Universidade Federal de Santa Catarina
Programa de Pós-Graduação em Engenharia de Produção

RICARDO MARTINS CURY

UMA ABORDAGEN DIFUSA PARA O PROBLEMA DE FLOW-SHOP SCHEDULING

Florianópolis
Abril 1999

Universidade Federal de Santa Catarina
Programa de Pós-Graduação em Engenharia de Produção

RICARDO MARTINS CURY

UMA ABORDAGEM DIFUSA PARA O PROBLEMA DE FLOW-SHOP SCHEDULING

Tese submetida à Universidade Federal de Santa Catarina para a obtenção do Grau de Doutor em Engenharia

Orientador: Prof. Fernando A. O. Gauthier

Florianópolis
Abril 1999

UMA ABORDAGEM DIFUSA PARA O PROBLEMA DE FLOW-SHOP SCHEDULING

RICARDO MARTINS CURY

Esta tese foi julgada adequada para obtenção do Título de Doutor em Engenharia, Especialidade em Engenharia de Produção e aprovada em sua forma final pelo Programa de Pós-Graduação

Prof. Ricardo Miranda Barcia – Ph.D
Coordenador do PPGEP

Banca Examinadora:

_____	_____
Orientador: Prof. Fernando Gauthier, Dr.	Membro: Prof. Suresh K. Khator, Ph. D.
_____	_____
Moderador: Prof. Flávio Lapolli, Dr.	Membro: Prof. Marco A. B. Cândido, Dr.
_____	_____
Membro: Profª Edis Mafra Lapolli, Dr.	Membro: Prof. Alejandro Martins, Dr.

DEDICATÓRIA

À minha querida companheira Martha pelo tempero em minha vida,
À meus pais e irmãos pelo apoio incondicional,
À minha avó Leonor (in memorian) pelo estimulo à carreira acadêmica.

AGRADECIMENTOS

Aos meus pais, cujo apoio nunca deixou de existir dando-me forças a continuar.

Aos colegas do Departamento de Administração da Universidade Estadual de Maringá, pelo apoio durante o desenvolvimento desta tese.

À CAPES e ao CNPq que financiaram esta pesquisa.

Aos meus orientadores, Prof. Dr. Fernando A. O. Gauthier e Prof. Suresh K. Khator, Ph.D.que foram fundamentais para a realização desta pesquisa junto a Universidade Federal de Santa Catarina e University of South Florida.

Ao meu grande amigo Rodrigo Becke Cabral pelos imprescindíveis conhecimentos em C++.

ABSTRACT

Earlier studies on scheduling problems usually have been conducted by treating parameters and constraints as exact or crisp. This fuzzy approach considers the fuzzification of parameters such as processing time, due dates and precedence relations between jobs. They are considered uncertain or subjected to preferences, so fuzzy mathematics is used. The motivation for a fuzzy approach has two main reasons: first, the customer and/or manufacturer have a satisfaction degree; second, the fuzzy precedence relation can reflect more than technical ordering with respect to production. This satisfaction degree is considered as a membership function. A fuzzy Branch and Bound Algorithm is presented. The branch strategy as well bounding strategy and feasibility tests are detailed. The model was applied to three problems and results are shown. It could be concluded that the fuzzy approach is suitable to scheduling problems.

RESUMO

Estudos anteriores sobre problemas de scheduling foram usualmente conduzidos tratando parâmetros e restrições como exatos ou crisp. Esta abordagem difusa considera a fuzificação de parâmetros tais como tempo de processamento, datas de entregas e relações de precedência entre tarefas. Eles são considerados incertos ou sujeitos a preferências, logo, matemática difusa é usada. A motivação para a abordagem difusa tem duas razões principais: primeiro, o cliente e/ou fabricante tem um grau de satisfação; segundo, as relações de precedência difusas podem refletir mais que um ordenamento técnico com respeito a produção. Este grau de satisfação é considerado como uma função de pertinência. Um algoritmo difuso de Branch-and-Bound é apresentado. A estratégia de ramificação, bem como, a estratégia de poda e testes de viabilidade são apresentados. O modelo foi aplicado a três problemas e os resultados são mostrados. Pode ser concluído que a abordagem difusa é eficaz para problemas de scheduling.

Sumário

Lista de Figuras

Lista of Tabelas

Lista de Equações

1 Introdução

Este trabalho descreve o uso do algoritmo de Branch and Bound (BB) aplicado ao problema de flow shop scheduling, para minimizar uma função de penalidade para atrasos e adiantamentos (lateness). A motivação para a minimização desta função de penalidade esta no fato que vários sistemas produtivos estão sob influência de custos de dois tipos básicos. Os custos para manter um ítem no inventário, ou os custos por não ser capaz de acabar um item a tempo, de acordo com algum plano gerencial. Neste capítulo são apresentados o conhecimento básico para o entendimento do problema, os objetivos do trabalho e sua estrutura.

1.1 Contexto

Problemas de scheduling podem ser entendidos como alocação de recursos para tarefas com respeito ao tempo. Estes problemas tem sido estudados na literatura de pesquisa operacional desde o pioneiro trabalho de Johnson no início dos anos 50.

Nos últimos 40 anos algum progresso foi feito em termos de métodos de solução, métodos de classificação do problema e modelagem. Medidas de performance usuais em problemas de scheduling são o nível de utilização dos recursos de produção, o percentual de tarefas atrasadas, o atraso médio ou máximo para um conjunto de tarefas, e o tempo de fluxo, médio e máximo, para um conjunto de tarefas. Para estas medidas existem funções associadas, como função de penalidade que será discutida neste trabalho.

Os problemas de job shop são os mais gerais problemas de scheduling de produção em qualquer classificação e também os de mais difícil solução. Aqui não existem restrições nos passos de processamento de uma tarefa e roteiros alternativos para uma tarefa podem ser permitidos. Adicionalmente, não existem restrições aos requisitos associados com cada tarefa, assim, uma tarefa pode requer processamento em qualquer subconjunto de processadores em qualquer ordem concebível [1].

Tradicionalmente, existem três tipos de abordagem para os problemas de job shop: regras de prioridade, otimização combinatorial e análise de restrições. A primeira abordagem é computacionalmente eficiente, mas não existe garantia com relação a solução obtida, especialmente se restrições temporais devem ser respeitadas. É sabido que algumas dessas regras tem melhor desempenho que outras, em termos médios. As abordagens de otimização são mais rigorosas, mas não são tratáveis em problemas grandes. A terceira abordagem procura um conjunto de soluções viáveis que preenchem várias restrições temporais e tecnológicas, deixando a decisão final para o usuário [2].

O scheduling em flow shops é menos complexo que o scheduling em job shops. De fato, ele é um caso especial de job shop onde as máquinas são numeradas em ordem crescente. Para cada tarefa considerada, a operação K é realizada em uma máquina de numeração mais alta que a operação J se J precede K.

O permutation flow shop é caracterizado pelo fato que a sequência de operações tem que ser mantida em todas as máquinas na planta de fabricação. Portanto, se a máquina 1 realiza a operação J antes da operação K, todas as máquinas no sistema vão realizar operação J antes da operação K.

Desde o trabalho de Graves [1], a questão da incerteza começou a ser considerada. Um frequente comentário em muitos ambientes de fabricação é que não existe um problema de scheduling mas um problema de rescheduling. “Pode ser fácil construir um programa de produção (schedule); o que é difícil é a constante revisão requerida pela dinâmica do ambiente. Logo, é importante entender como implementar um programa quando as condições da planta são incertas. Em particular, existe uma necessidade de entender a robustez do programa. Gerentes de manufatura desejam métodos de programação que ou explicitamente reflitam a natureza incerta da informação disponível, ou forneçam alguma garantia de insensibilidade do programa a futura informação”.

Neste sentido, BLAZEWICZ et al [3] sugerem o uso de lógica difusa como uma maneira de obter melhores programas reativos. As duas principais razões para o uso de lógica difusa são a modelagem do sistema usando lógica difusa e regras difusas.

WANG et al [4] justificam o uso da abordagem difusa em problemas de scheduling como uma maneira de fazer-se compromissos entre critérios elementares. Por exemplo, o balanceamento de cada regra de despacho elementar pode ser feito pela modificação da base de regras de decisão difusa. O ponto de vista deles considera o sistema de manufatura como um sistema multi-critério e multi-objetivo.

De acordo com McCAHON et al [5] o tempo de processamento geralmente pode ser estimado como estando dentro de um dado intervalo. Os gerentes raramente sabem a distribuição de probabilidade dos tempos de processamento dentro desse intervalo. Por causa desta característica de estimação do intervalo, a representação do tempo de processamento da tarefa pode ser mais realisticamente e naturalmente alcançada através do uso de um número difuso.

Além do tempo de processamento, outros parâmetros e dados podem ser considerados incertos ou melhor representados através de graus de satisfação ou requerem estimativas com alto custo para obter a precisão necessária para um tratamento crisp. Eles são a data de liberação (release date), datas de entrega (due dates) e relações de precedência (precedence relations).

FAGIER [6] afirma que scheduling sobre longos horizontes, considerando restrições temporais, tais como data de liberação e data de entrega como compulsórios, podem levar a rejeição de um schedule eficiente, ainda que a violação destas restrições seja insignificante com relação a precisão dos limites realistas de previsibilidade. Embora, restrições frequentemente provem ser mais ou menos relaxáveis ou sujeitas a preferências; isto é particularmente verdade para restrições de data de entrega em scheduling. Ela declara que conjuntos difusos e teoria da possibilidade tem surgido como uma aplicável estrutura para a representação de restrições flexíveis.

ISHIBUCHI et al [7] abordou o problema de programação da produção usando como critério de minimização o número de tarefas atrasadas onde a data de entrega de cada tarefa é um número difuso. Neste problema o nível de satisfação gradualmente decresce depois que a data de entrega tenha sido excedida. A função que representa este nível de satisfação é uma função de pertinência não linear.

Um exemplo realista da aplicação de datas de entrega difusas é mostrado em HAN et al [8]. Eles mostraram como uma companhia manufatureira poderia usar este tipo de parâmetro difuso. Bens para exportação devem ser produzidos estritamente dentro da data de partida do navio para seu destino, desde que, se atrasado, a companhia pode ter que esperar um mês ou mais até o próximo navio disponível. Embora, para bens de consumo doméstico o uso de entregas expressas pode cobrir pequenos atrasos ainda que este custo extra reduza o nível de satisfação da companhia.

Considerando-se, especificamente a questão de relações de precedência vários autores [2, 3, 6, 9] por exemplo, trabalharam ou consideraram relações de precedência em seus estudos sobre scheduling difuso. Em síntese, eles concordam que relações de precedência podem ser quebradas permitindo algum relaxamento de restrições que deveria ser avaliado através de uma abordagem difusa ou até linguística.

TURKSEN [10], KURODA et al [11], GRABOT et al [12], e CUSTODIO et al [13], declaram que os problemas de scheduling podem ser mais amplamente vistos como um processo de decisão difuso. O tomador de decisão em situações reais em sistemas de manufatura está interessado em soluções viáveis, em alguns casos usando variáveis linguísticas e, considerando incertezas.

Neste trabalho, a lógica difusa é usada para operar com “intervalos de confiança possibilísticos” que representam janelas temporais. Geralmente um problema de scheduling é composto por um conjunto de tarefas que devem ser completadas num intervalo de tempo que tem uma data de início e uma data de entrega. Assim, o problema é como inserir as tarefas dentro dessa janela, considerando um dado conjunto de restrições adicionais.

Problemas de programação da produção podem modelar uma constelação de diferentes cenários. Entre as mais importantes metas está a minimização dos custos de se obter as tarefas prontas considerando-se penalidades por atrasos e adiantamentos. Portanto, o objetivo geral desse trabalho é modelar a programação da produção considerando-se as incertezas e preferências inerentes nesse sistema, consequentemente, de forma mais próxima a realidade encontrada em sistemas produtivos. Este objetivo geral nos leva aos seguintes objetivos específicos:

- Mostrar que a abordagem difusa pode generalizar a abordagem crisp;

- Mostrar que esta maneira de modelar e resolver problemas de scheduling é mais próxima ao processo de raciocínio humano do que os métodos tradicionais de pesquisa operacional;

- Mostrar que o modelo minimiza o custo de tarefas atrasadas e adiantadas em cada possível sequência em problemas de flow shop scheduling;

- Mostrar que a função de penalidade de custo usada como função objetivo é um parâmetro de decisão que avalia a viabilidade de programas de produção; e

- Mostrar que diferentes intervalos de confiança possibilísticos podem ser usados para projetar o custo total de um programa de produção.

1.2 Organização da Tese

Este trabalho está organizado em seis capítulos. O capítulo 1 introduz o conceito geral de scheduling difuso e os objetivos do trabalho. O capítulo 2 traz a revisão bibliográfica sobre os elementos difusos em problemas de scheduling e as técnicas de Branch and Bound. Na verdade, a revisão bibliográfica mostra alguns conceitos da formulação difusa de problemas de scheduling que vão além do escopo do modelo proposto. O capítulo 3 apresenta o modelo proposto, sua análise e design usando o paradigma de orientação a objetos. O capítulo 4 traz os experimentos realizados e seus resultados. O capítulo 5 discute a modelagem abordada e os resultados obtidos. O capítulo 6 apresenta conclusões e recomendações para futuras pesquisas.

2 Revisão Bibliográfica

2.1 Introdução

Considerando-se os objetivos dessa dissertação alguns conceitos fundamentais devem ser examinados antes de se abordar a solução proposta. Logo, este capítulo explora conceitos básicos e faz algumas conclusões parciais. Os conceitos são: elementos difusos em problemas de programação da produção e Branch and Bound.

O objetivo é mostrar que a abordagem difusa para problemas de programação podem ser vistos como uma extensão natural da abordagem crisp e ela modela o ambiente de manufatura de uma forma muito melhor. Esta modelagem é mais próxima a forma que programadores humanos tomam suas decisões nos casos de scheduling preditivo e reativo.

2.2 Elementos Difusos em Problemas de Scheduling

Neste trabalho problemas de scheduling são entendidos como tendo dois grupos básicos de elementos onde a teoria de conjuntos difusos pode ser aplicada. O primeiro grupo abrange variáveis e dados. O segundo grupo consiste das restrições e regras. É lógico que esta classificação depende do sistema sendo estudado, como também do método de solução. Esta classificação pode ajudar-nos a entender os componentes difusos em problemas de scheduling.

