horas extras ou deixar de atender prazos de entrega (assumindo

Figura 3.2: Caixa de ferramentas

Figura 3.3: Ferramentas

custos de atraso) e agrupar *jobs* em produtos. Nesses modelos, o número de trocas de ferramentas gera impacto indireto aos custos da solução uma vez que não fazem parte da função objetivo.

Os *jobs* possuem datas de entrega dentro do horizonte de planejamento. O não atendimento dessas datas gera custos associados a multas contratuais e insatisfação dos clientes. A escolha do que produzir em um dia, dentro do horizonte de planejamento, é baseada na relação que existe entre reduzir custos com horas extras, produzir um conjunto de *jobs* que compõem um produto e minimizar custos relativos ao não atendimento de prazos. Na prática é comum que os *jobs* sejam produzidos por ordem de serviço gerada e, consequentemente, agrupados por produtos. Porém, essa tática pode gerar altos tempos de *setup* uma vez que essa ordenação não leva em conta a diversidade do conjunto de ferramentas demandado por cada *job*.

O tempo necessário para realizar uma operação de *setup* entre dois *jobs* pode ser dividido em duas parcelas. A primeira delas configura o tempo fixo da atividade de *setup* que independe da quantidade de ferramentas que serão trocadas. Ná prática esse é o tempo de remover a caixa de ferramentas, posicioná-la para as atividades de trocas e colocá-la novamente na máquina. A segunda parcela é variável e depende da quantidade de ferramentas trocadas durante aquela operação. A troca das ferramentas é um processo uniforme para qualquer que seja a ferramenta.

Se as ferramentas são sequenciadas sem observar os custos de *setup*, é comum que ocorram várias paradas com poucas trocas de ferramentas. Na prática, observa-se que, praticamente a cada *job* que é processado, ocorre uma parada para troca de ferramentas. Essas paradas são focadas na produção apenas do próximo *job*, assim, muitas vezes perde-se a oportunidade de usar espaços disponiveis (ferramentas que estão na caixa, mas, que não serão utilizadas na sequência) e trocar ferramentas que não serão utilizadas imediatamente, mas, que serão utilizadas em seguida, muitas vezes reduzindo uma atividade de *setup*.

3.2. Modelagem Matemática

Nesse trabalho adequamos e estendemos o modelo de Tang e Denardo (1988a) para explorar o sequenciamento em um horizonte de planejamento com mais de um período. Assim, o planejamento é mais adequado ao processo em questão. Geralmente, as atividades de planejamento envolvem 6 períodos e são atualizadas diariamente. De acordo com o sequenciamento definido podem surgir custos pela utilização de horas extras (que podem, inclusive, ter custos distintos de acordo com o dia da semana e o horário) e de atrasos (multas contratuais e insatisfação de clientes).

Introduzimos ao problema algumas características do contexto industrial:

(i) horas extras e atraso por *job*;

(ii) horas extras e atraso por produto;

(iii) horas extras e atraso por produto com tempo fixo de *setup*;

Além da possibilidade de explorar o sequenciamento em mais de um período, inicialmente, introduzimos o cálculo dos custos relacionados a realização de horas extras e os custos relacionados a atrasos em prazos de entrega. A esse modelo demos o nome de MCHEA. Adicionalmente o modelo passou a considerar que *jobs* constituem produtos. Assim, os custos de atraso passaram a ser considerados por produto, o que significa dizer que, caso um *job* atrase, o custo relacionado ao produto do qual faz parte é computado na função objetivo. Esse modelo foi denominado MCHEAAP. Por fim, para alinhar o modelo ao contexto prático estabelecemos que a cada atividade de *setup* um tempo fixo passa a ser computado. Esse é o modelo mais aderente à realidade e recebeu o nome de MCHEAAPTF.

3.2.1. Modelo de Minimização do Custo com Horas Extras e Atrasos (MCHEA)

Consideramos que dado um conjunto J de *jobs* a serem sequenciados numa máquina CNC dentro de um horizonte composto por um conjunto T de períodos. Cada período corresponde a um dia de trabalho. Cada dia é dividido em $|N|$ posições que identificam a ordem de execução das atividades no dia formando o conjunto $N = \{1,...,n\}$, ou seja, se uma tarefa for executada na posição $n = 2$, essa então é a segunda tarefa processada pela máquina no dia. Para processar o conjuto de *jobs* são necessárias ferramentas compostas pelo conjunto M. A capacidade da caixa de ferramenta é constante e igual a C, que representa o número de ferramentas que podem ser carregadas simultaneamente. A capacidade da caixa de ferramentas é menor que o total de ferramentas requeridas para processar todos os *jobs* ,($C < |M|$), e, por isso, são necessárias trocas de ferramentas para que todas as tarefas sejam realizadas.

Para estabelecer as ferramentas disponíveis na máquina em um determinado período e as ferramentas que devem entrar ou sair, utilizamos as variáveis binárias (conforme proposto por Tang e Denardo (1988a)), q_t^m que indicam através do valor 1 as ferramentas que entram na máquina no instante inicial de cada período (antes de processar a primeira atividade), $w_{n,t}^m$ que indicam com o valor 1 as ferramentas que estão na máquina na posição n do período t e $p_{n,t}^m$ que indicam com valor 1 as ferramentas que entram na máquina após executada a tarefa da posição n do período t.

Vale notar que a variável $p_{n,t}^m$, para $n < |N|$, controla as ferramentas que entram na máquina antes de iniciar a tarefa $n+1$, e, para $n = |N|$, controla as ferramentas que entram na máquina ao final do período t, visando, se vantajoso, adiantar o *setup* para a primeira tarefa do período $t+1$. A variável binária $x_{n,t}^j$ indica com o valor 1, se a tarefa $j, j \in J$, deve ser executada na posição $n, n \in N$, do período $t, t \in T$. O parâmetro a_m^j indica se a tarefa $j, j \in J$, requer a ferramenta $m, m \in M$; e o tempo de processamento de cada tarefa é dado pelo parâmetro g_j. O tempo gasto com cada troca de ferramenta é constante, dado pelo parâmetro s.

A função objetivo tem duas componentes. A primeira está relacionada com minimizar os custos gerados com a realização de horas extras. Essas horas extras podem possuir custos unitários que diferem de acordo com o dia da semana,ou pela hora no dia em que ela é realizada. São considerados dois tipos de horas-extras: tipo 1 e tipo 2. O modelo possui as variáveis v_t^1 e v_t^2 que indicam a quantidade de horas extras do tipo 1 e 2, respectivamente, que foram realizadas no período t, $t \in T$. A essas horas extras são associados custos unitários o^1 e o^2, respectivamente, sendo que $o^1 < o^2$. Os limites de horas extras que podem ser realizadas num período t são b_t^1 e b_t^2 para as horas do tipo 1 e 2, respectivamente. A capacidade de produção (em horas) em cada período t é dada por u_t.

A segunda componente da função objetivo está relacionada com o prazo de entrega da tarefa, ou *due date*. Nesse caso, é possível assumir os custos de *backlog*. Assim, o modelo sempre tem solução viável, pois, mesmo que não seja possível realizar todos os *jobs* dentro do limite de horas-extras, é possível deixar de realizá-los dentro do horizonte de planejamento (através do atraso além do último período).

O modelo também foi ajustado para que os produtos pudessem ser divididos em grupos de acordo com a sua criticidade. Geralmente, clientes podem ser divididos em grupos de criticidades diferentes em relação ao *backlog* em virtude do *duedate*, sendo assim, pode ser mais vantajoso atender a um determinado cliente que a outro. Na prática as características das negociações comerciais e o relacionamento pode diferir muito entre os clientes de uma empresa. Muitas vezes pode ser vantajosos atrasar um *job* de um determinado cliente priorizando a produção de um *job* de outro cliente em virtude de um custo de atraso menor ou de ganhos comerciais melhores. Um determinado pedido pode gerar custos de *setup* elevados, mas, em contrapartida, pode representar faturamento elevado para a empresa, o que pode ser um fator decisivo na hora de posicioná-lo no sequenciamento. No estudo em questão, os clientes foram divididos em três grupos distintos de acordo com o custo de *backlog* de seus pedidos, multas contratuais existentes e impacto no relacionamento causado por atrasos.