2.2.1 Variáveis e Dados Difusos

O conceito de conjuntos difusos lida com a representação de classes cujos limites não são exatamente determinados. Em um conjunto difuso alguns elementos que são considerados como “marginais” ou “menos aceitáveis” recebem um grau de pertinência que é intermediário entre 0 e 1. Quando um conjunto difuso é usado para representar o valor de uma variável, o grau de pertinência atrelado a variável expressa o quão possível é, que ela assuma aquele valor, DUBOIS [14].

Quando considerados sob incerteza, os problemas de scheduling podem ter diferentes elementos que podem ser fuzificados. Os mais intuitivos são a data de liberação para a fabricação (release date) e a data de entrega (due date). A teoria dos conjuntos difusos é uma ferramenta capaz de representar restrições temporais flexíveis em problemas de scheduling [6].

Nas próximas seções, parâmetros difusos serão descritos e discutidos em mais detalhes. Datas de entrega, tempos de processamento e relações de precedência serão apresentados como uma extensão natural desses parâmetros em sua visão crisp.

2.2.1.1 Datas de Entrega Difusas

A data de entrega difusa é caracterizada por uma tarefa que deve ser preferencialmente completada antes de sua data de entrega d_j^{inf} ou tão cedo quanto possível depois dessa data de entrega. O limite de aceitação pelo cliente é a data de entrega d_j^{sup}, depois disso a encomenda não é válida. Assim como a data de entrega, a data de liberação pode ser vista da mesma maneira. É melhor começar a tarefa próximo a última data que ela pode ser completada r_j^{sup} mas a data de liberação pode ocorrer antes desta data r_j^{inf}, figura 2-1 [6].

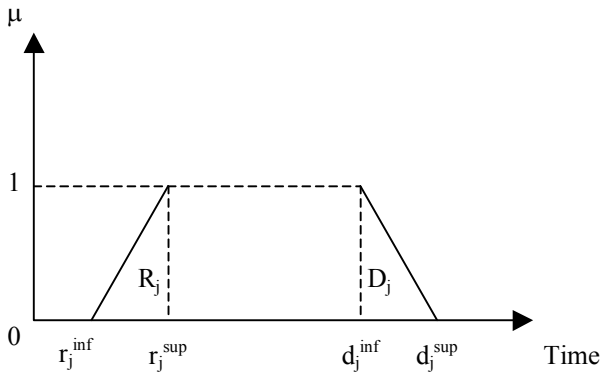

Figura 2-1. Datas de liberação e entrega difusas.

O número triangular difuso R_j representa o grau de satisfação para a data de início da tarefa j. O número triangular difuso D_j representa o grau de satisfação para a data de término (entrega) da tarefa j.

KURODA et al [11] introduziram aspectos difusos em datas de entrega (figuras 2-2 e 2-3). O tempo de término da tarefa j é definido C_j. Para a data de entrega d_j define uma função de pertinência C_j que é um caso especial da equação 2-1 quando a constante positiva u = 1.

ISHIBUCHI et al [7], abordou a função de pertinência de datas de entrega como uma função não linear. Eles declaram que existem muitas situações onde o grau de satisfação rapidamente decresce, como na figura 2-2(a), ou decresce lentamente , como na figura 2-2(b). A equação 2-1 representa estas funções de pertinência não lineares.

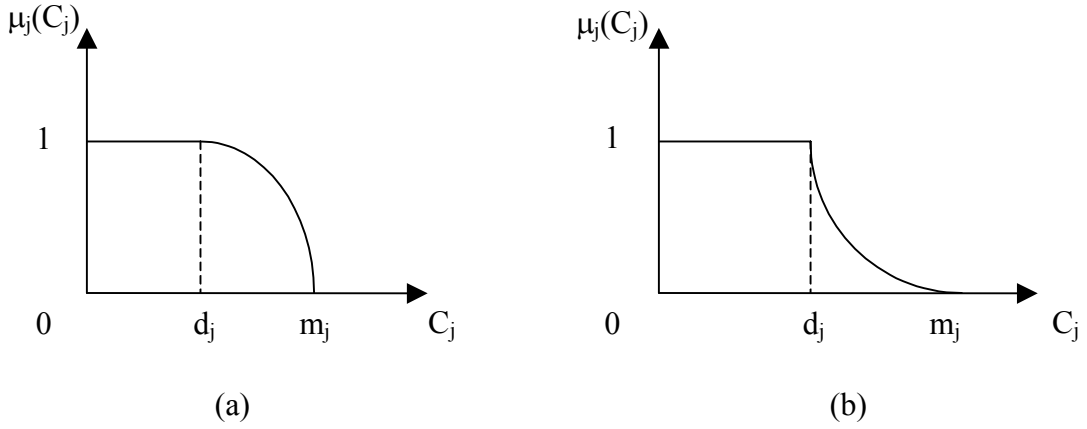

Figura 2-2. Função de pertinência não-linear para data de entrega: (a) côncava, e (b) convexa

$$\mu_j\left(C_j\right) = \begin{cases} 1 & \text{if } 0 < C_j \le d_j \\ \left\{1 - \dfrac{\left(C_j - d_j\right)}{\left(m_j - d_j\right)}\right\}^u & \text{if } d_j < C_j \le m_j \\ 0 & \text{if } C_j > m_j \end{cases} \qquad (2\text{-}1)$$

A constante positiva u é igual a 1 quando a função de pertinência é linear (figura 2-1). Quando u > 1 a função de pertinência é convexa (fig. 2(b)). Quando a

função de pertinência é côncava 0 < u < 1 (fig. 2(a)). Detalhes da determinação da função de pertinência e constante u podem ser vistos em ISHIBUCHI et al [7].

De acordo com ISHIBUCHI et al [15], desde que a data de entrega difusa de cada tarefa representa o grau de satisfação de um tomador de decisão, a forma de sua função de pertinência deveria ser escolhida conforme a preferência do tomador de decisão. Diferentes funções de pertinência podem ser apropriadas para cada problema e cada tarefa.

Em um ambiente de fabricação Just-In-Time (JIT) o uso de funções de pertinência trapezoidais (figura 2-3) para datas de entrega pode representar melhor a preferência do tomador de decisão desde que, se uma tarefa fica pronta muito adiantada, as premissas do ambiente JIT não são respeitadas. Desta forma, a função de pertinência pode ser escrita como na equação 2-2.

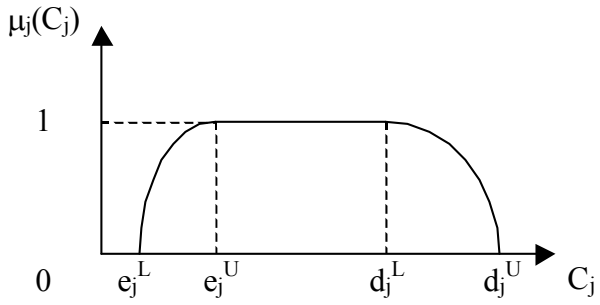

Figura 2-3. Função de pertinência trapezoidal para data de entrega

$$\mu_j(C_j) = \begin{cases} 0 & \text{if } C_j \le e_j^L \\ \left[1 - \left(e_j^U - C_j\right) / \left(e_j^U - e_j^L\right)\right]^{\mathrm{u}} & \text{if } e_j^L < C_j < e_j^U \\ 1 & \text{if } e_j^U \le C_j \le d_j^L \\ \left[1 - \left(C_j - d_j^L\right) / \left(d_j^U - d_j^L\right)\right]^{\mathrm{v}} & \text{if } d_j^L < C_j < d_j^U \\ 0 & \text{if } C_{\mathrm{j}} \ge d_j^U \end{cases} \qquad (2\text{-}2)$$

Onde u e v são constantes positivas tais que u = v = 1 para funções de pertinência lineares ou u ≠ 1 e/ou v ≠ 1 para funções de pertinência não-lineares.

Um exemplo de preferência do tomador de decisão relativo a data de entrega difusa foi mostrado na introdução da tese (companhia com demanda doméstica e para exportação). Cada uma daquelas situações pode ser modelada usando diferentes formas para suas funções de pertinência.

Em problemas de scheduling, as datas de entrega tem um forte peso na determinação de soluções viáveis para o problema de sequenciamento. Quando as datas de entrega mudam, a natureza das relações de precedência entre operações, num mesmo recurso, podem ser alteradas. Considerando a janela temporal definida previamente, soluções viáveis impõem uma relação de precedência particular entre operações. Representando o problema desta forma, o programa pode ser construído pela busca por relações essenciais [19]. De fato, esta abordagem é um método "quase-ótimo" para problemas de scheduling. Ele tem sido usado objetivando o desenvolvimento de um sistema de apoio à decisão para ajudar os usuários a escolher um conjunto de datas de entrega e sugerir a solução viável correspondente. Esta abordagem foi apresentada por ERSCHELER et al [20] e será detalhada nas seções seguintes.

Através da propagação das restrições de data mais cedo de início e data mais tarde de entrega, é possível definir uma janela temporal para cada operação, correspondendo ao intervalo de tempo durante a qual a operação tem que ser realizada. O programa de produção resultante pode incluir conjuntos de operações permutáveis, o qual pode ser usado como um grau de liberdade quando mudanças ocorrem [12].

A função de pertinência da data de entrega difusa é denotada por $\mu_j(C_j)$, que representa o grau de satisfação de um tomador de decisão quando o tempo de término daquela tarefa é C_j. Assim, quando as n tarefas são processadas na ordem de x, o grau de satisfação para o tempo de término $C_{xk}(x)$ da k-ésima tarefa x_k é representado como $\mu_{xk}(C_{xk}(x))$. Consequentemente, o problema de encontrar a sequênca x das n tarefas que maximizam o mínimo grau de satisfação S_{min} é:

$$\text{Maximize } S_{min}(x) = \min\left\{\mu_{xk}\left(C_{xk}(x)\right): k = 1, 2, \ldots, n\right\} \qquad (2\text{-}3)$$

O grau de satisfação das n tarefas pode ser considerado como outro critério de programação. Logo, o problema é encontrar a sequência x que maximiza o grau de satisfação total S_{sum}.

$$\text{Maximize } S_{sum}(x) = \sum_{k=1}^{n} \mu_{xk}\left(C_{xk}(x)\right) \tag{2-4}$$

Vários autores concordam que deve existir uma função de pertinência associada com cada tarefa que descreve o grau de satisfação com respeito ao tempo de término.

HAN et al [8] declaram que nem as funções de pertinência não-lineares são suficientes para descrever uma situação verdadeira de datas de entrega flexíveis. Eles sugerem que para generalizar um problema de scheduling ordinário para um modelo mais flexível outros fatores tais como tempos de processamento podem também ser fuzificados. Este fator será explorado na próxima seção.

2.2.1.2 Tempos de Processamento Difusos

O tempo de processamento pode ser abordado como um número difuso. Primeiro, porque sua duração pode estar sujeita a preferências quando eles são variáveis de decisão sob controle. Segundo, a duração pode estar parcialmente fora de controle e serem incertas.

Para uma dada operação ser otimamente realizada valores ideais para parâmetros de ajuste existem. Existe um "trade-off" entre a duração de uma operação e a qualidade do processamento. Quanto mais curta a duração, melhor para o preenchimento das restrições de scheduling, embora, o valor ótimo de ajuste dos parâmetros pode levar a um tempo de processamento mais longo garantindo uma melhor qualidade de processamento [2].

O grau de pertinência da duração reflete a qualidade do processamento. Nesse sentido, é possível selecionar tempos de início primeiro, e então escolher a ótima duração de acordo com esses tempos de início.

Na segunda interpretação, duração difusa é entendida como parâmetro parcialmente conhecido, resultado de possíveis perturbações e pode novamente ser representado como uma distribuição de possibilidade. Na verdade, distribuições de possibilidade podem ser usadas na modelagem de incertezas como também na modelagem de preferências. Tudo depende se a variável x atrelada a distribuição de possibilidade π_x é controlável ou não. Do contrário, se x não é controlável, então π_x modela uma incerteza sobre o valor que x assumirá.

Se dados estatísticos sobre o tempo de processamento estão disponíveis, o modelo estocástico é perfeitamente justificável. Considerando muitos casos onde dados estatísticos estão fora de alcance ou é duvidoso assumir a repetibilidade das durações, alguma informação sobre que duração é mais plausível do que outra é frequentemente disponível. Esta informação qualitativa sobre parâmetros incertos pode ser modelada por meio de distribuições de possibilidade. Elas podem ser vistas como um tipo de intervalos de confiança variação nos níveis de plausibilidade [2].

O tempo de processamento difuso parece ser o mais polêmico dos parâmetros considerados nesta tese. Não existe acordo entre ISHIBUCHI et al [16], McCAHON et al [17], e TSUJIMURA et al [18]. Os primeiros autores [16] declaram que os tempos de processamento difusos devem ser somados através de um operador de adição que não distorça a forma básica dos números difusos. Os outros usam um operador de máximo que distorce os números trapezoidais difusos que representam os tempos de processamento. Aqui, o conceito de tempo de processamento difuso é apresentado sem consideração sobre que tipo de operador usar, Esta discussão ocorrerá nos capítulos seguintes.

Sabendo-se que tempos de processamento são frequentemente obtidos de dados de tempo padrão eles podem ser incertos e consequentemente representados através de uma distribuição de possibilidade $\pi_{ij}(t)$ [11]. A equação 2-5 e a figura 2-4 mostram essa distribuição de possibilidade.

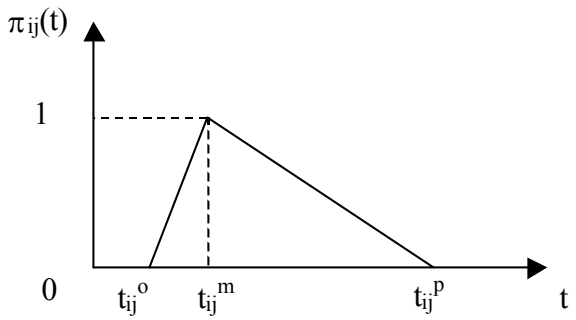

Figura 2-4. Distribuição de possibilidade dos tempos de processamento

$$\pi_{ij}(t) = \left(t_{ij}^{0}, t_{ij}^{m}, t_{ij}^{p}\right) = \begin{cases} \dfrac{t - t_{ij}^{0}}{t_{ij}^{m} - t_{ij}^{0}} & when\ t_{ij}^{0} \le t < t_{ij}^{m} \\ 1 - \dfrac{t - t_{ij}^{m}}{t_{ij}^{p} - t_{ij}^{m}} & when\ t_{ij}^{m} \le t \le t_{ij}^{p} \end{cases} \tag{2-5}$$

Onde t^0_{ij}, t^m_{ij}, e t^p_{ij} são o tempo otimista, o mais provável tempo e o tempo pessimista para realizar a i-ésima operação da tarefa j, respectivamente.