O *due date* do job j é d_j, o que significa que é esperado que j seja processado a até o final do dia d_j para que não ocorram custos por atraso. A penalidade por atraso é dada por h_j associada ao total de atraso calculado pela variável $y_k^j = 1$, que indica que o *job* j não foi processado até o *k-ésimo* dia após seu *due date* (d_j). O total de atraso de um *job* é dado por $\sum_{k=1}^{|T|+1-d_j} y_k^j$. Por exemplo, se considerarmos um sequenciamento em seis dias ($|T| = 6$) e um *job* j cujo prazo de fabricação é o segundo dia ($d_j = 2$). Se esse *job* foi fabricado no quarto dia, então teremos que $y_1^j = 1$, $y_2^j = 1$, $y_3^j = 0$, $y_4^j = 0$ e $y_5^j = 0$, pois, o *job* j atrasou o primeiro e o segundo dia após sua data de fabricação. Se, para esse caso, tivermos um valor de $y_5^j = 1$ significa que o *job* não foi produzido dentro desse sequenciamento, pois, se o *job* não foi processado até o quinto dia após seu *due date*, significa que até o sexto dia (que é o último dia do sequenciamento) ele não foi produzido.

O modelo de Minimização de Custos com Horas Extras e Atrasos (MCHEA) pode ser visto seguir:

$$min \sum_{t \in T} \left(o^1 v_t^1 + o^2 v_t^2\right) + \sum_{j \in J} \sum_{k=1}^{|T|+1-d_j} h_j y_k^j \tag{3.1}$$

$$\left(\sum_{m \in M} \sum_{n \in N} p_{nt}^m + \sum_{m \in M} q_t^m \right) \cdot s + \sum_{j \in J} \sum_{n \in N} g_j x_{nt}^j \leq u_t + v_t^1 + v_t^2 \qquad \forall t \in T, \tag{3.2}$$

$$v_t^i \leq b_t^i \qquad i = 1{,}2; \quad \forall t \in T, \tag{3.3}$$

$$w_{(n+1)t}^m - w_{nt}^m - p_{nt}^m \leq 0 \qquad \forall n \in N | n < |N|; \forall m \in M, \forall t \in T, \tag{3.4}$$

$$w_{(1)t}^m - w_{N(t-1)}^m - p_{N(t-1)}^m - q_t^m \leq 0 \qquad \forall t \in T | t > 1; \forall m \in M, \tag{3.5}$$

$$w_{(1)(1)}^m - q_{(1)}^m \leq 0 \qquad \forall m \in M, \tag{3.6}$$

$$a_m^j x_{nt}^j - w_{nt}^m \leq 0 \qquad \forall j \in J, \forall n \in N, \forall m \in M, \forall t \in T, \tag{3.7}$$

$$\sum_{m \in M} w_{nt}^m \leq C \qquad \forall n \in N, \forall t \in T, \tag{3.8}$$

$$\sum_{j \in J} x_{(1)t}^j = 1 \qquad \forall t \in T, \tag{3.9}$$

$$\sum_{j \in J} \left(x_{(n+1)t}^j - x_{nt}^j \right) \leq 0 \qquad \forall n \in N | n < |N|; \forall t \in T, \tag{3.10}$$

$$\sum_{n \in N} \sum_{t=1}^{k} x_{nt}^j + y_{(k+1-d_j)}^j = 1 \qquad \forall j \in J; k = d_j, \ldots, |T| - 1 \tag{3.11}$$

$$y_{(k+1)}^j - y_k^j \leq 0 \qquad \forall j \in J; k = 1, \ldots, |T| + 1 - d_j \tag{3.12}$$

$$\sum_{k=|T|+2-d_j}^{|T|} y_k^j = 0 \qquad \forall j \in J \tag{3.13}$$

$$v^1, v^2 \geq 0 \tag{3.14}$$

$$x,y,w,p,q \in \{0,1\} \tag{3.15}$$

A função objetivo do modelo (3.1) computa o total dos custos com horas extras do tipo 1 e 2 e os custos de *backlog*. A restrição (3.2) garante que o tempo total gasto com as trocas de ferramentas somado ao tempo de processamento das atividades não exceda a capacidade de processamento (capacidade de realização de horas normais acrescidas das horas extras do tipo 1 e do tipo 2). A restrição (3.3) garante que a realização de horas extras não exceda o limite. A restrição (3.4) controla a troca de ferramentas para que aquelas ferramentas necessárias numa posição $n+1$ e que não estão na máquina na posição n sejam inseridas (entrando com valor 1 na variável $p^m_{n,t}$). As ferramentas necessárias à realização da primeira tarefa do dia podem ser colocadas na caixa de ferramentas através de trocas ao final do período anterior ou, antes do seu processamento, no início do dia. A restrição (3.5) garante que se um ferramenta requerida no instante 1 de um determinado período não está na máquina no último instante do período anterior ela deve ser colocada ou através de uma troca ao final do período anterior ($p^m_{N,(t-1)}$) ou através de uma troca no início do próprio período (q^m_t). No caso do período 1, as ferramentas requeridas ao processamento da primeira tarefa devem, necessariamente, ser inseridas antes do seu processamento , conforme prevê a restrição (3.6). Todas as ferramentas necessárias à realização de uma tarefa devem estar na caixa de ferramentas na posição em que essa tarefa é processada, conforme garante a restrição (3.7). A restrição (3.8) implica que o total de ferramentas na caixa de ferramentas não exceda sua capacidade. A restrição (3.9) exige que a primeira posição de cada dia seja ocupada por exatamente uma tarefa. A restrição (3.10) exige que um posição $n+1$ de um período t apenas seja utilizada caso algum job tenha sido alocado a posição anterior. A restrição (3.11) contabiliza os dias de atraso com relação à data de entrega da tarefa j. As restrições (3.12) fazem o encadeamento da variável y^j_t para contabilizar os dias de atraso de cada atividade. A restrição (3.13) atribui zero aos atrasos maiores que o horizonte de planejamento (casos em que a atividade não será realizada dentro do horizonte). As restrições (3.14) e (3.15) impõem os domínios das variáveis.

No modelo MCHEA (composto por (3.1) a (3.15)), como mencionado anteriormente, existe a possibilidade de não se processar uma determinada tarefa dentro do horizonte de planejamento caso não seja viável ou, por exemplo, o custo com as horas extras necessárias exceda o custo de atraso.

3.2.2. Modelo de Minimização de Custos com Horas Extras e Atrasos Agrupado por Produtos (MCHEAAP)

Para alinhar ainda mais a aplicação do modelo ao contexto real, introduzimos variáveis e restrições para agrupar *jobs* em produtos maiores. A ideia é que se um único *job* componente atrasa, o atendimento ao prazo de entrega do produto inteiro é comprometido. Os produtos são denotados por uma coleção de conjuntos não vazios Θ_r, $r = 1, \ldots, R$, que compõem uma partição de J. Como *jobs* que fazem parte de um mesmo produto podem ter sequências tecnológicas diferentes a partir da máquina CNC, é possível que tenham prazos de término diferentes, por exemplo, um produto pode ser composto por três *jobs*, um deles está pronto após o processo de puncionamento na máquina CNC, ou seja, não precisa passar por outros processos. Esse *job* terá o prazo de término do processo CNC igual ao prazo do produto completo. Um segundo *job* componente desse mesmo produto necessita ser dobrado e soldado após o processo de puncionamento. O prazo de término do processo CNC desse *job* deverá ser menor para que, no prazo de entrega do produto, ele esteja pronto. Cada *job* j componente do produto Θ_r pode ter seu próprio *due date* d_j, mas, no caso de atrasos, a penalidade é associada ao total de *jobs* que compõem aquele produto e é denotada por h_{Θ_r}. O *due date* do produto é representado por d_r que denota o *due date* mais cedo dentre os *jobs* pertencentes a Θ_r. De maneira análoga a variável y_k^j, proposta anteriormente, a variável binária $z_k^r = 1$ indica que o produto Θ_r atrasou o *k-ésimo* dia em virtude de um ou mais *jobs* que o compõem não terem sido finalizados.

O máximo de dias que um produto pode ser considerado atrasado é a diferença entre $|T| + 1$ e o *duedate* mais cedo entre os *jobs* que o compõem. Como exemplo, em um planejamento de seis dias ($|T| = 6$), um dos produtos a serem sequenciados é composto por 2 *jobs* ($\Theta_r = \{j_1, j_2\}$). Os prazos de término são: $d_{j_1} = 2$ e $d_{j_2} = 4$. Nesse caso, o atraso máximo que pode ser verificado é de cinco dias, pois, $T + 1 - d_{j_1} = 5$. No exemplo, o *due date* utilizado para o produto Θ_r foi $d_{j_1} = 2$, pois, é o prazo mais cedo dentre os *jobs* componentes de Θ_r.