Em adição a KURODA et al [11], vários outros pesquisadores [2, 4, 5, 8, e 18] também estudaram os tempos de processamentos difusos. As mesmas razões básicas para usar tempo de processamento difuso, duração controlável e incerteza justificam suas abordagens e elas são detalhadas como segue.

McCAHON et al [5] utilizaram números difusos normalizados para representar tempos de processamento difusos (normalizados porque os valores das funções de pertinência sempre variam entre zero e um com moda igual a 1).Embora números difusos gerais possam ser usados para representar estes tempos de processamento, frequentemente o aumento em acuracidade do resultado é encoberto pelo aumento do esforço computacional. Adicionalmente, desde que o tempo de processamento difuso é apenas uma estimativa, pode ser difícil se não impossível, construir uma representação geral do número difuso deste tempo de processamento. Números difusos triangulares poderiam também ser usados, como mostra WANG et al [4], e ADAMOPOULOS et al [18], considerando que estes são casos especiais de números trapezoidais difusos.

O número trapezoidal difuso será representado por (a, b, c, d). A função de pertinência vale 1 no intervalo de b até c e, torna-se zero nos dois pontos extremos a e d. A representação gráfica do número trapezoidal difuso é mostrada na figura 2-5, onde $\mu(x)$ é a função de pertinência e x é o tempo de processamento.

Por exemplo, um gerente pode dizer que o tempo de processamento para tarefa A é geralmente de b a c minutos. Mas, devido a outros fatores que não podem ser controlados ou previstos, o tempo de processamento pode ser ocasionalmente lento como d minutos ou tão rápido como a minutos. Esta situação pode ser expressa naturalmente como um número difuso trapezoidal.

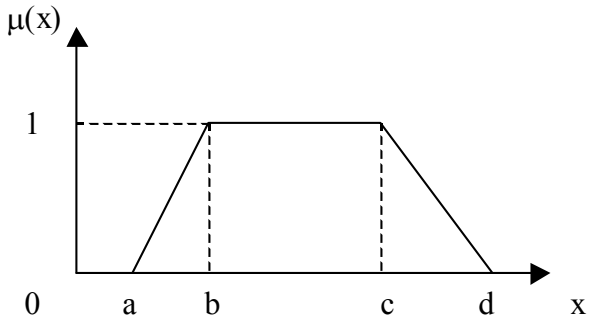

Figura 2-5. Número Difuso Trapezoidal representando tempo de processamento

WANG et al [4] usaram o controlador difuso para fuzzificar variáveis de entrada como tempo de processamento (p) e data de entrega (d). O primeiro passo é representar essas duas variáveis simbolicamente e, associar conjuntos difusos ou funções de pertinência a elas como mostrado na figura 2-6. XS, S, N, L, XL significam muito curto, curto, normal, longo, muito longo; XC, C, N, F, XF significam muito perto, perto, normal, distante, muito distante. Um subsistema no controlador difuso transforma as entradas de valores numéricos em valores difusos via um processo de mapeamento. Considerando a observação p_0 para o tempo de processamento (figura 2-6a), os valores difusos correspondentes são μ_{xs} e μ_s no conjunto XS e no conjunto S, respectivamente. Considerando a observação d_0 para a data de entrega (figura 2-6b) seu valor difuso é μ_{xc} no conjunto XC. Os valores difusos representam quanto o valor observado pertence a cada conjunto associado com a variável. No trabalho deles, o processo de raciocínio avalia as regras de decisão a partir dos valores difusos obtidos.

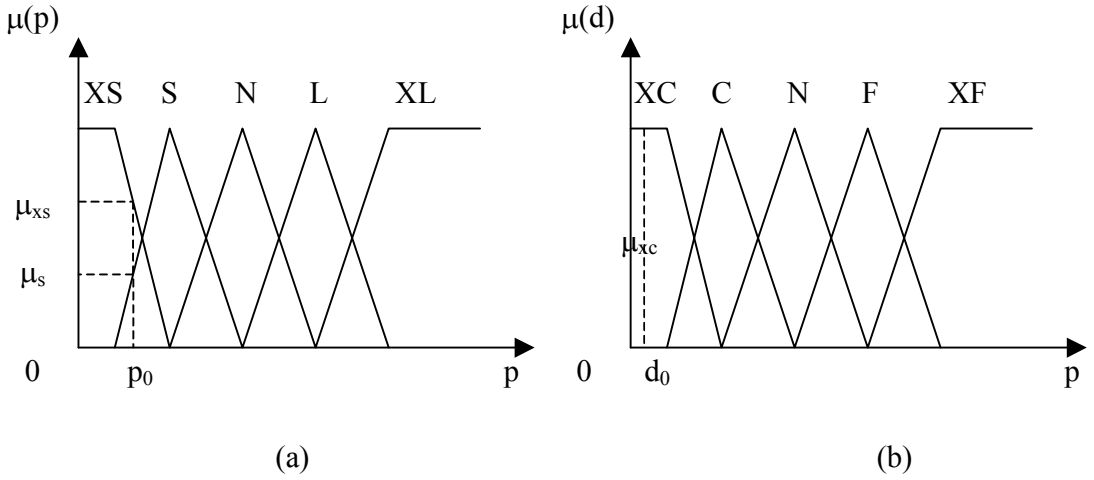

Figura 2-6. Processo de controle difuso para tempo de processamento(a) e data de entrega (b).

ADAMOPOULOS et al [18] também representaram o tempo de processamento através de variáveis linguísticas. A diferença entre o modelo deles e o modelo de WANG et al [4] esta no número triangular difuso usado para as variáveis linguísticas. No modelo de Adamopoulos cada tarefa tem três números difusos triangulares, que representam: (a) o tempo de processamento máximo chamado 'tempo normal', (b) o tempo de processamento mínimo chamado 'crash time' e, (c) o custo por unidade de tempo que se incorre quando se comprime o tempo de processamento abaixo do 'tempo normal'.

Se o sistema produtivo tem capacidade para alocar recursos adicionais para o processamento de uma tarefa, o modelo de Adamopoulos pode perfeitamente ser usado. Eles declaram que uma quantidade mínima de recursos esta alocada se o processamento da tarefa é completado em 'tempo normal'. Da mesma forma, o processamento de uma tarefa em 'crash time' requer o máximo de recursos.

Logo, existe um custo por unidade de tempo G_i que se refere a compressão do tempo de processamento abaixo do 'tempo normal'. A compressão da duração da tarefa pode ter qualquer valor x_i, $0 \leq x_i \leq m_i$, onde m_i é a diferença entre o tempo normal e o 'crash time' Consequentemente, o custo alocado para uma tarefa x_i is $G_i x_i$. Assim, a idéia é determinar o tempo de processamento de tarefas para sequenciar as tarefas na

máquina e determinar a data de entrega comum tal que o valor da função custo seja próxima ao ótimo.

Uma duração mínima t_i^{inf} e uma duração preferida t_i^{sup} pode descrever o possível tempo de processamento da operação O_i. Um número difuso T(i) é definido como R(i) e D(i) na seção 2.2.1.1. O tempo de processamento t_i de cada operação O_i é uma variável de decisão cujos valores permitidos estão limitados pela restrição flexível $t_i \in T(i)$, que deve ser levado em conta no momento do cálculo do grau de satisfação de uma alocação [2].

De acordo com DUBOIS et al [2] existe uma única formulação que lida com tempos de processamento controláveis (flexíveis) como também incertos. Nos casos que os tempos de processamentos são parâmetros de decisão sujeitos a restrições flexíveis, os diferentes tipos de restrições sobre os possíveis valores de tempos de início podem ser expressos por:

$$\begin{aligned}
&\text{precedence constraints } P_{i\to k}\text{: } s_k - s_i \in \left[T(i), +\infty\right) \\
&\text{capacity constraints } C_{i\to k}\text{: } s_k - s_i \in \left[T(i), +\infty\right) \text{ or } s_i - s_k \in \left[T(k), +\infty\right) \\
&\text{release and due dates: } s_i \in \left[R(i), D(i) - T(i)\right]
\end{aligned} \tag{2-6}$$

Considerando (2-6) cada arco conjuntivo $P_{i\to k}$ representa uma restrição de precedência onde O_i deve preceder O_k, por meio de uma inigualdade difusa do tipo $s_k - s_i \geq T(i)$, onde s_k e s_i são os tempos de início. A restrição de capacidade $C_{i\to k}$ define arcos não-conjuntivos ($s_k - s_i \geq T(i)$ ou $s_i - s_k \geq T(k)$) que representam conflitos do tipo O_i precede O_k ou O_k precede O_i. Este tipo de gráfico pode também representar restrições envolvendo tempos de processamento incertos, figura 2-7. Na verdade neste caso, uma relação de precedência $P_{i\to k}$ requer que $s_k - s_i$ pertença ao $]T(i), +\infty)$ conjunto de valores que são necessariamente maiores que T(i), ou em outros termos, que $s_k - s_i$ pertença ao $[\theta(i), +\infty)$, conjunto de valores que são possivelmente maiores que $\theta(i)$, $\theta(i)$ sendo definido por:

$$\theta_i^{\inf} = t_i^*$$
$$\theta_i^{\sup} = \bar{t}_i \qquad \text{for } t_i^* \le t_i \le \bar{t}_i \tag{2-7}$$
$$\mu_{\theta(i)}(t_i) = 1 - \mu_{T(i)}(t_i)$$

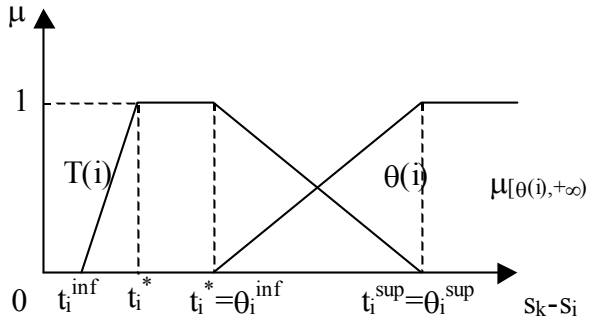

Figura 2-7. Restrições para tempos de processamento incertos

Então, a nova forma das seguintes restrições quando o tempo de processamento é incerto é:

$$\begin{aligned}&\text{precedence constraints } P_{i\to k}\text{: } s_k - s_i \in [\theta(i),+\infty)\\ &\text{capacity constraints } C_{i\to k}\text{: } s_k - s_i \in [\theta(i),+\infty) \text{ or } s_i - s_k \in [\theta(k),+\infty)\\ &\text{release and due dates: } s_i \in [R(i), D(i) - \theta(i)]\end{aligned} \tag{2-8}$$

A principal importância de separar tempos de processamento controláveis (flexíveis) de incertos recai na interpretação dos possíveis valores que o tempo de processamento pode assumir. Em outras palavras, em caso de tempo de processamento flexível, o rótulo de duração difusa no gráfico vem da especificação de preferências e representa os possíveis valores que podem ser alocados para o tempo de processamento. No caso de duração imprecisamente conhecida, este rótulo vem da incerteza a respeito do valor real dessas durações[2].

2.2.1.3 Relações Difusas de Precedência

Antes de examinar-se as relações difusas de precedência, deve-se notar que estas relações são consideradas quando o setup é dependente da sequência de processamento ou um ordenamento técnico deve ser seguido. O problema abordado aqui

não é desse tipo. Entretanto é importante incorporar esta questão na estrutura apresentada aqui no sentido de alcançar-se a generalização dos problemas de scheduling assumida como um dos objetivos específicos desta tese.

Relações de precedência entre operações podem ser definidas como $T_i \ \pi \ T_j$. Isto significa que o processamento de T_i deve ser completado antes que T_j possa ser iniciado. Um conjunto de operações com relações de precedência é usualmente representado como um digrafo no qual os nós correspondem as tarefas e os arcos às restrições das precedências. É assumido que não existem arcos transitivos em grafos de precedências.

CONWAY et al [21] considera situações onde certos ordenamentos são proibidos, ou por restrições tecnológicas ou por alguma política externa imposta. Eles mostram dois principais tipos de questões para relações de precedência: sequências parciais de tarefas (strings) requeridas e, relações de precedências gerais.

Quando os tempos de setup são altamente dependentes da sequência, algumas decisões com respeito a sequência podem ser feitas antes que o programa completo esteja feito. Por exemplo, grupos d tarefas com setups similares podem ser agrupados para um mínimo tempo de troca. O problema aqui é arranjar os grupos para minimizar alguma medida de desempenho.

Em geral é assumido que as n tarefas originais tenham sido agrupadas em k strings com n_1, n_2,…, n_k como o número de tarefas nas várias strings. Considere que uma vez começada, toda a string deve ser processada até o fim. Portanto, deve-se selecionar uma das k! maneiras na qual as strings podem ser sequenciadas.

Neste caso, CONWAY et al [21] declaram que é possível considerar cada string como uma tarefa e usar a formulação básica de scheduling para sequenciar as strings.

Considerando o caso onde um conjunto de tarefas requer certas precedências mas não requerem que as strings sejam processadas sem interrupção, o ordenamento pode ser especificado pelo uso de relações do tipo precede ou diretamente

precede. Cada uma das tarefas pode ser particionada em grupos de operações tais que as operações dentro de um grupo poderiam ser realizadas em qualquer ordem. Embora, todas as operações de um dado grupo tenham que ser completadas antes de qualquer operação do grupo seguinte possa ser realizada.

Se uma probabilidade P é especificada para uma tarefa que começa um novo grupo, é possível medir o grau de flexibilidade da sequência. Isto é feito pelo cálculo para cada tarefa da razão entre o número de sequências permitidas pelo número que seria possível se não existissem restrições de precedência. O resultado é o fatorial do tamanho dos grupos dividido pelo fatorial do número total de operações. Um conjunto de tarefas é caracterizado pela média destas razões para tarefas individuais. Aqui, o uso de uma sequência alternativa simplesmente adia a entrada da tarefa em uma fila congestionada [21].