Para adequar o MCHEA ao agrupamento de produtos, alteramos a função objetivo (3.16) e acrescentmos mais duas restrições (3.17) e (3.18).

$$min \sum_{t \in T} \left(o^1 v_t^1 + o^2 v_t^2\right) + \sum_{r=1}^{R} \sum_{k=1}^{|T|+1-d_r} h_{\Theta_r} z_k^r \tag{3.16}$$

$$y_k^j - z_k^r \leq 0 \qquad \forall r = 1, \ldots, R; \forall j \in \Theta_r; k = 1, \ldots, |T| + 1 - d_j \tag{3.17}$$

$$z \geq 0 \tag{3.18}$$

O segundo termo da função objetivo (3.16) (que substitui o segundo termo de (3.1)) representa os custos de atrasos de produtos. Esse termo computa cada período de atraso de cada produto o que gera impacto negativo no valor da função objetivo.

A restrição (3.17) garante que o atraso de qualquer *job* componente de Θ_r é computado como o atraso do produto como um todo. Por exemplo, dado um produto ($r = 1$, por exemplo) composto por 2 *jobs* ($\Theta_1 = \{j_1, j_2\}$), se apenas um dos dois *jobs*, por exemplo o primeiro, atrasar o primeiro e o segundo dia (o segundo *job* foi realizado dentro do prazo), teremos que $y_1^{j_1} = 1$ e $y_2^{j_1} = 1$ e $y_1^{j_2} = 0$ e $y_2^{j_2} = 0$. Nesse caso será computado o atraso do produto para o primeiro e segundo período após o *due date* ocasionado pelo atraso de j_1. Logo $z_1^1 = 1$ e $z_2^1 = 1$.

Assim, o MCHEAAP está composto pela função objetivo (3.16) e pelas restrições (3.2) a (3.15), (3.17) e (3.18).

3.2.3. Modelo de Minimização de Custos com Horas Extras e Atrasos Agrupado por Produtos com Tempos Fixos de Parada (MCHEAAPTF)

O tempo gasto em uma atividade de *setup* varia de acordo com a quantidade de ferramentas que serão trocadas, entretanto, existe um parcela desse tempo que é fixa. Essa parcela é representada pelo tempo utilizado pelo operador para retirar a caixa de ferramentas da máquina e depois inserí-la novamente. Esse tempo é fixo e tem que ser computado sempre que houver troca de ferramentas (independente da quantidade de ferramentas que foram trocadas). Assim, para se trocar uma ou dez ferramentas esse tempo deverá ser computado. Dessa forma, no sequenciamento das atividades pode ser interessante diminuir o número de vezes em que atividades de troca são realizadas, ou, caso uma parada seja requerida, realizar mais trocas diluindo assim esse custo fixo.

Introduzimos as variáveis f_{nt} que recebem 1 quando houve troca de ferramentas na posição n do período t e l_t que recebem 1 quando houve troca de ferramentas antes de iniciar as atividades do período t (associada a variável q_t^m). O tempo fixo gasto em uma atividade de setup é dado pelo parêmetro b.

A restrição (3.2) é alterada conforme segue:

$$\left(\sum_{m\in M}\sum_{n\in N} p_{nt}^m + \sum_{m\in M} q_t^m\right)\cdot s + \left(\sum_{n\in N} f_{nt} + l_t\right)\cdot b + \sum_{j\in J}\sum_{n\in N} g_j x_{nt}^j \leq u_t + v_t^1 + v_t^2 \qquad \forall t \in T \tag{3.19}$$

Duas restrições precisam ser inseridas ao modelo para computar o tempo fixo de parada a cada vez que trocas forem realizadas:

$$p_{nt}^m - f_{nt} \leq 0 \qquad \forall m \in M; \forall n \in N; \forall t \in T \tag{3.20}$$

$$q_t^m - l_t \leq 0 \qquad \forall m \in M; \forall t \in T \tag{3.21}$$

$$f_{nt}, l_t \geq 0 \tag{3.22}$$

Assim, o MCHEAAPTF está composto pela função objetivo (3.16) e pelas restrições (3.3) a (3.15) e (3.19) a (3.22).

Capítulo 4

Experimentos Númericos

Os experimentos numéricos foram realizados com instâncias obtidas a partir de sequências de produção realizadas no contexto industrial. Essas instâncias foram obtidas a partir de 4 cenários. Cada cenário corresponde ao planejamento de uma semana (segunda-feira a sábado) que possui 6 períodos. Para fins de análise, os cenários também foram divididos em instâncias menores variando de 2 a 6 períodos. Assim, serão apresentados testes computacionais com 20 situações distintas. Para avaliar o comportamento dos modelos, cada instância foi testada com cada um dos 3 modelos (MCHEA, MCHEAP e MCHEAPTF).

Na ambiente fabril em questão, os *jobs* são ordenados de acordo com as prioridades de produção (data de entrega mais próxima) e agrupados por produto. Assim, todas as ordens de produção de um determinado produto são colocadas em produção em sequência. Geralmente, os *jobs* entram em produção na ordem que os pedidos são recebidos, mas, eventualmente, quando um determinado produto apresenta um prazo de produção mais curto, pode ter prioridade.

Os custos verificados através da utilização de horas extras para expandir a capacidade produtiva foram divididos em dois níveis. Na prática, esses custos variam de acordo com o dia da semana, com o total de horas realizadas em único dia e em um mês. Esses custos são de R$33,00 por hora para horas extras do tipo 1 (v^1) (realizadas de segunda a sexta-feira até um limite de duas horas por dia). O custos com horas extras do tipo 2 (v^2) (realizadas aos sábados e com limite de 5 horas ou horas realizadas além do limite de duas horas nos dias de semana) e de R$42,00. Em cada dia da semana (segunda-feira a sábado) estão disponiveis 9 horas de processamento normal (sem qualquer custo adicional), duas horas do tipo 1 e duas horas do tipo 2. Aos sábados estão disponíveis apenas 4 horas de processamento normal e até 5 horas do tipo 2. Não é possível realizar horas do tipo 1 aos sábados.

Cada *job* possui um tempo de processamento que pode ser obtido através do *software* utilizado na geração do programa de fabricação. Esse tempo dificilmente sofre alterações por fatores externos. O tempo para trocar uma ferramenta qualquer é igual a 3 minutos. O tempo fixo do *setup*, equivalente ao tempo necessário para remover a caixa de ferramentas da máquina e depois colocá-la novamente, é igual a 12 minutos. Essa parcela fixa de tempo foi considerada apenas na apuração dos custos no modelo MCHEAPTF. A capacidade da caixa de ferramentas é de 10 ferramentas.

Os clientes foram divididos em 3 grupos, de acordo com sua criticidade. Assim foram verificados custos relativos ao atraso de um período de um determinado produto para cada grupo de clientes. Esses custos são de R$50,00 , R$150,00 e R$300,00. Nos cálculos utilizados no modelo MCHEA (que não considera os jobs agrupados em produtos) o custo de cada produto foi dividido pelo total de *jobs* que o compõem.

Durante os testes foi utilizado o pacote de otimização CPLEX 11 (com configurações padrão) com tempo limitado em duas horas. Esses testes foram realizados utilizando um processador Intel Core i3, com 4GB de memória RAM.

4.1. Descrição dos cenários

Conforme descrito anteriormente as instâncias foram obtidas a partir de sequências de produção executadas no contexto industrial. Cada cenário gerou 5 instâncias que variam de 2 até 6 períodos.

4.1.1. Cenário 1

O cenário 1 está composto pelo sequenciamento de 17 produtos. Esses produtos são compostos por 44 *jobs*. Para processar todos os 44 jobs são necessárias 35 ferramentas. Em média, os *jobs* demandam 4 ferramentas para serem processados. O *job* que requer menos ferramentas em seu processamento utiliza uma ferramenta e o *job* que requer mais utiliza oito. O tempo médio de processamento de cada *job* é de 62,80 minutos e o máximo e mínimo são, respectivamente, 23 e 107 minutos. No sequênciamento realizado no chão de fábrica para executar as atividades do cenário 1 foram utilizadas um total de 10 horas extras do tipo 1 e 6,58 horas extras do tipo 2. O custos com horas extras somaram R$606,50. Mesmo com o emprego de horas extras, houve atrasos em prazos de entregas de alguns produtos. Tais atrasos geraram custos de R$1.350,00. Para efeitos de avaliação do modelo MCHEA, que não contempla o fato dos produtos serem composto por conjuntos de *jobs*, foram calculados custos de atraso para os *jobs* isoladamente, o que gerou um custo de R$870,00. No total, a realização do cenário 1 na prática gerou custos de R$1.956,50 (com *jobs* agrupados por produtos, como ocorre na prática).

A síntese dos resultados obtidos com a aplicação dos modelos nas instâncias obtidas a partir do cenário 1 pode ser vista nas tabelas 4.1, 4.2, 4.3 e 4.4.

A aplicação do modelo MCHEAPCF, na instância com 5 períodos, observou-se que, no prazo de 2 horas, não houve resposta de boa qualidade. Nessa instância a sequência obtida gerou um resultado com custo maior que aquela realizada na prática.

As tabelas 4.1, 4.2, 4.3 apresentam os resultados obtidos com os testes computacionais e a comparação entre esses resultados e o valores verificados na prática. As colunas $|T|$ e $|J|$ indicam o número de períodos e *jobs* na instância, respectivamente. As quatro colunas seguintes são referentes aos valores ocorridos na prática e indicam a quantidade de horas do tipo 1 (v^1), do tipo 2 (v^2), os custos gerados pela realização de horas extras ($o(\$)$) e os custos gerados por atrasos em prazos de entrega ($h(\$)$). Em seguida são apresentados os dados relativos a sequencia obtida com a aplicação dos modelos. As quatro colunas seguintes apresentam os valores análogos aos apresentados nos resultados práticos. Em seguida são apresentados os valores totais dos custos obtidos com o modelo (total($)) que são o resultado da soma dos custos de realização de horas extras e de atrasos. A coluna r(%) apresenta a redução percentual no custo calculado através da divisão da diferença entre o resultado prático (RP) e o resultado do modelo (RM), pelo resultado prático (RP) ($r(\%) = \frac{RP-RM}{RP}$).

A tabela 4.4 sintetiza as informações fornecidas pelo pacote de otimização. A primeira coluna indica a qual modelo os resultados se referem. A segunda coluna indica o número de períodos da instância. A terceira coluna apresenta o valor da função objetivo obtido com as relaxações lineares dos modelos. A coluna quatro apresenta os $GAPs$ absolutos entre os melhores limitantes superior e inferior gerados pelo CPLEX após duas horas. Na quinta coluna a primeira solução de cada instância é apresentada. Na sexta e sétima colunas número total de nós e o número de nós deixados, respectivamente. A oitava coluna informa a solução final do CPLEX (no tempo estabelecido), a nona coluna os valores de *lower bound* final de cada instância e na décima coluna o $GAPs$ ($GAP = \frac{Sol.final - LBfinal}{Sol.Final}$).

Tabela 4.1: Resultados com a aplicação do MCHEA - Comparativo do Cenário 1

		Prática				MCHEA					
$\|T\|$	$\|J\|$	v^1	v^2	$o(\$)$	$h(\$)$	v^1	v^2	$o(\$)$	$h(\$)$	total($)	r(%)
2	17	4,00	0,80	165,60	300,00	1,62	0,00	53,35	0,00	53,35	88,54%
3	28	6,00	3,22	333,10	420,00	1,53	0,00	50,60	75,00	125,60	83,32%
4	32	8,00	3,33	404,00	570,00	1,75	0,00	57,75	75,00	132,75	86,37%
5	40	10,00	3,57	479,80	870,00	3,07	0,00	101,20	62,50	163,70	87,87%
6	44	10,00	6,58	606,50	870,00	3,53	0,00	116,60	50,00	166,60	88,72%

Tabela 4.2: Resultados com a aplicação do MCHEAAP - Comparativo do Cenário 1

		Prática				MCHEAAP					
$\|T\|$	$\|J\|$	v^1	v^2	$o(\$)$	$h(\$)$	v^1	v^2	$o(\$)$	$h(\$)$	total($)	r(%)
2	17	4,00	0,80	165,60	600,00	1,62	0,00	53,35	0,00	53,35	93,03%
3	28	6,00	3,22	333,10	750,00	2,77	0,00	91,30	50,00	141,30	86,95%
4	32	8,00	3,33	404,00	1050,00	3,03	0,00	100,10	53,50	153,60	89,44%
5	40	10,00	3,57	479,80	1350,00	3,92	0,45	148,15	50,00	198,15	89,17%
6	44	10,00	6,58	606,50	1350,00	3,80	0,40	142,20	650,00	792,20	59,51%

Tabela 4.3: Resultados com a aplicação do MCHEAAPTF - Comparativo do Cenário 1

		Prática				MCHEAATCF					
$\|T\|$	$\|J\|$	v^1	v^2	$o(\$)$	$h(\$)$	v^1	v^2	$o(\$)$	$h(\$)$	total($)	r(%)
2	17	4,00	0,80	165,60	600,00	3,18	0,13	110,65	0,00	110,65	85,55%
3	28	6,00	3,22	333,10	750,00	2,00	0,88	103,10	150,00	253,10	76,63%
4	32	8,00	3,33	404,00	1050,00	3,98	0,63	158,05	350,00	508,05	65,06%
5	40	10,00	3,57	479,80	1350,00	4,95	1,63	231,95	550,00	781,95	57,27%
6	44	10,00	6,58	606,50	1350,00	2,48	1,90	161,64	2650,11	2811,75	-43,71%

Tabela 4.4: Síntese dos resultados do pacote de otimização - Cenário 1

Modelo	$\|T\|$	Relax. Lin.	*GAP* ABS	1ª Sol. CPLEX	Num Nós	Nós Deixados	Sol. Final	LB. Final	*GAP* Ot.
MCHEA	2	22,11	7,70	1712,50	344500	270070	53,35	45,65	0,14
MCHEAAP	2	22,11	8,94	1900,00	375400	310274	53,35	44,41	0,17
MCHEAAPTF	2	23,63	63,24	1900,00	228400	185187	110,65	47,42	0,57
MCHEA	3	62,38	31,60	3195,00	95300	70959	125,60	94,00	0,25
MCHEAAP	3	70,10	44,08	3702,20	91100	68565	141,30	97,22	0,31
MCHEAAPTF	3	72,09	138,29	3753,35	31600	22822	253,10	114,81	0,55
MCHEA	4	62,38	40,54	4897,50	87600	67932	132,75	92,21	0,31
MCHEAAP	4	70,10	50,84	5652,20	38900	33557	153,60	102,76	0,33
MCHEAAPTF	4	72,09	393,59	5705,55	7402	4632	508,05	114,46	0,77
MCHEA	5	62,38	75,53	7572,50	26183	17602	163,70	88,17	0,46
MCHEAAP	5	70,10	100,79	8854,95	13800	9815	198,15	97,36	0,51
MCHEAAPTF	5	72,09	670,67	8959,45	4506	3856	781,95	111,26	0,86
MCHEA	6	62,38	85,00	10027,00	24500	16538	166,60	81,60	0,51
MCHEAAP	6	70,10	691,13	11935,90	11131	10325	792,20	101,07	0,87
MCHEAAPTF	6	179,65	2697,22	11500,00	717	234	2811,75	114,53	0,96

4.1.2. Cenário 2

No cenário 2 foram realizados 40 *jobs*. Esses *jobs* correspondem a um total de 16 produtos. Em média, esses *jobs* demandam 4 ferramentas para a sua execução. A exemplo do cenário 1, o *job* que requer o menor número de ferramentas para seu processamento utilza 1 ferramenta o *job* que requer o maior número utiliza 8. Para realizar todos os *jobs* foram necessárias 34 ferramentas. O tempo médio de processamento dos *jobs* é igual a 69,3 minutos e os valores mínimos e máximo são, respectivamente, 27 e 112 minutos. Nesse cenário foram realizadas 3,30 horas extras do tipo 2, pois, ao final do planejamento verificou-se que seria necessário trabalhar no sábado. Como não houve atraso em prazos de entrega, o custo total do cenário 2 foi R$138,60.

Como não houve atraso em prazos de entrega, o custo total do cenário 2 foi R$138,60. A síntese dos resultados obtidos com a aplicação dos modelos nas instâncias obtidas a partir do cenário 2 pode ser vista nas tabelas 4.5, 4.6, 4.7 e 4.8.

Em todos os testes realizados nas instâncias obtidas no cenário 2 os resultados foram satisfatórios.