RARDIN [22] expressa o requisito de precedência que uma tarefa j deva ser completada no processador k antes que a atividade em k' como segue:

$$x_{j,k} + p_{j,k} \leq x_{j,k'} \tag{2-9}$$

Onde $x_{j,k}$ denota o tempo de início da tarefa j no processador k, $p_{j,k}$ é o tempo de processamento de j em k e, $x_{j,k'}$ é o tempo de início da tarefa j no processador k'.

ERSCHLER et al [20] supôs que é possível associar com cada operação (i, j) um tempo de início mais cedo $c_{i\,j}$ e um tempo de término mais tarde $f_{i\,j}$. Para duas operações (i, j), (k, l) tais que $m_{i\,j} = m_{k\,l}$, o intervalo de tempo alocado para o par ordenado de operações [(i, j), (k, l)] pode ser definido por:

$$\Delta_{ij}^{kl} = f_{kl} - c_{ij} \tag{2-10}$$

A relação entre as duas operações do par [(i, j), (k, l)] na sequência, pode ser representada por uma variável binária tal que:

$$X_{ij}^{kl} = \begin{cases} 1 \text{ if (i, j) precedes (k, l)} \\ 0 \text{ if (k, l) precedes (i, j)} \end{cases} \tag{2-11}$$

Pela comparação dos intervalos alocados e a soma dos tempos de processamento das operações, pode-se definir as duas relações seguintes:

$$\begin{aligned} \Delta_{kl}^{ij} < p_{ij} + p_{kl} \Rightarrow X_{ij}^{kl} = 1 \\ \Delta_{ij}^{kl} < p_{ij} + p_{kl} \Rightarrow X_{ij}^{kl} = 0 \end{aligned} \tag{2-12}$$

Quando uma das duas relações é satisfeita é possível ordenar as duas operações correspondentes. Eles chamaram a relação da sequência resultante de (2-12) 'essencial', porque não é possível encontrar um schedule que satisfaça as dadas restrições e não corresponda a essas relações.

ERSCHLER et al [20] também considerou uma operação (i, j) e um conjunto de operações $O_{kl} = \{(k_1, l_1), (k_2, l_2), \ldots, (k_q, l_q)\}$ tais que $m_{i\,j} = m_{kv\,lv}$ v = 1, …, q. É possível associar uma operação fictícia (k, l) ao conjunto de operações O_{kl} tal que:

$$\begin{aligned} & p_{kl} = \sum_{v=1}^{v=q} p_{kvlv}, c_{kl} = \min_v c_{kvlv} \\ & \textit{and} \\ & f_{kl} = \max_v f_{kvlv} \end{aligned} \tag{2-13}$$

Considerando (2-12), é possível estabelecer as seguintes relações:

$$\begin{aligned} & \Delta_{kl}^{ij} < p_{ij} + p_{kl} \Rightarrow \prod_v X_{kvlv}^{ij} = 0 \; or \; \sum_v X_{ij}^{kvlv} = 1 \\ & \textit{and} \\ & \Delta_{ij}^{kl} < p_{ij} + p_{kl} \Rightarrow \prod_v X_{ij}^{kvlv} = 0 \; or \; \sum_v X_{kvlv}^{ij} = 1 \end{aligned} \tag{2-14}$$

Onde Π representa o operador Booleano E e Σ o operador Booleano OU.

Estas relações permitem concluir que necessariamente existe uma fixa relação de sequenciamento entre operação (i, j) e uma das operações do conjunto O_{kl}. Elas chamar-se-ão ‘condicionais’.

Até agora neste trabalho, as relações de precedência foram consideradas relações binárias entre operações. Se a tarefa J_i precede a tarefa J_j, a tarefa J_j só pode ser iniciada depois do término da tarefa J_i. Entretanto, é possível considerar um grau de relacionamento entre tarefas se elas são processadas em processadores diferentes.

ISHII et al [9] declaram que uma relação de precedência difusa poderia relaxar a restrição de precedência em alguns casos. Esta relação poderia refletir um nível de satisfação com respeito a precedência entre tarefas. Eles definiram uma relação de precedência difusa denotada com a função de pertinência μ_{ij} para todos pares de tarefas J_i e J_j, o qual denota um grau de desejo que a tarefa J_i seja processada antes da tarefa J_j. Eles assumem que se $0 < \mu_{ij} < 1$ então $\mu_{ji} = 1$ e se ambos μ_{ij} e $\mu_{ji} = 1$ significa que J_i e J_j são independentes.

VIEGAS et al [19] e outros [2, 7, 8, 9, e 11] declaram que em problemas de scheduling as datas de entrega tem um forte peso na determinação de soluções viáveis no que se refere a sequência. Quando as datas de entrega mudam, a natureza das relações de precedência, no mesmo recurso, pode ser alterada. Baseado no conjunto de restrições tecnológicas entre operações em cada tarefa e um conjunto de datas de entrega, a data de início mais cedo e a data de término mais tarde definem uma janela de tempo. As soluções viáveis impõem uma relação de precedência particular entre operações.

2.2.2 Restrições e Regras Difusas

O estudo de restrições e regras difusas será preliminarmente abordado a formulação do problema de scheduling difuso. A idéia dessa seção é apresentar como elementos difusos estão relacionados em problemas de scheduling.

SLANY [23] define restrições sob o escopo de problemas de satisfação de restrições, como segue:

"Constraints are mathematical objects used to make explicit the logic behind a problem. They are used to model decision making problems of, e.g., designing, planning, or scheduling. Classical constraint satisfaction problems are usually composed of 'crisp' constraints, sometimes called 'Boolean', 'yes-no', or hard constraints, i.e., relations that can be either satisfied or not, without intermediate state. A solution must satisfy all constraints of the problem. (...) If a problem has no solutions at all, ...some constraints must then be 'relaxed' to find acceptable solutions."

De acordo com SLANY [23] em problemas de otimização difusa multi-critério, restrições do domínio de aplicação são frequentemente vagamente especificadas, e isto, permite que elas possam ser perfeitamente formuladas como restrições difusas. Ele adotou o conceito de restrições crisp (clássicas) e expandiu-o para restrições soft.

Uma k-ésima restrição soft C_{soft} entre um conjunto de variáveis $x_1,\ldots, x_k \in D_1 \times \ldots \times D_k$, onde D_j é o domínio da variável x_j, pode ser formalizado como uma relação R_{soft} com sua função de pertinência μ_{soft}. R_{soft} aloca um valor de pertinência difusa $\mu_{soft}(x_1, \ldots, x_k)$ de [0, 1] para cada k-tuple $(x_1, \ldots, x_k)$. A função μ_{soft} representa o nível de preferência entre diferentes instanciações.

SLANY [23] especifica 'hard barriers' (barreiras duras) que são limites para o relaxamento de restrições soft quando $\mu_{soft}(x_1, \ldots, x_k) = 0$. Isto significa que não existe uma k-tuple $(x_1, \ldots, x_k)$ que satisfaça a relação R_{soft} totalmente, isto é, ela é uma k-tuple proibida. A definição de restrições soft com barreiras duras permite a compensação de restrições parcialmente satisfeitas por outras restrições sendo satisfeitas num grau maior, enquanto restrições violadas não podem ser contrabalançadas pela satisfação de outras restrições.

Restrições soft são principalmente justificadas através de sua habilidade para medir a satisfação de restrições nos seguintes casos:

No caso de problemas sub-restritos, para especificar preferências no domínio válido das variáveis;

No caso de problemas super-restritos, para especificar margens para restrições, onde elas podem ser relaxadas enquanto ainda sustentem resultados aceitáveis;

Como uma definição natural de prioridades entre restrições;

Na propagação de valores incertos tais como tempos de processamento desconhecidos, representados como distribuições de possibilidades, onde os valores são ordenados de acordo com seu nível de plausibilidade, e;

Em ambos os casos, problemas sub-restritos e super-restritos, para calcular numa maneira natural o valor de uma função objetivo que considera todas as restrições relevantes ao problema [2].

DUBOIS et al [2] sugerem que conjuntos difusos podem ser um modelo natural para restrições soft e uma solução ótima para um problema restrito difusamente. Ela é uma solução tal que o valor de sua pertinência para a interseção dos conjuntos difusos modelando as restrições é máximo. Isto vem ao encontro da maximização do grau de satisfação da restrição menos satisfeita.

DUBOIS et al [2], consideram que toda restrição T_{ij} relacionando um par de variáveis (x_i, x_j) é definida por um conjunto de intervalos $\{I_{ij}^1,\ldots, I_{ij}^n\}$ significando que $x_j - x_i$ deve pertencer a $I_{ij}^1 \,Y \ldots Y\, I_{ij}^n$. Similarmente, o domínio de cada variável x_i é uma restrição unária T_{ii} definida por um conjunto de intervalos. Então, um problema de scheduling pode ser definido como um problema de satisfação de restrições temporais, como segue:

- Restrições de data de liberação e data de entrega s_i para s_0: $s_i - s_0 \in \left[r_i, d_i - t_i\right]$

- Relações de precedência $P_{i\to k}$: $s_k - s_i \in \left[t_i, +\infty\right)$

- Restrições de capacidade $C_{i\to k}$: $s_k - s_i \in \left[t_i, +\infty\right) Y \left(-\infty, -t_k\right]$

Onde, s_0 é uma variável artificial que padroniza o começo do schedule.

Um problema de scheduling difuso é em fato um problema de otimização de restrições para o qual as melhores soluções são aquelas que requerem a menor relaxação de datas de liberação e datas de entrega[2].

Desconsiderada a restrição disjuntiva (restrição de capacidade), cada restrição de precedência $P_{i\to k}$ implica que a janela temporal associada com O_k (respectivamente O_i) deve ser tal que $r_k \geq r_i + t_i$ (respectivamente $d_i \leq d_k - t_k$). Logo, a janela temporal pode ser calculada como segue:

$$r_k := \max\{r_i + t_i \text{ para todo i tal que } P_{i\to k}\} \tag{2-15}$$

$$d_i := \min\{d_k - t_k \text{ para todo k tal que } P_{i\to k}\} \tag{2-16}$$

A dificuldade no problema de job shop scheduling recai na necessidade de "quebrar" as restrições disjuntivas para se obter um problema puramente conjuntivo que é facilmente solucionado via equações 2-15 e 2-16. Isto significa mudar todas as restrições $C_{i\to k}$ em restrições de precedência $P_{k\to i}$ ou $P_{i\to k}$. Isto vem ao encontro de se encontrar uma sequência de operações a serem desempenhadas em cada máquina. Esta sequência deve ser viável no sentido que o problema conjuntivo associado deva ter uma solução em termos dos tempos de início das tarefas. Para conseguir alguma eficiência deve-se usar um procedimento de olhar adiante (look-ahead procedure) que rapidamente cheque algumas consequências de selecionar uma nova restrição de precedência. Tais procedimentos dedicados tem sido proposto na estrutura da abordagem de análise de restrições para o job shop scheduling. Para cada disjunção $C_{i\to k}$ também chamada um conflito (O_i precede O_k ou O_k precede O_i) um teste é realizado como segue:

Test-1 (i, k)

$$\text{Se } d_k - r_i < (t_i + t_k) \text{ então } O_i \text{ não pode preceder } O_k \tag{2-17}$$

Claramente este teste pode levar a uma restrição de precedência que, se violada, leva a uma solução que não é viável (garantindo que O_i deve preceder O_k quando $d_k - r_i < (t_i + t_k)$ impede operações de serem desempenhadas na janela de

tempo [r_i, d_k]). Se ambas condições do test-1 (i, k) e test-1 (k, i) são verificadas, existe uma contradição e o problema de scheduling por si próprio não é viável. Se nenhum destes testes é positivo, então os conflitos permanecem sem solução (desde que nada pode ser inferido sobre a precedência entre O_i e O_k neste caso). Este tipo de teste proporciona apenas uma condição necessária de viabilidade.

Um outro teste proposto por DUBOIS et al [2], envolve mais que duas operações e não apresenta este inconveniente. Por exemplo, considere três operações O_i, O_k, e O_x, e o seguinte:

Test-2 (i, k)

$$\begin{aligned}&\text{Se } d_k - r_i < (t_i + t_k) \text{ ou } \max(d_k - r_i, d_k - r_x, d_x - r_i) < t_i + t_k + t_x \\ &\text{então } O_i \text{ não pode preceder } O_K\end{aligned} \tag{2-18}$$

Onde os três termos na operação de máximo pertencem as respectivas sequências: $O_i/O_x/O_k$, $O_x/O_i/O_k$ e $O_i/O_k/O_x$.

Modelos de scheduling também devem encarar a possibilidade de conflitos – tarefas programadas simultaneamente no mesmo processador. De acordo com RARDIN [22], estes conflitos podem ser modelados pela introdução de novas variáveis de decisão discretas, como segue:

$$y_{j,j',k} \equiv \begin{cases} 1 \text{ se j está programado antes de j' no processador k} \\ 0 \text{ do contrário} \end{cases} \tag{2-19}$$

Modelos de job shop podem impedir conflitos entre tarefas pela introdução de novas variáveis disjuntivas $y_{j,\,j',\,k}$ e o par de restrições:

$$\begin{aligned} x_{j,k} + p_{j,k} &\le x_{j',k} + M(1 - y_{j,j',k}) \\ x_{j',k} + p_{j',k} &\le x_{j,k} + M y_{j,j',k} \end{aligned} \tag{2-20}$$

Para cada j, j' ambos requerendo algum processador k. Aqui $x_{j,\,k}$ denota o tempo de início da tarefa j no processador k, $p_{j,\,k}$ seu tempo de processamento, M é uma

grande constante positiva, e o binário $y_{j, j', k} = 1$ quando j é programado antes que j' em k e $y_{j, j', k} = 0$ se j' é primeiro.

2.3 Branch and Bound

2.3.1 Conceitos

O princípio do Branch and Bound (BB) é a enumeração de todas as soluções viáveis de um problema de otimização combinatorial, diga-se um problema de minimização, tal que propriedades ou atributos não compartilhados por qualquer solução ótima são detectados tão cedo quanto possível. Um atributo (ou ramo da árvore de enumeração) define um subconjunto do conjunto de todas as soluções viáveis do problema original onde cada elemento do subconjunto satisfaz este atributo [3].