Tabela 4.5: Resultados com a aplicação do MCHEA - Comparativo do Cenário 2

		Prática				MCHEA					
$\|T\|$	$\|J\|$	v^1	v^2	o($)	h($)	v^1	v^2	o($)	h($)	total($)	r(%)
2	13	3,28	0,08	111,85	0,00	0,38	0,00	12,65	0,00	12,65	88,69%
3	21	5,28	1,48	236,65	0,00	0,38	0,00	12,65	0,00	12,65	94,65%
4	30	7,28	1,72	312,45	50,00	1,15	0,00	37,95	0,00	37,95	89,53%
5	37	9,28	1,88	385,45	350,00	1,20	0,00	39,60	0,00	39,60	94,62%
6	40	9,28	5,18	524,05	350,00	1,82	0,00	59,95	0,00	59,95	93,14%

Tabela 4.6: Resultados com a aplicação do MCHEAAP - Comparativo do Cenário 2

		Prática				MCHEAAP					
$\|T\|$	$\|J\|$	v^1	v^2	o($)	h($)	v^1	v^2	o($)	h($)	total($)	r(%)
2	13	3,28	0,08	111,85	0,00	0,38	0,00	12,65	0,00	12,65	88,69%
3	21	5,28	1,48	236,65	0,00	0,38	0,00	12,65	0,00	12,65	94,65%
4	30	7,28	1,72	312,45	150,00	1,35	0,00	44,55	0,00	44,55	90,37%
5	37	9,28	1,88	385,45	450,00	1,30	0,00	42,90	0,00	42,90	94,87%
6	40	9,28	5,18	524,05	450,00	1,82	0,00	59,95	0,00	59,95	93,85%

Tabela 4.7: Resultados com a aplicação do MCHEAAPTF - Comparativo do Cenário 2

		Prática				MCHEAAPTF					
$\|T\|$	$\|J\|$	v^1	v^2	o($)	h($)	v^1	v^2	o($)	h($)	total($)	r(%)
2	13	3,28	0,08	111,85	0,00	0,98	0,00	32,45	0,00	32,45	70,99%
3	21	5,28	1,48	236,65	0,00	2,57	0,00	84,70	0,00	84,70	64,21%
4	30	7,28	1,72	312,45	150,00	4,93	0,07	165,60	0,00	165,60	64,19%
5	37	9,28	1,88	385,45	450,00	5,90	0,00	194,70	0,00	194,70	76,70%
6	40	9,28	5,18	524,05	450,00	6,00	3,28	335,90	0,00	335,90	65,52%

Tabela 4.8: Síntese dos resultados do pacote de otimização - Cenário 2

Modelo	$\|T\|$	Relax. Lin.	*GAP* ABS	1ª Sol. CPLEX	Num Nós	Nós Deixados	Sol. Final	LB. Final	*GAP* Ot.
MCHEA	2	0,00	12,65	1562,50	26700	47	12,65	11,92	0,06
MCHEAAP	2	0,00	12,65	1651,10	53400	59	12,65	12,24	0,03
MCHEAAPTF	2	0,00	32,45	1950,00	85700	41	32,45	32,29	0,00
MCHEA	3	0,00	12,65	3187,50	78800	55	12,65	12,41	0,02
MCHEAAP	3	0,00	12,65	3451,10	61100	38	12,65	12,49	0,01
MCHEAAPTF	3	3,03	81,67	3750,00	50200	43350	84,70	3,03	0,96
MCHEA	4	0,00	37,95	5550,00	30200	25640	37,95	0,00	1,00
MCHEAAP	4	0,00	44,55	6150,00	34100	30398	44,55	0,00	1,00
MCHEAAPTF	4	0,00	165,60	6208,30	12200	10764	165,60	0,00	1,00
MCHEA	5	0,00	39,60	7375,00	28300	21909	39,60	0,00	1,00
MCHEAAP	5	0,00	42,90	8551,10	25800	21608	42,90	0,00	1,00
MCHEAAPTF	5	0,00	194,70	8550,00	5000	3748	194,70	0,00	1,00
MCHEA	6	0,00	59,95	11012,50	21300	17508	59,95	0,00	1,00
MCHEAAP	6	0,00	59,95	10876,15	11400	9136	59,95	0,00	1,00
MCHEAAPTF	6	0,00	335,90	11001,80	1400	757	335,90	0,00	1,00

4.1.3. Cenário 3

O terceiro cenário compreende 16 produtos compostos por um total de 41 *jobs*. Para o processamento de todos os *jobs* foram necessárias 35 ferramentas. O *job* que utiliza o menor número de ferramentas necessita de 1 ferramenta para o seu processamento. O *job* que utiliza o maior número de ferramentas necessita de 8 ferramentas. O valor médio de ferramentas demandas para a fabricação dos *jobs* e igual 4,66. O tempo médio de processamento foi de 65,27 minutos e os valores minimos e máximos de tempo de processamento foram: 24 e 111 minutos, respectivamente. Para a realização das atividades no sequenciamento foram utilizadas 14,40 horas extras, sendo que 8,97 foram horas extras do tipo 1 e 5,43 do tipo 2. Essas horas extras totalizaram custos de R$524,10. Mesmo com a aplicação de horas extras, nesse cenário foram apurados custos de atraso em entregas de produtos no valor de R$950,00 (ou R$770,00 quando os *jobs* são tratados isoladamente). O custo total para verificado no cenário foi R$1.474,10.

A síntese dos resultados obtidos com a aplicação dos modelos nas instâncias obtidas a partir do cenário 3 pode ser vista nas tabelas 4.9, 4.10, 4.11 e 4.12.

Tabela 4.9: Resultados com a aplicação do MCHEA - Comparativo do Cenário 3

		Prática				MCHEA					
$\|T\|$	$\|J\|$	v^1	v^2	o($)	h($)	v^1	v^2	o($)	h($)	total($)	r(%)
2	18	3,25	0,00	107,25	270,00	1,05	0,00	34,65	30,00	64,65	82,86%
3	23	5,25	1,38	231,35	420,00	1,05	0,00	34,65	30,00	64,65	90,07%
4	31	6,97	1,38	288,00	720,00	1,37	0,00	45,10	30,00	75,10	92,55%
5	38	8,97	1,53	360,30	770,00	2,68	0,27	99,75	30,00	129,75	88,52%
6	41	8,97	5,43	524,10	770,00	1,62	0,00	53,35	105,00	158,35	87,76%

Tabela 4.10: Resultados com a aplicação do MCHEAAP - Comparativo do Cenário 3

		Prática				MCHEAAP					
$\|T\|$	$\|J\|$	v^1	v^2	$o(\$)$	$h(\$)$	v^1	v^2	$o(\$)$	$h(\$)$	total($)	r(%)
2	18	3,25	0,00	107,25	300,00	2,65	0,05	89,55	0,00	89,55	78,01%
3	23	5,25	1,38	231,35	600,00	2,65	0,08	90,95	0,00	90,95	89,06%
4	31	6,97	1,38	288,00	900,00	2,78	0,05	93,95	0,00	93,95	92,09%
5	38	8,97	1,53	360,30	950,00	2,48	0,22	91,05	0,00	91,05	93,05%
6	41	8,97	5,43	524,10	950,00	2,45	0,38	96,95	0,00	96,95	93,42%

Tabela 4.11: Resultados com a aplicação do MCHEAAPTF - Comparativo do Cenário 3

		Prática				MCHEAAPTF					
$\|T\|$	$\|J\|$	v^1	v^2	$o(\$)$	$h(\$)$	v^1	v^2	$o(\$)$	$h(\$)$	total($)	r(%)
2	18	3,25	0,00	107,25	300,00	2,00	0,08	69,50	150,00	219,50	46,10%
3	23	5,25	1,38	231,35	600,00	2,98	0,00	98,45	450,00	548,45	34,03%
4	31	6,97	1,38	288,00	900,00	4,00	1,68	202,70	900,00	1102,70	7,18%
5	38	8,97	1,53	360,30	950,00	1,33	0,00	44,00	1500	1544,00	-17,84%
6	41	8,97	5,43	524,10	950,00	0,15	0,00	4,95	6750,00	6754,95	-358,24%

Tabela 4.12: Síntese dos resultados do pacote de otimização - Cenário 3

Modelo	$\|T\|$	Relax. Lin.	*GAP* ABS	1ª Sol. CPLEX	Num Nós	Nós Deixados	Sol. Final	LB. Final	*GAP* Ot.
MCHEA	2	35,79	28,86	1695,00	321600	282846	64,65	35,79	0,45
MCHEAAP	2	60,96	28,59	1950,00	130400	107440	89,55	63,96	0,29
MCHEAAPTF	2	71,36	148,14	2100,00	93600	78402	219,50	71,36	0,67
MCHEA	3	38,12	26,53	3150,55	110800	93228	64,65	38,12	0,41
MCHEAAP	3	58,84	32,11	3900,00	138500	116977	90,95	58,84	0,35
MCHEAAPTF	3	64,06	484,39	3900,00	21000	15917	548,45	64,06	0,88
MCHEA	4	41,57	33,53	4335,00	48200	29054	75,10	41,57	0,45
MCHEAAP	4	66,37	27,58	6300,00	49700	37676	93,95	66,37	0,29
MCHEAAPTF	4	66,96	1035,74	6300,00	9700	5613	1102,70	66,96	0,94
MCHEA	5	36,92	92,83	8105,00	36400	30612	129,75	36,92	0,72
MCHEAAP	5	54,92	36,13	9051,10	38400	26818	91,05	54,92	0,40
MCHEAAPTF	5	-944,08	2488,08	9000,00	6300	5753	1544,00	69,02	0,96
MCHEA	6	34,91	123,44	10590,00	20900	14444	158,35	34,91	0,78
MCHEAAP	6	62,33	34,62	12300,00	15600	10292	96,95	62,33	0,36
MCHEAAPTF	6	68,35	6686,60	12400,00	800	307	6754,95	68,35	0,99