O método Branch and Bound é um algoritmo que busca por uma solução ótima através do exame de somente uma pequena parte do número total de possíveis soluções. Ele trabalha quebrando o espaço de soluções viáveis em subproblemas menores até que uma solução ótima seja alcançada. Para cada subproblema gerado o custo total ou lucro é calculado. Subproblemas com pior custo ou lucro são descartados até que não se possa criar mais subproblemas [24].

O termo Branch and Bound refere-se a todos os métodos de busca no espaço de estados na qual todas as crianças do E-node (nó expandido) são gerados antes que qualquer outro nó vivo possa se tornar o E-node. A técnica Branch and Bound trabalha basicamente usando uma das duas estratégias, busca em largura (BFS) ou busca em profundidade (D-search). Na terminologia BB, uma busca BFS será chamada busca FIFO (primeiro que entra primeiro que sai) porque a lista de nós vivos é uma fila. Uma busca D-search será chamada busca LIFO(último que entra primeiro que sai) porque a lista de nós vivos é uma pilha. Como no caso de backtracking, funções de limitação (bounding functions) são usadas para ajudar a evitar a geração de sub-árvores que não contém um nó resposta [25].

BLAZEWICZ et al [3] resumiram o significado do Branch and Bound. Branching (ou ramificação) é o procedimento de particionar um problema grande em dois ou mais subproblemas, geralmente, mutuamente excludentes. Bounding (ou poda) calcula um limite inferior para o valor da solução ótima para cada subproblema gerado no processo de branching. Um nó pode ser eliminado não somente baseado no limite inferior, mas também por meio do chamado critério de eliminação proporcionado por propriedades de dominância ou condições de viabilidade desenvolvidas para um dado problema.

A escolha de um nó a partir do conjunto de nós gerados, que tenham até então, nem sido eliminados, nem levaram a ramificação é resultado da estratégia de busca adotada. Duas estratégias são adotadas frequentemente: jumptracking e backtracking. Jumptracking implementa a busca em largura, onde um nó com o mínimo limite inferior é selecionado para examinação. Backtracking implementa a busca em profundidade, onde os nós descendentes de um nó pai são examinados em uma ordem arbitrária ou em ordem de limites inferiores não-decrescentes. Na estratégia de jumptracking o processo de ramificação salta de um ramo para outro na arvore de busca. Na estratégia backtracking ele primeiro prossegue até o nível mais baixo por algum caminho para encontrar uma solução tentativa e então refaz aquele caminho para cima até o primeiro nível com nós ativos e assim por diante. É fácil notar que jumptracking tende a construir uma grande lista de nós ativos, enquanto backtracking mantém relativamente uns poucos nós na lista a qualquer momento. Embora, uma vantagem do jumptracking é a qualidade de suas soluções tentativas, que são geralmente muito mais próximas do ótimo do que soluções geradas por backtracking, especialmente nos estágios iniciais da busca.

Estas duas estratégias, jumptracking, e backtracking são conhecidas como BFS e D-search respectivamente. Para construir-se um algoritmo de barnch and bound para um dado problema, deve-se decidir sobre:

i. O procedimento de ramificação e a estratégia de busca;

ii. O procedimento de limitação ou critério de eliminação.

Este trabalho considerará somente a estratégia de backtracking. Isto é devido ao fato que é melhor ter-se uma completa quase-ótima solução que uma solução ótima incompleta se o tempo de computação para um dado problema crescer demasiadamente.

Algoritmos de backtracking determinam soluções pela busca sistemática no espaço de soluções para uma dada instância do problema. Usar uma organização em árvore para o espaço de soluções facilita a busca. Cada nó nesta árvore define um estado do problema. Todos os caminhos da raiz para outros nós definem o espaço de estados do problema. Estados soluções (s) são aqueles estados do problema para o qual o caminho da raiz até s define uma tupla que é um membro do conjunto de soluções (ele satisfaz as restrições implícitas) do problema. A organização em árvore do espaço de soluções é referida como árvore do espaço de estados.

Em muitas aplicações do método backtrack, a solução desejada é expressa como uma n-tuple $(x_1,\ldots, x_n)$, onde x_i são escolhidos do conjunto finito S_i. Frequentemente, o problema a ser resolvido determina que se encontre um vetor que maximize (ou minimize, ou satisfaça) uma função critério $P(x_1,\ldots, x_n)$. Algumas vezes ele procura por todos os vetores que satisfazem P.

Muitos problemas que são resolvidos usando-se backtracking requerem que todas as soluções satisfaçam um conjunto complexo de restrições. Para qualquer problema estas restrições podem ser divididas em duas categorias: explícitas e implícitas.

Restrições explícitas são regras que restringem cada x_i a somente tomarem valores de um dado conjunto. Elas dependem da instância particular I do problema sendo resolvido. Todas as tuplas que satisfazem as restrições explícitas definem um possível espaço de soluções para I.

As restrições implícitas são regras que determinam quais das tuplas no espaço de soluções de I satisfazem a função critério. Logo, restrições implícitas descrevem a maneira pela qual os x_i devem relacionarem-se uns aos outros.

A idéia básica do algoritmo de backtracking é construir o vetor solução inserindo um componente de cada vez e usar funções critério modificadas $P_i(x_1,\ldots, x_i)$ (algumas vezes chamadas funções de poda) para testar se o vetor sendo formado tem alguma chance de sucesso. A maior vantagem deste método é que se percebido que o vetor parcial $(x_1, x_2,\ldots,x_i)$ não pode de forma alguma levar a uma solução ótima, então, $m_{i+1}\ldots, m_n$ possíveis vetores podem ser ignorados inteiramente.

2.3.2 Como BB funciona

HOROWITZ et al [25] considera que a formulação do processo de backtracking pode ser apresentada como segue. Deixe $(x_1, x_2,\ldots,x_i)$ ser um caminho da raiz até um nó na árvore de espaço de estados. Deixe $T(x_1, x_2,\ldots,x_i)$ ser o conjunto de todos os possíveis valores de x_{i+1} tal que $(x_1, x_2,\ldots,x_{i+1})$ é também um caminho para um estado do problema. $T(x_1, x_2,\ldots,x_n) = \varnothing$. É assumida a existência de uma função de poda B_{i+1} (expressa como predicados) tal que se $B_{i+1}(x_1, x_2,\ldots,x_{i+1})$ do nó raiz ao estado do problema, então o caminho não pode ser estendido para alcançar-se um nó resposta. Logo, os candidatos para a posição i+1 do vetor solução $(x_1,\ldots,x_n)$ são aqueles valores que são gerados por T e satisfazem B_{i+1}.

RENDER et al [24] apresenta seis passos simples para resolver problemas de maximização em programação inteira através de branch and bound, que foram modificados e apresentados aqui para problemas de minimização:

1. Resolver o problema original usando programação linear. Se a resposta satisfaz a restrição inteira, esta é a solução, pare. Se não, estes valores proporcionam um limite inferior inicial.

2. Encontrar qualquer solução viável que preenche a restrição inteira para uso como um limite superior. Usualmente, arredondar cada valor de variável realizará isso.

3. Ramificar a variável do passo 1 que não tenha um valor inteiro. Dividir o problema em dois subproblemas baseados nos valores inteiros que estão imediatamente abaixo ou acima do valor não inteiro.

4. Criar nós no topo desses novos ramos pela solução dos novos problemas.

5. A) Se um ramo leva a uma solução inviável por programação linear, descarte o nó;

   B) Se um ramo leva a uma solução viável por programação linear, mas não uma solução inteira vá para o passo 6;

   C) Se o ramo leva a uma solução inteira viável, examine o valor da função objetivo. Se este valor é igual ao limite inferior, uma solução ótima foi alcançada. Se ele não é igual ao limite inferior, mas ele é menor que o limite superior, adote-o como um novo limite superior e vá para o passo 6. Finalmente, se ele é maior que o limite superior, descarte esse ramo.

6. Examine ambos os ramos novamente e adote como limite superior o valor máximo da função objetivo para todos os nós finais. Se o limite inferior é igual ao limite superior, pare. Se não, volte ao passo 3.

2.3.3 Vantagens e Desvantagens

AVRIEL [26] afirma que técnicas Branch and Bound (B&B) podem ser vista de diferentes formas como por exemplo, enumeração implícita, avaliação e separação progressivas na árvore de busca, particionamento estratégico e corte provisional. B&B tem duas qualidades principais que diferenciam estas técnicas de outras. Primeiro, elas podem ser aplicadas a problemas de programação inteira mista, essencialmente da mesma maneira que elas podem ser aplicadas a problemas de programação inteira pura. Segundo, elas tipicamente encaminham uma sucessão de soluções inteiras viáveis, dessa forma, garantindo a melhor solução à mão como uma candidata a ótima, mesmo quando o tempo computacional for muito grande.

Métodos B&B podem ser customizados para explorar determinadas estruturas do problema, permitindo que estas estruturas sejam manuseadas com maior eficiência e reduzida alocação de memória. De fato, exceto em termos muito gerais ,

não existe um método B&B, mas umas coleção deles que dividem um número de características comuns.

Considerando-se especificamente a estratégia de backtracking em B&B, HOROWITZ et al [25] afirmam que dois tipos de algorithmos são geralmente encontrados. Backtracking pode ser, recursivo ou geral . As vantagens de cada um devem ser analisadas considerando-se os quatro fatores descritos a seguir:

1. O tempo para gerar o próximo nó;

2. O número de nós que satisfazem as restrições explícitas;

3. O tempo para as funções de poda;

4. O número de nós que satisfazem as funções de poda.

A importância do backtracking reside na sua habilidade para resolver algumas instâncias de problemas num pequeno espaço de tempo. A única dificuldade está em predizer o comportamento de um algoritmo backtracking para a instância do problema sendo resolvida [25].

2.3.4 Aplicações em Problemas de Scheduling

CONWAY et al [21] descrevem um algoritmo B&B para resolver o problema do caixeiro viajante. Esta descrição é baseada no trabalho original de Little, Murty, and Sweeny que tinham introduzido a técnica B&B em 1963. Além deste uso para B&B, a minimização do flowtime médio e máximo foram também alcançados por B&B em casos mais gerais que os investigados por Johnson [27].

Considerando-se problemas de flow shop e job shop, CONWAY et al [21] analisaram o poder de B&B sob o ponto de vista da qualidade dos limites inferiores. Eles afirmaram que este poder é fortemente dependente da qualidade dos limites inferiores nos estágios iniciais da ramificação. Eles também sugeriram que as possibilidades mais promissoras para se alcançar estes limites inferiores poderiam ser realizadas por regras de despacho como por exemplo. A data de início mais cedo mais a

soma de todos os tempos de processamento de tarefas ainda não programadas tomadas em seu valor máximo ou, a data de início mais cedo mais a soma dos tempos de processamento das operações não programadas em cada máquina. Infelizmente, estes procedimentos para se determinar os limites inferiores são úteis quando se está trabalhando na minimização do máximo flowtime e nada pode ser afirmado sobre lateness considerando-se estes procedimentos.

Recentemente, B&B parece ter sido rediscoberta e aplicações em problemas de scheduling estão se tornando mais comuns. Exemplos de aplicações de B&B considerando-se diferentes objetivos podem ser encontrados em : STANFIELD et al [28] que discute uma aplicação com tempos de processamento e datas de entregas difusas; McMAHON et al [29], GIM et al [30], e PATTERSON et al [31] que aplicaram-na considerando as relações de precedência; CHENG et al [32], BRAH et al [33], NAGAR et al [34], e CARLIER et al [35 e 36], que resolveram uma variedade de problemas usando B&B.

2.4 Conclusões Parciais

Algumas conclusões parciais podem ser inferidas a este ponto. Elas ajudarão na realização dos objetivos apresentados na introdução do capítulo.

Pode ser concluído que as constantes u e v usadas nas equações 2-1 e 2-2 expressam as preferências do tomador de decisões numa maneira que a forma das funções de pertinência representam diferentes comportamentos dos clientes com relação a datas de entrega. Aqui, pode-se considerar clientes internos e externos a organização. A determinação destes parâmetros pode ser objeto de um outro estudo.

Quando durações difusas representam conhecimento impreciso sobre durações não-controláveis de uma tarefa, o grau de satisfação de datas de entrega e restrições de capacidade e precedência são graus de necessidade que são motivados pela busca de viabilidade e robustez nos programas de produção em face a eventos de risco. Isto está em total contraste com o caso em que t_i é uma variável de decisão controlável: os graus de satisfação de restrições de datas de entrega, relações de precedência e capacidade são graus de possibilidade porque é sempre possível escolher os melhores

valores para as durações. É então suficiente que existam durações satisfazendo as restrições da melhor maneira possível, desde que, a viabilidade seja expressa somente em termos de possibilidades. Em outras palavras existe uma diferença conceitual entre possibilidade e grau de satisfação. Uma refere-se ao valor que um parâmetro pode assumir. A outra representa quanto o valor de um parâmetro é satisfatório considerando-se um dado objetivo.

Finalmente, é possível concluir que a abordagem difusa modela melhor os problemas de scheduling porque ela permite relaxar e considerar restrições em graus diferentes. Por exemplo, um tomador de decisão pode decidir qual cliente deve ter a data de entrega mais ou menos respeitada e ele pode fazer isso com diferentes importâncias para cada um.

O estudo da programação da produção usando lógica difusa é um grande campo a ser explorado e pode ajudar a modelar e resolver problemas de programação de uma maneira mais próxima a como os especialistas humanos fazem isso. Este campo não pode ser coberto em apenas uma tese e, portanto, seria interessante que existisse mais pesquisa realizada nesta área.

3 Definição do Problema

3.1 Introdução

Este capítulo objetiva apresentar a definição do problema através do estabelecimento de seus conceitos básicos, o domínio do problema e seu campo de aplicação, gerando uma declaração formal do problema.

Desenvolver uma boa programação preditiva que satisfaça restrições temporais, tecnológicas e de outros tipos é basicamente um problema de busca. A solução requer poderosas heurísticas de busca e adequados meios de representação. Neste trabalho, restrições temporais flexíveis sobre tempos de processamento e datas de entrega, como também, tempos de processamento incertos são expressos e manuseados na estrutura da teoria da possibilidade desenvolvida por DUBOIS et al [2].

A técnica primária para problemas de programação é determinar a sequência na qual as tarefas serão processadas num conjunto de máquinas. A necessidade por programas de produção se deve a limitação de recursos disponíveis ao tomador de decisão [34].

Entre as aplicações neste campo, problemas de tomada de decisão oferecem um vasto campo de questões que podem beneficiar-se grandemente das metodologias em conjuntos difusos e teoria da possibilidade, tais como, modelagem de preferências, decisão sob incerteza, ou decisões baseadas em casos, individualmente ou coletivamente. Aqui, representações que são baseadas em conjuntos difusos e distribuições de possibilidades são consideradas através de conjunto de valoração finito linearmente ordenados. Mesmo se esta estrutura é claramente mais pobre que o intervalo [0, 1], o potencial desta estrutura simples é ainda de investigação válida para análise de decisão, desde que ela é menos exigente sobre a qualidade dos dados que são necessários para alimentá-la [37].

3.2 Domínio do Problema

O problema atacado neste trabalho é um problema de flow shop scheduling na qual a função objetivo a ser maximizada é o grau de satisfação da data de término em relação a data de entrega. A complexidade deste problema é aumentada pela incerteza relativa aos tempos de processamento e a consideração de datas de entrega como restrições flexíveis. Ambos, tempos de processamento e datas de entrega são vistos como distribuições de possibilidades representadas como números difusos trapezoidais. Mais especificamente, o objetivo é encontrar a sequência que maximiza o grau de satisfação pela minimização dos custos de adiantamento total e atraso total de todas as tarefas. O ambiente de programação está sujeito as seguintes restrições: não é permitida preempção, não é permitida a inserção de tempo ocioso e, todas as tarefas estão inicialmente disponíveis.

Existem dois entendimentos distintos de uma distribuição de possibilidade π, que é um mapeamento do universo U a um limitado conjunto de valoração linearmente ordenado S, tal que $\pi(u) = 0$ significando que u é uma situação impossível. Primeiro, uma distribuição de possibilidade pode codificar uma parte de conhecimento impreciso sobre uma situação, como em raciocínio aproximado. A outra visão é em termos de preferências e leva ao cálculo de restrições flexíveis baseadas no cálculo de relações difusas [37].

Portanto, o critério de minimização é o custo total de atrasos e adiantamentos, que podem ser considerados medidas regulares de desempenho. Todas as tarefas são comparadas com suas respectivas datas de entrega. Se a tarefa está atrasada ou adiantada é associado um custo a ela. A determinação de tarefas atrasadas ou adiantadas é uma comparação entre dois números difusos trapezoidais. O operador de comparação, assim como outros operadores usados, serão discutidos no próximo capítulo.

Com o aumento de popularidade da filosofia Just-In-Time (JIT) na última década, tem havido um esforço pelos gerentes de operações para reduzir todas as formas de desperdício dos processos de produção . Estoque de material em processo (WIP) tem sido identificado como uma das fontes primárias de desperdício pela filosofia JIT [34].

Portanto, o objetivo de minimizar-se os atrasos e adiantamentos é também compatível com a filosofia JIT que visa minimizar os custos de estoque pela produção de bens tão próximos quanto possível às suas datas de entrega [38].

Considerando-se a premissa que a tarefa deve ser completada tão perto quanto possível de sua data de entrega, o problema pode ser entendido como a minimização da seguinte função objetivo com penalidades para atrasos e adiantamentos. Esta função objetivo (equação 3-1) é uma extensão da equação de penalidades por atrasos e adiantamentos apresentada por ISHIBUCHI et al [15].

$$Minimize\, f = \sum_{j=1}^{n}\left[\alpha_j E_j \mu\left(E_j\right) + \beta_j T_j \mu\left(T_j\right)\right] \tag{3-1}$$

Onde α_j e β_j são o custo por uma unidade de adiantamento e o custo por uma unidade de atraso, respectivamente.

De fato, sete cenários diferentes são possíveis quando se considera tempos de término e datas de entrega difusas. Assumindo estes dois números difusos como intervalos ou "janelas temporais" quando $C_j\mu(C_j) \cap D_j\mu(D_j) = Ø$ a sequência é inviável. As figuras 3-1 e 3-2 mostram estes dois cenários.

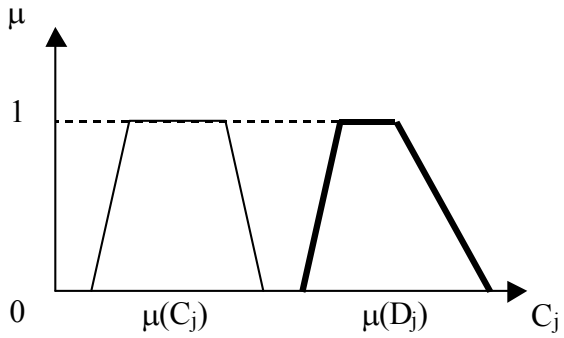

Figura 3-1. $C_j\mu(C_j) \cap D_j\mu(D_j) = Ø$ com tempo de término menor que a data de entrega

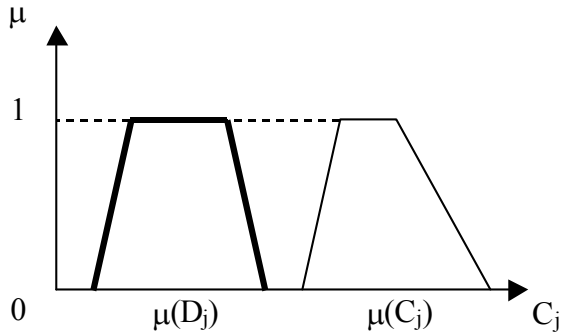

Figura 3-2. $C_j\mu(C_j) \cap D_j\mu(D_j) = Ø$ com tempo de término maior que a data de entrega

Considerando-se estes dois cenários que representam $C_j\mu(C_j) \cap D_j\mu(D_j) = Ø$ é evidente que sequências com estes tempos de término podem ser eliminados da árvore de busca porque o grau de satisfação para a função objetivo é zero. É importante entender que a restrição soft data de entrega só pode aceitar valores dentro do alcance especificado através de sua função de pertinência.

Os outros cinco cenários possíveis, quando $C_j\mu(C_j) \cap D_j\mu(D_j) \neq Ø$, geram soluções viáveis. Figuras 3-3 à 3-5 e equações 3-2 à 3 –4 mostram as funções de pertinência para três cenários exemplos. Por questões de simplicidade, as equações e figuras serão mostradas usando somente adiantamentos. Para atrasos a extensão é obvia.

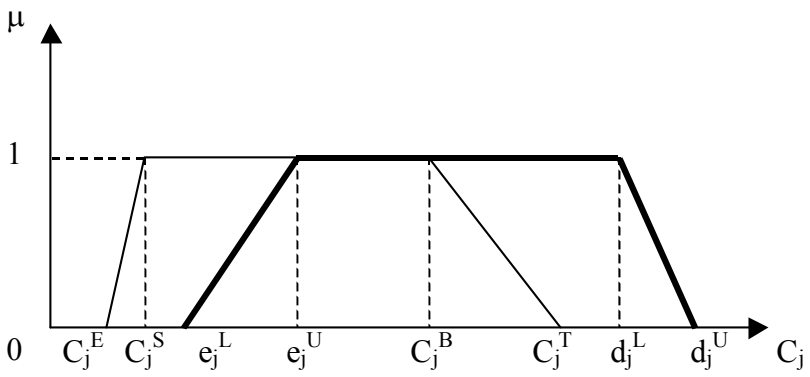

Figura 3-3. Cenário 3

$$\mu_j\left(E_j\right)=\begin{cases}0 & \text{if } C_j < e_j^L \\ 1-\left(e_j^u-C_j\right)\Big/\left(e_j^u-e_j^L\right) & \text{if } e_j^L \le C_j < e_j^U \\ 1 & \text{if } e_j^u \le C_j < C_j^B \\ 1-\left(C_j^T-C_j\right)\Big/\left(C_j^T-C_j^B\right) & \text{if } C_j^B \le C_j < C_j^T \\ 0 & \text{if } \mathrm{C_j} \ge C_j^T \end{cases} \tag{3-2}$$

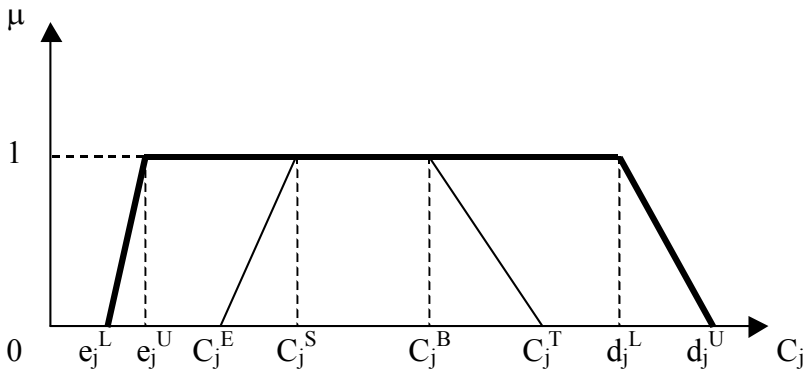

Figura 3-4. Cenário 4

$$\mu\left(E_j\right)=\begin{cases}0 & \text{if } C_j < C_j^E \\ 1-\left(C_j^S-C_j\right)\Big/\left(C_j^S-C_j^E\right) & \text{if } C_j^E \le C_j < C_j^S \\ 1 & \text{if } C_j^S \le C_j < C_j^B \\ 1-\left(C_j^T-C_j\right)\Big/\left(C_j^T-C_j^B\right) & \text{if } C_j^B \le C_j < C_j^T \\ 0 & \text{if } C_j \ge C_j^T \end{cases} \tag{3-3}$$

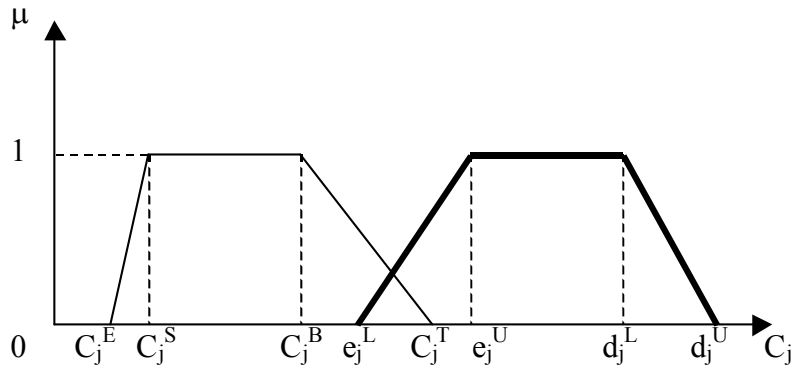

Figura 3-5. Cenário 5

$$\mu\left(E_j\right)=\begin{cases}0 & \text{if } C_j < e_j^L \\ 1-\left(C_j^T-C_j\right)\Big/\left(C_j^T-e_j^L\right) & \text{if } e_j^{\mathrm{L}} \le C_j < C_j^T \\ 0 & \text{if } C_j \ge C_j^T\end{cases} \tag{3-4}$$

Cada cenário apresentado acima é entendido como uma possível localização do tempo de término (linha mais fina) da j-ésima tarefa relativa a sua data de entrega (linha mais grossa)

No caso da figura 3-3 e equação 3-2 a função de pertinência $\mu_j(E_j)$ indica que a tarefa j é terminada "about before" sua data de entrega. Existe uma situação opsta, não apresentada aqui, onde a tarefa j é terminada "about after" sua data de entrega.

Figura 3-4 e equação 3-3 mostram a situação onde a incerteza da data de entrega é maior em comparação com o tempo de término. Similarmente, figura 3-5 e equação 3-4 mostra que a tarefa j é completada "almost certainly before" sua data de entrega. Aqui novamente, existe um caso oposto onde a tarefa j é completada "almost certainly after" sua data de entrega.

Esta formulação difusa para o problema de scheduling pretende guiar a busca por melhores sequências no domínio de soluções viáveis no problema de minimização de penalidades por atrasos e adiantamentos.

3.3 Campos de Aplicação para este Problema

De acordo com CHENG et al [32], este problema clássico de scheduling ainda é importante em sistemas de manufatura flexível que objetivam uma completa automação do chão de fábrica com vários itens e pequenos lotes de produção.

Este problema tem sua principal aplicação quando o custo para produzir-se um item é fortemente dependente de quando ele estará pronto. O primeiro exemplo foi mencionado no capítulo um. Um segundo exemplo refere-se a itens que tem uma data de validade. Se eles são produzidos muito cedo existe um custo de deterioração quando

eles estão no estoque. Se eles são produzidos muito tarde o risco é que o cliente não aceite mais a encomenda.

Para todos os tipos de sistemas de produção existe um custo para ser manter um item em estoque e este custo deve ser minimizado. Na verdade, as medidas regulares de desempenho para problemas de programação da produção como makespan, flowtime e flowtime médio tem seu principal uso quando a demanda para um dado item é maior que a oferta para aquele item. Do contrário é importante ter –se os itens prontos para a entrega no prazo, nem antes e nem depois das datas prometidas aos clientes.

Em casos reais é importante ter-se alguma flexibilidade que é traduzida para modelo através da restrição soft data de entrega difusa ainda que custos sejam incorridos. Portanto, casos reais também deveriam ser modelados incorporando-se as variabilidades do processo e conhecimento incompleto sobre tempos de processamento, já que eles podem ser muito instáveis.

4 O Modelo Proposto

4.1 Descrição do Modelo

O modelo proposto neste capítulo busca a melhor solução para o problema apresentado no capítulo 3 usando uma técnica Branch and Bound (B&B) que avalia todas as soluções viáveis para todo espaço de estados do problema [45].

Como mencionado anteriormente, o problema a ser resolvido é do tipo flow shop. Este problema tem sido estudado nos últimos cinquenta anos sempre considerando-se que todos os tempos são exatos e facilmente obtidos. Entretanto, essa consideração nem sempre é verdade. Tempos de processamento geralmente são obtidos de observações por amostragem das tarefas sendo produzidas, usando-se algum procedimento de tempo padrão. Estas tarefas estão sujeitas a vários fatores que não estão sob controle dos tomadores de decisão. Trabalhadores diferentes tem diferentes ritmos, postos de trabalho similares tem diferentes taxas de trabalho, etc.

Por exemplo, suponha um problema de alocação de salas de emergências em grandes hospitais. Médicos e Administradores tem alguma idéia quanto tempo leva um certo procedimento, mas este tempo pode mudar de acordo com a resposta do paciente, tipo de tratamento sendo realizado e, logo, toda uma programação pode ser influenciada.