4.1.4. Cenário 4

O cenário 4 está composto pelo sequenciamento de 15 produtos. Esses produtos são compostos por 40 *jobs*. Para processar todos os 40 jobs são necessárias 35 ferramentas. Em média, os *jobs* demandam 4,75 ferramentas para serem processados. O *job* que requer menos ferramentas em seu processamento utiliza uma ferramenta e o *job* que requer mais utiliza oito. O tempo médio de processamento de cada *job* é de 65,23 minutos e o máximo e mínimo são, respectivamente, 27 e 112 minutos. No sequênciamento realizado no chão de fábrica para executar as atividades do cenário 4 foram utilizadas um total de 6,80 horas extras do tipo 1 e 4,73 horas extras do tipo 2. O custos com horas extras somaram R$423,20. Houve atrasos em prazos de entregas de alguns produtos e tais atrasos geraram custos de R$750,00. A realização do cenário 4 na prática gerou custos de R$1.173,20.

A síntese dos resultados obtidos com a aplicação dos modelos nas instâncias obtidas a partir do cenário 4 pode ser vista nas tabelas 4.13, 4.14, 4.15 e 4.16.

Tabela 4.13: Resultados com a aplicação do MCHEA - Comparativo do Cenário 4

		Prática				MCHEA					
$\|T\|$	$\|J\|$	v^1	v^2	$o(\$)$	$h(\$)$	v^1	v^2	$o(\$)$	$h(\$)$	total($)	r(%)
2	15	1,30	0,00	42,90	100,00	0,57	0,00	18,70	0,00	18,70	86,91%
3	19	2,80	0,00	92,40	100,00	0,57	0,00	18,70	0,00	18,70	90,28%
4	29	4,80	1,90	238,20	137,50	0,57	0,00	18,70	0,00	18,70	95,02%
5	37	6,80	2,78	341,30	237,50	0,58	0,00	19,25	0,00	19,25	96,67%
6	40	6,80	4,73	423,20	237,50	0,57	0,00	18,70	0,00	18,70	97,17%

Tabela 4.14: Resultados com a aplicação do MCHEAAP - Comparativo do Cenário 4

		Prática				MCHEAAP					
$\|T\|$	$\|J\|$	v^1	v^2	$o(\$)$	$h(\$)$	v^1	v^2	$o(\$)$	$h(\$)$	total($)	r(%)
2	15	1,30	0,00	42,90	300,00	0,57	0,00	18,70	0,00	18,70	94,55%
3	19	2,80	0,00	92,40	300,00	0,57	0,00	18,70	0,00	18,70	95,23%
4	29	4,80	1,90	238,20	450,00	0,57	0,00	18,70	0,00	18,70	97,28%
5	37	6,80	2,78	341,30	750,00	0,62	0,00	20,35	0,00	20,35	98,14%
6	40	6,80	4,73	423,20	750,00	0,80	0,00	26,40	0,00	26,40	97,75%

Tabela 4.15: Resultados com a aplicação do MCHEAAPTF - Comparativo do Cenário 4

		Prática				MCHEAAPTF					
$\|T\|$	$\|J\|$	v^1	v^2	o(\$)	h(\$)	v^1	v^2	o(\$)	h(\$)	total(\$)	r(%)
2	15	1,30	0,00	42,90	300,00	1,17	0,00	38,50	0,00	38,50	88,77%
3	19	2,80	0,00	92,40	300,00	1,17	0,00	38,50	0,00	38,50	90,19%
4	29	4,80	1,90	238,20	450,00	3,52	0,00	116,05	0,00	116,05	83,14%
5	37	6,80	2,78	341,30	750,00	5,75	0,42	207,25	0,00	207,25	81,01%
6	40	6,80	4,73	423,20	750,00	4,60	2,35	250,50	1350,00	1600,50	-36,42%

Tabela 4.16: Síntese dos resultados do pacote de otimização - Cenário 4

Modelo	$\lvert T \rvert$	Relax. Lin.	*GAP* ABS	1ª Sol. CPLEX	Num Nós	Nós Deixados	Sol. Final	LB. Final	*GAP* Ot.
MCHEA	2	0,00	18,70	925,00	554600	25	18,70	18,66	0,00
MCHEAAP	2	0,00	18,70	1400,00	476800	47	18,70	18,70	0,00
MCHEAAPTF	2	21,25	17,25	526,40	357600	219664	38,50	21,25	0,45
MCHEA	3	0,00	18,70	2050,00	611000	90	18,70	18,62	0,00
MCHEAAP	3	14,36	4,35	2651,65	545200	183658	18,70	14,35	0,23
MCHEAAPTF	3	8,79	29,71	2350,00	301100	225190	38,50	8,79	0,77
MCHEA	4	3,29	15,41	4475,00	56100	40931	18,70	3,29	0,82
MCHEAAP	4	5,35	13,35	4950,00	118900	89199	18,70	5,35	0,71
MCHEAAPTF	4	0,00	116,05	4950,00	10900	8589	116,05	0,00	1,00
MCHEA	5	0,00	19,25	7062,50	23400	19116	19,25	0,00	1,00
MCHEAAP	5	10,00	10,35	7602,20	20400	16386	20,35	0,00	1,00
MCHEAAPTF	5	0,00	207,25	7300,00	3016	2130	207,25	0,00	1,00
MCHEA	6	0,00	18,70	9387,50	19300	12654	18,70	0,00	1,00
MCHEAAP	6	0,00	26,40	10324,55	5600	3087	26,40	0,00	1,00
MCHEAAPTF	6	0,00	1600,50	10515,40	1100	553	1600,50	0,00	1,00

4.1.5. Síntese dos Resultados

Através da análise dos resultados obtidos com a implementação dos três modelos nos quatro cenários, é possível verficar que o problema é de solução bastante dispendiosa do ponto de vista de tempo computacional.

Com o tempo limite de processamento estabelecido em duas horas, verificamos que apenas em poucas instâncias os modelos foram capazes de obter respostas ótimas. Além disso, nessas instâncias apenas dois ou três períodos estavam sendo sequenciados. No entanto, em apenas quatro instâncias os resultados obtidos com os modelos tiveram custos maiores que os resultados obtidos na prática. Nesses casos, o horizonte de planejamento era de 5 (em um caso) ou 6 períodos (três casos) e o modelo era o MCHEAAPTF. É possivel que o problema nesses casos seja o tempo computacional de duas horas que não possibilitou que fossem encontradas soluções de boa qualidade.

Assim, como experimento adicional, as mesmas instâncias foram testadas com o mesmo modelo, porém, com tempos computacionais maiores. A síntese dos resultados obtidos pode ser vista na tabela 4.17.

Tabela 4.17: Testes adicionais com tempo computacional extra

Cenário	$\|T\|$	$\|J\|$	Tc	Prática	MCHEAAPTF	Redução	GAP
1	6	44	5	1956,50	483,50	75,29%	83,35%
3	5	38	5	1310,30	191,45	85,39%	98,06%
3	6	41	10	1474,10	1263,45	14,29%	64,87%
4	6	40	5	1173,20	475,95	59,43%	94,40%

Na tabela 4.17 a primeira coluna descreve o cenário ao qual a instância pertence, a segunda coluna informa o número de períodos, a terceira coluna informa o número de jobs da instâncias. A quarta coluna, denominada Tc, informa o tempo computacional utilizado no experimento. O custos verificados na prática e através do MCHEAPTF são representados na quinta e sexta coluna, respectivamente, a sétima coluna informa a redução (em termos percentuais) obtida com a aplicação do modelo e a oitava coluna apresenta o GAP de otimalidade fornecido pelo *solver* ao final do tempo computacional estabelecido.