Um outro caso pode ser visto em aeroportos. Se uma aeronave chegando para aterrizagem tem uma abordagem incorreta, uma nova abordagem é necessária e algum tempo deve ser acrescido ao processo de aterrizagem.

Em ambientes de manufatura muitos fatores podem agir sobre os tempos de processamento. Máquinas podem falhar, trabalhadores podem não estar disponíveis, dispositivos, ferramentas e outros recursos podem estar for a do alcance das áreas que as necessitam num dado momento.

Para casos como estes apresentados acima, a abordagem difusa pode ser muito mais realista que a abordagem crisp a este tipo de problema. Sendo assim este modelo pretende modelar problemas de scheduling de forma mais fiel.

Num típico ambiente de manufatura, qualquer sequência (schedule), uma vez determinado está quase que imediatamente sujeito a novas condições, demanda e/ou restrições. Isto torna quase mandatória a reprogramação. Todo o movimento em direção a 'resposta rápida', 'sistema flexível de manufatura', etc, claramente indicam a necessidade de um método de planejamento da produção que suporte uma eficiente reprogramação [42].

De cordo com LI et al [42], tipicamente descreve as operações (ou tarefas) requeridas para processar-se um produto, máquinas na qual as operações serão desempenhadas e, tempos de início e de fim de cada operação. Por causa da complexidade dos modernos sistemas de manufatura e do fato que eles operam num ambiente estocástico e dinâmico, a reprogramação é frequentemente requerida.

Uma grande variedade de condições, demanda e restrições necessitam revisões num programa existente. Alguns desses fatores de reprogramação são:

- Quebras de máquinas;
- Interrupções devido a chegada de encomendas 'quentes';
- Falta de material;
- Problemas de qualidade;
- Super ou sub estimativas do tempo de processamento;
- Cancelamento de uma encomenda;
- Mudanças de datas de entrega;
- Estar atrás ou na frente do programa corrente.

Algumas vezes é necessário gerar o programa novamente como uma reação as estratégias seguidas em resposta a eventos causando uma falha do programa corrente. Entre estes fatores estão:

- Overtime;
- Subcontratações no processo;
- Mudança no processo ou re-roteamento;
- Substituição de máquinas.

Embora possa existir algumas significativas diferenças entre estas duas listas de fatores acima, ambas chamam por uma ação de reprogramação [42].

WARD et al [43], afirmam que muitos trabalhadores concordam que os existentes modelos de programação da produção não estão sendo implantados na prática. Varias explicações tem sido apresentadas:

- É difícil forçar os administradores a entenderem os relacionamentos em suas plantas de fabricação com o rigor matemático requerido pelas técnicas de otimização;
- Tomadores de decisão desconfiam da qualidade dos modelos;
- Programadores de produção reclamam que seus parâmetros de programação são não somente únicos mas suficientemente diferentes de quaisquer outros parâmetros requeridos para a solução de um problema específico;
- Todos dizem que as técnicas de solução não representam adequadamente as realidades de suas operações.

Um dos problemas mais desafiadores da Inteligência Artificial é lidar eficientemente com as explosões combinatoriais. Aplicações práticas de Inteligência

Artificial devem atacar os problemas de explosão combinatorial em todas as áreas do conhecimento, incluindo aquelas como, sistemas baseados em conhecimento, robótica, scheduling e reconhecimento de padrões.

Produtos customizados e por encomenda tem que ser feitos com respeito à específica janela de tempo, pela eficiente utilização dos recursos de manufatura. A maioria das tradicionais abordagens de programação da produção são baseadas na consideração que para cada peça existe somente um plano de processo disponível [44].

Num problema de satisfação de restrições quer-se determinar um programa de produção que satisfaça restrições de recursos e da janela de tempo.

DUBOIS et al [14] apresentaram várias facetas da metodologia de conjuntos difusos que oferecem ferramentas para a modelagem de satisfação parcial de restrições ou critérios e para representação de informação incompleta.

Portanto, conclui-se através destas afirmações acima apresentadas que a teoria dos conjuntos difusos é uma alternativa interessante de uma estrutura para lidar com o planejamento e a programação da produção. Suas principais vantagens são:

- Ela leva a uma redução do esforço da engenharia do conhecimento e permite a representação e inferência do conhecimento como em humanos, desde que muito do conhecimento a ser introduzido no sistema é proporcionado por especialistas humanos, na forma de regras difusas que são muito próximas a linguagem natural;

- Ela simplifica o desenvolvimento, manutenção e experimentação, desde que é muito mais fácil mudar regras que alterar programas;

- Ela permite a representação de conceitos vagos, isto é, ela incorpora e pondera as incertezas do sistema.

Resumindo, a abordagem difusa permite a inclusão de novas maneiras de representação da experiência do especialista, a inferência de novos conhecimentos de decisão e controle e o melhoramento da interação entre operadores humanos e

computadores de controle. Ela é bem sucedida na modelagem de sistemas pobremente estruturados ou nada estruturados [13].

Finalmente, o modelo pode ser entendido através das seguintes equações:

$$r_k := \max\{r_i + t_i \text{ para todo i tal que } \mathrm{P}_{\mathrm{i}\to\mathrm{k}}\} \tag{4-1}$$

$$d_i := \min\{d_k - t_k \text{ para todo k tal que } \mathrm{P}_{\mathrm{i}\to\mathrm{k}}\} \tag{4-2}$$

$$Minimize\, f = \sum_{j=1}^{n}\left[\alpha_j E_j \mu\left(E_j\right) + \beta_j T_j \mu\left(T_j\right)\right] \tag{4-3}$$

A notação usada aqui foi apresentada no capítulo 2.

4.2 Processo de Solução

O processo para obter-se a melhor solução inicia pela entrada dos tempos de processamento de cada tarefa em cada estação de trabalho e a preferida data de entrega para cada tarefa. O sistema computacional que implementa este modelo calcula o tempo de término para cada tarefa considerando que a tarefa j seja programada primeiro. Este processo é realizado expandindo-se o nó raiz em n nós, onde n é o número de tarefas a serem programadas.

Para cada tarefa, o 'lateness' é calculado e a viabilidade daquele ramo é verificada. Nós inviáveis são eliminados e não serão expandidos. Os nós viáveis são mantidos na pilha para futura expansão. Este processo é repetido até que todas as tarefas tenham sido programadas.

Como mencionado no capítulo 2, este modelo implementa uma busca em profundidade na qual a estratégia de ramificação é quebrar o problema em n-1 subproblemas para cada nível na árvore de busca. A estratégia de poda é a comparação entre o tempo de término e a data de entrega para cada tarefa. Na verdade, este é o valor da função objetivo para cada estado de espaços.

É importante lembrar que se $C_j\mu(C_j) \cap D_j\mu(D_j) = Ø$ a referida solução parcial é inviável e aquele nó como também seus nós filhos serão eliminadas. Este mecanismo melhora tremendamente o critério de eliminação e resultados para problemas médios e grandes são rapidamente obtidos.

O modelo pode ser entendido melhor através dos seguintes passos:

1. **Carregar os dados do problema** que consistem em tempos de processamento, datas de entrega e custos para atrasos e adiantamentos;

2. **Inicializar a árvore de busca** pela criação do nó raiz. O nó raiz torna-se o primeiro nó vivo esperando pela ramificação. O nó raiz é sempre viável e não existe um "nó vencedor";

3. **Ramificar o próximo nó vivo disponível**. Este passo cria mais nós que são mantidos até avaliação e ordenação dos custos (passos seguintes);

4. **Calcular o tempo de término e o custo acumulado corrente**. Ordenar os nós recentemente criados que são programados na pilha de nós vivos. O primeiro nó vivo neste ramo será o próximo nó a ser ramificado (a menos que ele seja podado ou considerado inviável);

5. **Se o primeiro nó vivo não é nem viável nem menos custoso** que o "nó vencedor" corrente, pode este nó e avance para o próximo nó vivo. Este passo leva a um ciclo até que o próximo nó viável seja encontrado– ir para o próximo passo, ou não existem mais nós vivos – ir para o passo nove;

6. **Se o próximo nó disponível não é um nó folha**, retorne ao passo três, do contrário, continue;

7. **Aloque o próximo nó vivo disponível para ser o "nó vencedor"**. Este nó é um nó folha (uma sequência completa) e tem um custo mais baixo que o vencedor prévio (como avaliado no passo 5).

8. **Continue avaliando e ramificando** nós vivos pelo retorno ao passo 5.

9. **Se um nó vencedor foi alocado no passo sete**, o problema é viável e uma solução ótima foi encontrada. Do contrário, o problema é considerado inviável.

4.3 Formulação Principal

O modelo proposto tem sua fundação construída sobre formulações para sequenciamento de tarefas, que são expandidos de acordo com as regras de ramificação e poda. Cada sequência tem um tempo de término e custo associados. A cardinalidade da sequência é dada pelo número de tarefas. A sequência na raiz tem cardinalidade igual a zero. Uma sequência completa tem cardinalidade igual ao número de tarefas.

Uma sequência é denotada por S_k, onde *k* é um identificador da sequência. A sequência raiz tem *k* igual a zero: $S_0 = \varnothing$. A cardinalidade da sequência é denotada por $|S_k| = q$, onde *q* é o número de tarefas. A *i-ésima* tarefa da sequência *k* é denotada por $S_k(i)$.

Para calcular-se o tempo de término de uma tarefa é necessário determinar o tempo de saída de cada tarefa em cada máquina.

Deixe *l* ser o tempo de saída de uma dada tarefa *b* em S_k, considerando *j* o número da máquina.

$$l_{b,j} = \max(l_{b,j-1}, l_{a,j})(+)p_{b,j} \tag{4-1}$$

Onde $b = S(i)$ e $a = S(i-1)$ são tarefas indexadas por $1 \le i \le |S_k|$; $(+)$ é o operador difuso de adição; e $p_{b,j}$ é o tempo de processamento da tarefa *b* na máquina *j*.

Por razões de consistência, uma tarefa hipotética 0 é considerada na formulação e, seu tempo de saída em qualquer máquina é zero, como a seguir:

$$L_{0,j} = (0,0,0,0), \quad 1 \le j \le m \tag{4-2}$$

Finalmente, o tempo de término para uma dada tarefa *i* pode ser calculada como:

$$C_i = l_{i,m} \tag{4-3}$$

Onde *m* é a última máquina.

4.4 Operadores Difusos Usados

Este modelo usa três operadores difusos para efetuar os cálculos necessários.

A primeira operação a ser considerada é o método de Lee-Li. Ele também é conhecido pelo nome de valor médio generalizado (GMV) e é citado em McCAHON et al [5]. O GMV para um número difuso trapezoidal com densidade uniforme é calculado como:

$$m(A) = \frac{\int_a^d x\mu_A(x)dx}{\int_a^d \mu_A(x)dx} \tag{4-4}$$

Onde m(A) é o GMV e μ_A é o valor da função de pertinência num dado x.

A equação 4-7 é aplicada a equação 2-2 que representa os números difusos trapezoidais (tempos de processamento e datas de entrega). O número difuso trapezoidal com maior GMV é considerado maior que o número difuso com GMV menor. Se acontece dos números difusos terem o mesmo GMV, a dispersão s(A), é calculada para cada número difuso e aquele com maior dispersão é classificado maior. A dispersão de um número trapezoidal difuso com densidade uniforme é calculada como na equação 4-8.

Ambas equações (4-7 e 4-8) foram desenvolvidas para a formulação apresentada no capítulo 2 gerando 4-9 e 4-10.

$$s(A) = \left(\frac{\int_a^d x^2 \mu_A(x) dx}{\int_a^d \mu_A(x) dx} - [m(A)]^2 \right)^{1/2} \tag{4-5}$$

$$m(A) = \frac{-a^2 - b^2 - ab + c^2 + d^2 + cd}{3(-a - b + c + d)} \tag{4-6}$$

$$s(A) = \left(\frac{-ab^2 - a^2 b - a^3 + c^3 - b^3 + dc^2 + cd^2 + d^3}{6(-a - b + c + d)} - (m(A))^2 \right)^{1/2} \tag{4-7}$$

Nas equações 4-9 e 4-10, a, b, c, e d correspondem aos vértices do trapézio que representa o número difuso, como nas figuras 2-3 e 2-5.

Os demais operadores difusos considerados são adição e subtração, (+) e (-) respectivamente.

Uma propriedade importante dos números trapezoidais difusos, de acordo com KAUFMANN et al [39], é que a adição ou subtração de dois números trapezoidais difusos resulta um número trapezoidal difuso. Representando dois números trapezoidais difusos por quadruplas, tem-se:

$$A = (a_1, a_2, a_3, a_4) \text{ e } B = (b_1, b_2, b_3, b_4) \tag{4-8}$$

Adição:

$$A (+) B = (a_1 + b_1, a_2 + b_2, a_3 + b_3, a_4 + b_4) \tag{4-9}$$

Subtração:

$$A (-) B = (a_1 - b_4, a_2 - b_3, a_3 - b_2, a_4 - b_1) \tag{4-10}$$

Maiores detalhes sobre operações com números difusos podem ser vistos em DUBOIS et al [40] e ZIMMERMANN [41].

5 Experimentos e Resultados

Este capítulo apresenta um experimento composto de três problemas e os resultados obtidos. O primeiro problema usa tempos de processamento difusos obtidos de McCAHON et al [5] e datas de entrega estimadas através da média de tempos de término de todas as possíveis sequências. Para o segundo problema, os tempos de processamento e as datas de entrega difusas foram gerados arbitrariamente. O terceiro problema foi gerado usando-se um gerador de problemas desenvolvido para o "Fuzzy Scheduler Problem Solver" (FSPS). Os resultados dos três problemas são discutidos depois de suas descrições. Para todos eles é assumido um custo de 10 unidades para cada unidade de tempo atrasado e um custo de 1 unidade para cada unidade de tempo em adiantamento.

5.1 Problema 1

Os tempos de processamento de um problema de flow shop com 4 tarefas e 4 estações de trabalho são apresentados na tabela 5-1.

Tabela 5-1. Dados para o Problema 1.