Esses experimentos demonstraram que a elevação do tempo computacional pode melhorar consideravelmente a qualidade das soluções. Passando o tempo de 2 horas para 5 horas houve ganho suficiente para melhorar o resultado comparado a prática em 3 instâncias. Apenas para a instância com

6 períodos do cenário 3 essa elevação no tempo não foi suficiente. Assim, um novo teste com 10 horas de processamento foi conduzido e o resultado também superou o custo verificado na prática.

De um modo geral, a exceção dos quatro casos citados anteriormente, os resultados obtidos com os três modelos são bastante bons e encorajadores. No contexto fabril a sua aplicação pode gerar ganhos expressivos que, nos casos que avaliamos, podem chegar, frequentemente, a mais de 90% de redução de custos.

4.1.6. Experimentos computacionais adicionais

A partir da observação de que algumas instâncias demandam tempos computacionais elevados para serem processadas e fornecer resultados de boa qualidade, alguns experimentos adicionais foram avaliados com foco em melhorar os resultados, principalmente nas instâncias maiores ($T = 6$, por exemplo).

Inicialmente testamos uma estratégida de relaxar e fixar (*Relax and Fix*, ver Wolsey (1998)) as variáveis binárias para obter soluções melhor qualidade nas instâncias onde os modelos não foram capazes de obter resultados melhores que os obtidos na prática. Essa estratégia foi utilizada com sucesso por Ferreira et al. (2009) no contexto industrial de produção.

A heurística consiste em dividir os conjuntos de variáveis binárias em sub-conjuntos. A cada iteração as variáveis de apenas um desses sub-conjuntos são definidas como inteiras enquanto as demais são relaxadas e o podem assumir valores contínuos. A idéia é que, um modelo com número menor de variáveis inteiras, seja mais fácil de ser resolvido. A cada iteração, o critério utilizados para a fixação das variáveis foi que as variávies inteiras com valores iguais a um são fixadas e passam a fazer parte da solução enquanto seus respectivos *jobs* são retirados dos dados de entrada do modelo.

Na prática os testes foram realizados com os dados do cenário 3, pois, nos testes computacionais anteriores, essa instância apresentou os piores resultados.

As variáveis foram divididas em 3 sub-conjuntos, o primeiro composto pelo primeiro e segundo períodos ($t = 1$ e $t = 2$), o segundo sub-conjunto pelas variáveis relativas ao terceiro e quarto períodos ($t = 3$ e $t = 4$) e o último sub-conjunto pelas variáveis relativas ao quinto e sexto períodos ($t = 5$ e $t = 6$).

Assim, a heurística realizou três iterações relaxando e fixando variáveis.

Na primeira, as variáveis do primeiro conjunto foram mantidas binárias e as demais foram relaxadas. As variáveis desse conjunto foram fixadas com os valores a apresentados na solução obtida com o tempo de duas horas. Na segunda iteração, as variáveis do segundo conjunto foram mantidas como binárias e apenas as do terceiro conjunto foram relaxadas (dado que as variáveis do primeiro conjunto já haviam sido fixadas). Com a solução obtida em duas horas as variáveis do segundo conjunto foram fixadas. Na terceira iteração, apenas com as variáveis do terceiro conjunto, pois, as demais haviam sido fixadas nas iterações anteriores, não houve relaxação e o problema foi resolvido para fixá-las.

Entretanto, os resultados obtidos nos primeiros testes não foram satisfatórios. Mesmo com algumas variáveis relaxadas, o modelo apresentou dificuldade para solucionar o problema no tempo estabelecido (de duas horas). Poucas variáveis foram fixadas na primeira e segunda iteração e houve muitos atrasos que incorreram em custos elevados.

Os resultados obtidos podem ser avaliados na Tabela 4.18. É possivel notar que, com a aplicação da heurística *Relax and Fix* não houve custo com realização de horas extras, entretanto, muitos *jobs* não foram produzidos dentro do prazo, o que gerou custos elevados de *backlog*.

Tabela 4.18: Resultados com a aplicação da Heurística *Relax and Fix* com o modelo MCHEAAPTF

		MCHEAAPTF					MCHEAAPTF *R&F*				
T	J	v^1	v^2	o($)	h($)	total($)	v^1	v^2	o($)	h($)	total($)
6	41	0,15	0,00	4,95	6750,00	6754,95	0	0	0	7350,00	7350,00

Outro teste adicional foi realizar o pré-processamento das instâncias objetivando reduzir seu tamanho e complexidade. Primeiramente verificamos quais *jobs*, se processados imediatamente em sequência não demandavam trocas. Para essa análise é criada uma matriz $|J| \times |J|$ com o valor de trocas entre dois *jobs* que são processados um imediatamente após o outro. Nos casos em que os valores de trocas necessárias são iguais a zero (e não estão na diagonal da matriz), os *jobs* foram agrupados. No caso de um *job* não demandar trocas ao ser processado imediatamente após o processamento de mais de um *job*, o primeiro deles foi o escolhido. Todos os jobs são analisados e, enquanto for possivel, os *jobs* são agrupados. Esse critério difere do critério proposto por Bard (1988) uma vez os *jobs* podem ter *due dates* distintos. Portanto esse critério de agrupamento não garante otimalidade da solução.

Para agrupar os *jobs* parâmetros precisam ser atualizados. Conforme descrito anteriormente, *jobs* possuem tempos de processamento. Quando dois ou mais *jobs* são agruapados, seus tempos de processamento precisam ser somados. Outro parâmetro que precisa ser revisto são os prazos de término. Quando dois *jobs* que possuem prazos de término diferentes são agrupados, o prazo menor torna-se o prazo para o novo *job*.

Os custos de atraso então são ajustados para evitar distorções, ou seja, se temos dois *jobs* (j_1 e j_2) que serão agrupados e um deles tem prazo de término em $t = 1$ ($d_1 = 1$) e o segundo em $t = 4$ ($d_2 = 4$), então, o prazo de término do novo *job* (j_3) será $t = 1$ ($d_3 = 1$). Nesse caso, os custos de *backlog* para o primeiro, segundo e terceiro dias de atraso serão relativos apenas ao primeiro *job* (j_1), pois, nesses casos, o segundo *job* (j_2) ainda estará sendo fabricado dentro do prazo. A partir do quarto dia de atraso do novo *job* (j_3) os dois *jobs* originais estarão atrasos, assim, os custos de *backlog* será relativos a ambos (j_1 e j_2).

Durante o pré-processamento, quando *jobs* são agrupados, pode ocorrer que eles pertençam a produtos distinos. Isso significa que o novo *job* que será criado a partir do agrupamento pertencerá a dois produtos diferentes. Na prática existe uma matriz binária que indica com valor igual a um os casos em que um *job* pertence a um produto e com zero, caso contrário. Quando *jobs* de dois produtos distintos são agrupados, o novo *job* terá dois valores iguais a um nas colunas quer representam os produtos aos quais ele pertence.

Passada essa primeira etapa (quando os *jobs* agrupados não demandavam trocas entre si), o agrupamento de *jobs* pode ser realizado entre *jobs* cujos valores de trocas são baixos, por exemplo: uma ou duas trocas. Nesses casos, além das atualizações descritas anteriormente, será necessário criar uma nova matriz de ferramentas por *job*, onde as ferramentas demandas por cada novo *job* (denotadas pelo valor um na matriz binária) serão a união das ferramentas demandas pelos *jobs* originais. É importante notar que, mesmo que dois *jobs* requeiram um valor baixo de trocas para serem processados em sequência, é condição básica para o seu agrupamento que a união das ferramentas demandas pelos dois não exceda a capacidade da caixa de ferramentas. Por exemplo, mesmo que o critério de agrupamento seja ter um número de trocas igual a no máximo uma troca, caso um dos *jobs* originais requeira 10 ferramentas (que no nosso caso é o limite de capacidade da caixa de ferramentas), o mesmo não poderá mais ser agrupado, pois, não haverá espaço na caixa de ferramentas para receber mais ferramentas.

Nos testes que foram realizados para avaliar a diferença no desempenho dos modelos em instâncias que foram pré-processadas, foram realizadas iterações enquanto foi possível agrupar *jobs* com no máximo uma troca entre eles. Os resultados obtidos em dados retirados da instância de seis períodos ($T = 6$) do cenário 1 podem ser vistos na Tabela 4.19 que apresenta os valores das soluções obtidas com os modelos aplicados aos dados originais do problema, com a aplicação do pré-processamento com agrupamento de *jobs* que não demandava trocas para serem processados em sequência ($PP0$) e com a aplicação de agrupamentos entre *jobs* que demandavam até uma troca ($PP1$). Com a aplicação do primeiro pré-processamento ($PP0$) a instância, que continha 44 *jobs*, foi reduzida para 33 *jobs*. Com a aplicação do segundo pré-processamento ($PP1$) a instância foi reduzida para 24.

Os resultados demonstraram que agrupar os *jobs* conforme os critérios apresentados não garante melhoria na solução. Na medida em que os *jobs* são agrupados, os tempos de processamento sobem consideravelmente, o que pode ser um fator dificultador na obtenção de soluções de boa qualidade. Nos testes realizados com o modelo MCHEA houve redução sensível na qualidade das soluções obtidas após duas horas de processamento tanto para agrupamentos entre *jobs* que não demandavam trocas como para agrupamentos entre *jobs* que demandavam até uma troca.

Tabela 4.19: Resultados com a aplicação do pré-processamento do dados

Modelo	Dados originais	$PP0$	$PP1$
MCHEA	166,60	497,10	886,00
MCHEAAP	792,20	4222,00	3557,30
MCHEAAPTF	5800,00	12317,65	6355,00

Após os primeiros testes de pré-processamento uma nova estratégia foi avaliada. Na primeira etapa, apenas *jobs* que não demandavam trocas entre eles foram agrupados. No próximo passo os conjuntos de *jobs* que compõem os produtos foram avaliados. Em cada produto, se existirem *jobs* cuja soma de ferramentas necessárias ao processamento não exceda a capacidade da caixa de ferramentas, os mesmo também são agrupados. Assim, mesmo que um *job* componente de um produto requeira oito ferramentas e outro *job* componente do mesmo produto requeira três ferramentas, se ambos tiverem pelo menos uma ferramenta em comum, os mesmo serão agrupados, pois, todas as ferramentas podem ser alocadas numa caixa de ferramentas com capacidade igual a dez.

Os testes com esse critério de pré-processamento foram realizados com os mesmos dados dos testes anteriores (obtidos a partir do cenário 1). Essa instância tinha 44 *jobs* e, com essa tática, o número de *jobs* caiu para 23. Os dados então foram testados com os modelos MCHEA, MCHEAAP e MCHEAAPTF.

Tabela 4.20: Resultados com a aplicação do pré-processamento do dados

Modelo	Dados originais	Dados pré-processados
MCHEA	166,60	551,00
MCHEAAP	792,20	3691,05
MCHEAAPTF	5800,00	4562,70

Os resultados são apresentados na tabela 4.20. Nos testes realizados, houve perda na qualidade da solução nos testes com os modelos MCHEA e MCHEAAP. Embora o resultado obtido com o modelo MCHEAAPTF tenha custo menor utilizando os dados pré-processados, esse resultado possui custo mais elevado que a sequência realizada na prática.

Os resultados com os teste computacionais complementares não foram satisfatórios. Isso reforça a importância de estabelecer novos critérios tanto para a herística de *Relax and Fix* como para o pré-processamento dos dados.

Capítulo 5

Considerações Finais e Proposições para a Continuidade dos Estudos

5.1. Considerações Finais

Nessa dissertação foi apresentada uma abordagem para o tratamento do problema de minimização de trocas de ferramentas. Algumas características importantes verificadas no ambiente fabril como a possibilidade de trabalhar com um horizonte de planejamento fragmentado em dias, expansão de capacidade pelo emprego de horas extras e penalizações por atrasos foram incorporadas aos modelos para fomentar sua aplicação prática. Embora o problema seja bastante específico, existe um vasto campo para a sua aplicação, na medida que vários processos produtivos apresentam características análogas. Os resultados obtidos através dos testes computacionais demonstram o potencial da pesquisa e sua capacidade de trazer ganhos e maior competitividade aos processos produtivos aplicáveis.

Como propostas para continuidade dos estudos ficam o desenvolvimento de novos modelos para o PMTF que possibilitem a solução de problemas de grande porte; a aplicação de heurísticas baseadas no modelos existentes; a aplicação de modelos para resolução de subproblemas; a alocação dos jobs aos dias através de um método heurístico e o sequenciamento por método exato dos *jobs* em cada dia, introdução de métodos exatos que acelerem a convergência dos modelos apresentados; e o pré-processamento dos dados como forma de simplificar as instâncias e melhorar o tempo computacional necessário para soluciona-las.

Referências Bibliográficas

AlFawzan, M. A. e AlSultan, K. S. (2002). A tabu search based algorithm for minimizing the number of tool switches on a flexible machine. *Computers and Industrial Engineering*, 44:35–47.

Bard, J. F. (1988). A heuristic for minimizing the number of tool switches on flexible machine. *IIE Transactions*, 20:382–391.

Błazewics, J.; Eckera, K.; Pesch, E.; Schmidt, G. e Weglarz, J. (1996). *Scheduling Computer and Manufacturing Processes*. Springer.

Belady, L. A. (1966). A study of replacement algorithms for virtual storage computers. *IBM Systems Journal*, 5:78–101.

Crama, Y.; Kolen, A. W. J.; Oelermans, A. G. e Spieksma, F. C. R. (1994). Minimizing the number of tools switches on a flexible machine. *International Journal of Flexible Manufacturing Systems*, 6:33–54.

Cunha, Jr., J. J. e Souza, M. C. (2009). Minimização de troca de ferramentas numa máquina *CNC*: Aplicação de uma heurística gulosa a um caso real, *Anais do XLI Simpósio Brasileiro de Pesquisa Operacional, Porto Seguro, Brasil*. pp. 842–850, Porto Seguro, Brasil.

Cunha, Jr., J. J. e Souza, M. C. (2010). Sequenciamento de tarefas em máquinas de manufatura flexível para reduzir cstos com horas extras e atrasos de entregas, *Anais do XLII Simpósio Brasileiro de Pesquisa Operacional*. pp. 1201–1212, Bento Gonçalves, Brasil.

Cunha, Jr., J. J.; Souza, M. C. e Yanasse, H. H. (2011). Scheduling jobs on a flexible machine to minimize overtime and weighted tardiness costs, *Proceedings of International Conference on Industrial Engineering and Systems Management*. pp. 121–125, Metz, França.

Dantzig, G. R.; Fulkerson, D. R. e Johnson, S. M. (1954). Solution of a largescale traveling salesman problem. *Operations Research*, 2:393–401.

Ferreira, D.; Morabito, R. e Rangel, S. (2009). Solution approaches for the soft drink integrated production lot sizing and scheduling problem. *European Journal of Operational Research*, 196:697–706.

Gendreau, M.; Hertz, A. e Laporte, G. (1992). New insertion and postoptmimization procedures for the traveling salesman problem. *Operations Research*, 40:1086–1094.

Hertz, A.; Laporte, G.; Mittaz, M. e Stecke, K. (1998). Heuristics for minimizing tool switches when scheduling part types on a flexible machine. *IIE Transactions*, 30:689–694.

Laporte, G. (1992). The travelling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59:231–248.

Laporte, G.; Salazar, J. J. e Semet, F. (2003). Exact algorithms for the job sequencing and tool switching problem. *IIE Transactions*, 35:1–9.

Pinedo, M. L. (2008). *Scheduling. Theory, Algorithms, and Systems*. Prentice Hall.

Resende, M. G. C. e Ribeiro, C. C. (2003). Greedy randomized adaptive search procedures, *in: Glover, F.; Kochenberger, G. (eds.), Handbook of Metaheuristics, Kluwer Academic Publishers*. pp. 219–249.

Tang, C. S. e Denardo, E. V. (1988a). Models arising from a flexible manufacturing machine. part i: Minimization of the number of tool switches. *Operations Research*, 36:767–777.

Tang, C. S. e Denardo, E. V. (1988b). Models arising from a flexible manufacturing machine. part ii: Minimization of the number of switching instantes. *Operations Research*, 36:767–777.

Wolsey, L. A. (1998). *Integer Programming*. John Wiley & Sons.

Yanasse, H. H. (2007). Limitante inferior para o problema de minimizar o número de trocas de ferramentas, *Anais do XXXIX Simpósio Brasileiro de Pesquisa Operacional*. pp. 1886–1892, Fortaleza, Brasil.

Yanasse, H. H.; Rodrigues, R. C. M. e Senne, E. L. F. (2009). Um algoritmo enumerativo baseado em ordenamento parcial para resolução do problema de minimização de trocas de ferramentas. *Gestão Produção*, 16:370–381.

Yanasse, H. H.; Senne, E. L. F. e Rodrigues, R. C. M. (2008). An improved partial ordering scheme for solving the minimization of tool switches problem, *XIV Latin Ibero-American Congress on Operations Research*. Cartagena, Colômbia.