Jobs	W1	W2	W3	W4
1	(4, 5, 6, 7)	(5, 5, 6, 7)	(1, 3, 4, 5)	(2, 3, 5, 6)
2	(2, 3, 4, 6)	(6, 7, 7.5, 8)	(1, 3, 4, 5)	(5, 5.5, 6, 7)
3	(8, 9, 11, 12)	(4, 5, 6, 9)	(3, 5, 6, 7)	(2, 4, 5, 6)
4	(3, 4, 5, 8)	(5, 6, 8.5, 9)	(3, 4, 5, 6)	(1, 2, 3, 4)

Neste problema a data de entrega para todas as tarefas é o número trapezoidal difuso (29, 37, 45, 55). A sequência com o menor custo de lateness é a

sequência vencedora e as tarefas são 3-2-1-4, nesta ordem de processamento. Valores individuais associados com cada tarefa nesta sequência estão resumidos na tabela 5-2.

O custo difuso apresentado na tabela 5-2 é de difícil compreensão, mas é possível calcular seu centróide e dispersão obtendo-se o valor crisp do custo. O custo da sequência vencedora é 45.66 e sua dispersão é 134.27.

Tabela 5-2. Resultados do Problema 1.

Job	Completion Time	Lateness	Cumulative Cost
3	(17, 23, 28, 34)	(-5, 9, 22, 38)	(-5, 9, 22, 38)
2	(24, 29.5, 34.5, 41)	(-12, 2.5, 15.5, 31)	(-17, 11.5, 37.5, 69)
1	(26, 32.5, 39.5, 47)	(-18, -2.5, 12.5, 29)	(-35, 9, 50, 98)
4	(32, 38, 47, 55)	(-23, -7, 10, 26)	(-265, -61, 150, 358)

Outro resultado interessante a ser analisado é o comportamento do Branch and Bound durante a busca. Vide tabela 5-3.

Tabela 5-3. Comportamento do Branch and Bound no Problema 1.

Parameter	Amount	Percentage
Total nodes	65	100
Expanded Nodes	20	30.77
Skipped Nodes	45	69.23
Bounded Nodes	3	4.62
Leaf Nodes	6	9.23

Os relatórios gerados para este problema são apresentados no anexo do problema 1, eles mostram algumas informações extras sobre os resultados, assim como, as sequências para os 6 nós folhas.

5.2 Problema 2

O problema dois é do tipo flow shop com 10 tarefas e quatro estações de trabalho na qual as datas de entrega para todas as tarefas são diferentes. Ele foi gerado arbitrariamente. Algumas tarefas não visitam todas as máquinas e seus tempos nessas máquinas são considerados zero. Os dados para o problema 2 sào apresentados nas tabelas 5-4 e 5-5.

A sequência com o melhor custo de lateness é a sequência vencedora e as tarefas são 3-7-4-9-6-2-8-5-10-1, nesta ordem de processamento. Outros resultados são apresentados na tabela 5-6.

Tabela 5-4. Tempos de Processamento do Problema 2.

Job	W1	W2	W3	W4
1	(10,11,12,13)	(5,6,7,7)	(1,3,5,7)	(5,6,7,8)
2	(2,3,4,9)	(1,3,4,5)	(2,3,4,5)	(10,11,12,13)
3	(0,0,0,0)	(1,3,5,7)	(3,4,6,10)	(5,6,7,7)
4	(1,3,4,5)	(7,8,9,10)	(2,3,4,9)	(3,8,9,10)
5	(5,6,7,8)	(8,9,10,11)	(4,5,7,8)	(1,2,3,4)
6	(3,8,9,10)	(4,5,7,8)	(5,6,7,7)	(0,0,0,0)
7	(1,1,3,4)	(2,3,4,9)	(1,3,4,5)	(3,4,6,10)
8	(1,3,5,7)	(3,8,9,10)	(10,11,12,13)	(1,3,4,5)
9	(3,4,6,10)	(0,0,0,0)	(1,1,3,4)	(7,8,9,10)
10	(5,6,7,7)	(10,11,12,13)	(1,3,5,7)	(5,6,7,8)

Tabela 5-5. Datas de Entrega do Problema 2.

Job	Due date	Job	Due date
1	(49,70,86,100)	6	(12,19,23,25)
2	(15,20,24,32)	7	(7,11,17,28)
3	(9,13,18,24)	8	(15,25,30,35)
4	(13,22,26,34)	9	(11,13,18,24)
5	(18,22,27,31)	10	(45,65,81,96)

O comportamento do Branch and Bound durante a busca é apresentado na tabela 5-7. Uma atenção especial deve ser dada ao número de nós escapados. Ele é obtido através do cálculo de todos os filhos de um nó que não foram expandidos mais o próprio nó não expandido.

Tabela 5-6. Resultados do Problema 2.

Job	Completion Time	Lateness	Cost
3	(9,13,18,24)	(-15,-5,5,15)	(-150,-50,50,150)
7	(12,17,24,34)	(-16,0,13,27)	(-310,-50,180,420)
4	(15,25,33,44)	(-19,-1,11,31)	(-500,-60,290,730)
9	(22,33,42,54)	(-2,15,29,43)	(-520,90,580,1160)
6	(22,33,42,54)	(-3,10,23,42)	(-550,190,810,1580)
2	(32,44,54,67)	(0,20,34,52)	(-550,390,1150,2100)
8	(33,47,58,72)	(-2,17,33,57)	(-570,560,1480,2670)
5	(31,50,64,77)	(0,23,42,59)	(-570,790,1900,3260)
10	(40,61,76,91)	(-46,-11,20,56)	(-616,779,1920,3316)
1	(45,67,83,99)	(10,36,57,78)	(-516,1139,2490,4096)

Tabela 5-7. Comportamento do Branch and Bound no Problema 2.

Parameter	Amount	Percentage
Total nodes	9.8641e+06	100
Expanded Nodes	63649	0.645
Skipped Nodes	9.80045e+06	99.355
Bounded Nodes	48854	0.5
Leaf Nodes	31	3.14e-04

Defuzificando-se o custo da sequência vencedora através do centróide, é obtido 1800.02 com uma dispersão de 981.00.

5.3 Problema 3

O problema 3, assim como os demais, é do tipo flow shop com 10 tarefas e 10 estações de trabalho na qual os tempos de processamento e as datas de entrega foram geradas através de um gerador de problemas que trabalha com o FSPS. Ele gera tempos de processamento e datas de entrega usando distribuições uniformes para o centróide e a dispersão. O arquivo de entrada alimenta-se das seguintes informações:

- Número de tarefas;
- Número de máquinas;
- Tempos de Processamento
    - Centróide uniforme (min max);
    - Otimista uniforme (min max);
    - Regular uniforme (min max);

- Pessimista uniforme (min max).

- Datas de Entrega

  - Centróide uniforme (min max);
  - Otimista uniforme (min max);
  - Regular uniforme (min max);
  - Pessimista uniforme (min max).

Devido ao tamanho do problema 3 estas informações e seus relatórios são apresentados no anexo do problema 3. Como na seção anterior, os resultados e o comportamento do algoritmo de busca estão apresentados nas tabelas 5-8 e 5-9. Eles estão apresentados como valores inteiros, por questão de espaço, mas são valores decimais. O anexo do problema 3 traz os valores exatos.

Tabela 5-8. Resultados do Problema 3

Job	Completion Time	Lateness	Cost
1	(117,125,158,168)	(-28,4,66,98)	(-28,4,66,98)
2	(138,146,180,191)	(-19,16,88,119)	(-48,20,153,217)
4	(149,159,196,208)	(-19,-1,11,31)	(-82,21,235,332)
8	(164,176,216,196)	(-35,0,82,120)	(-434,21,1054,1530)
9	(180,193,235,249)	(-65,-28,36,75)	(-500,-8,1090,1604)
6	(188,202,247,263)	(-65,-28,38,75)	(-1150,-289,1466,2358)
3	(204,219,268,286)	(-74,-38,44,85)	(-1896,-672,1901,3210)
5	(222,237,288,307)	(-79,-36,42,80)	(-1975,-708,1944,3290)
7	(229,246,301,320)	(-80,-42,41,84)	(-2776,-1126,2356,4132)
10	(241,258,316,336)	(-74,-33,68,109)	(-3516,-1459,3038,5224)

Tabela 5-9. Comportamento do Branch and Bound no Problema 3

Parameter	Amount	Percentage
Total nodes	9.8641e+06	100
Expanded Nodes	283691	2.876
Skipped Nodes	9580410	97.124
Bounded Nodes	148796	1.508
Leaf Nodes	17868	0.181

5.4 Análise dos Resultados

O B&B foi criticado durante algum tempo devido ao tempo necessário para se obter uma resposta para problemas grandes. Com os avanços em hardware e software este problema está se tornando menos importante. Na verdade, para o problema 1, que é um problema pequeno, o resultado é praticamente imediato. Os problemas 2 e 3 levaram poucos segundos para sua execução num computador Pentium 233 MHz com 64 MB de memória RAM. Melhoramentos tem sido feitos no programa no sentido de se obter resultados ainda mais rapidamente.

Considerando-se os resultados obtidos, pode-se afirmar que a sequência vencedora é fortemente dependente das datas de entregas. Era esperado que as datas de entrega tivessem esta importância.

Um outro ponto a ser analisado é o comportamento do B&B. Os resultados mostram que para problemas maiores o teste de viabilidade pode cortar ainda mais nós, por exemplo, problemas 2 e 3.

Os resultados obtidos para estes três problemas mostram que a incerteza pode ser considerada em problemas de programação da produção e, quando dados estatísticos não são disponíveis ou a variabilidade do processamento é grande, a abordagem difusa é capaz de gerar bons resultados. Da mesma forma, a abordagem

difusa permite a quantificação do grau de satisfação e preferências para tempos de processamento e datas de entregas.

Um outro problema que foi testado usando esta abordagem refere-se ao exemplo apresentado por McCAHON et al [5]. O resultado obtido para a minimização do makespan foi melhor que o resultado alcançado por McCAHON et al [5]. Este problema não foi incluído aqui porque seu objetivo era a minimização do makespan e não da função de penalidade para lateness. A despeito do melhor resultado, ele será incluído em futura publicação pelo autor desta tese.

6 Conclusões e Sugestões

Considerando-se os objetivos mencionados no capítulo um deste trabalho, algumas conclusões importantes podem ser inferidas.

Problemas de scheduling podem ser e devem ser modelados como sistemas difusos. A incerteza relativa a este tipo de problema não pode ser coberta somente através da abordagem estocástica. Por exemplo, a probabilidade de cara ou coroa no lançamento de uma moeda é de 50% para cada uma. Entretanto é possível lançar-se a moeda várias vezes e obter-se um resultado que é diferente de para cada face da moeda.

A abordagem difusa para problemas de scheduling pode generalizar a abordagem crisp permitindo ao tomador de decisão incorporar preferências, bem como incertezas. Informação incompleta sobre os tempos de processamento e datas de entrega são usadas e isto também ajuda a generalizar a abordagem crisp.

Programadores humanos tem preferências e eles pensam considerando intervalos para os tempos de processamentos das tarefas. Mesmo quando eles usam o flowtime ou makespan para tomarem suas decisões, a decisão final recai no custo do programa de produção em questão. Portanto, esta maneira de programar a produção aproxima-se da maneira como humanos programam tarefas, porque ela considera custos e tempos como intervalos.

O modelo minimiza o custo total de lateness (earliness e tardiness). Ele busca em todas as sequências viáveis qual é a de mínimo custo de lateness.

A função de penalidade de custo usada como função objetivo é um parâmetro de decisão que avalia a viabilidade das sequências. Obviamente, se uma certa tarefa tem seu tempo de término fora do intervalo da data de entrega, isto significa que o grau de satisfação para que aquela tarefa esteja naquela posição nesta sequência é zero. Consequentemente, esta sequência é inviável.

A formulação usada aqui permite que diferentes “intervalos de confiança possibilísticos” possam ser usados. Isto significa que é possível definir um grau de certeza para o tempo de processamento e datas de entregas fazendo-se as restrições soft, mais ou menos restritivas.

Pode-se também concluir que o modelo proposto nesta tese é mais realista que as abordagens que não incorporam incertezas relativas aos processos de produção.

Como sugestões, seria importante continuar a pesquisa sobre a abordagem difusa para estes problemas. Como exemplos, outras funções objetivo devem ser otimizadas, como, flowtime, flowtime médio, makespan, etc.

Outro campo de atuação interessante refere-se as relações de precedência entre tarefas, que foram mencionadas neste trabalho, mas não foram incorporadas ao modelo. Avaliações difusas das relações de precedência podem permitir todo um novo tratamento da composição de custos no ambiente de produção.

Como uma última conclusão, este sistema permite que tomadores de decisões reprogramem tarefas mais facilmente, considerando intervalos. Também as datas de entrega podem ser melhor estabelecidas desta forma.

7 Referências

[1] GRAVES, S. C. -*"A review of production scheduling"*- Operations research. Vol. 29, n 4, 1981. Pp. 646-675.

[2] DUBOIS, D.; FAGIER, H.; PRADE, H. -*"Fuzzy constraints in job shop scheduling"*- Journal of Intelligent Manufacturing. Vol. 6, 1995. Pp. 215-234.

[3] BLAZEWICZ, J.; ECKER, K. H.; PESCH, E. SCHMIDT, G.; WEGLARZ, J. – "Scheduling Computer and Manufacturing Process"- Springer–Verlag, Berlin, 1996. 491 p.

[4] WANG, H. G.; ROODA, J.E.; HAAN, J.F. -*"Solve scheduling problems with a fuzzy approach"* - Proceedings of the 1996 5th IEEE International Conference on Fuzzy Systems. Part 1 (of 3), New Orleans, USA. Pp 194-198.

[5] McCAHON, C. S.; LEE, E. S. –*"Fuzzy Job Sequencing for a Flowshop"*- European Journal of Operational Research, vol 62, 1992. Pp 294 – 301.

[6] FAGIER, H. -*"Fuzzy scheduling principles and experiments"*- Chapter 39 in Fuzzy Information Engineering. A guided Tour of Applications Edited by: Didier Dubois, Henri Prade, Ronald R. Yager. John Wiley & Sons, Inc. 1997, New York 712 p.

[7] ISHIBUCHI, H.; YAMAMOTO, N.; MISAKI, S.; TANAKA, H. -*"Local search algorithms for flow shop scheduling with fuzzy due dates"*- International Journal of Production Economics. Vol. 33(1), 1994. Pp. 53-66.

[8] HAN, S.; ISHII, H.; FUJII, S. -*" One machine scheduling problem with fuzzy due dates"*- European Journal of Operational Research. Vol. 79, 1994. Pp. 1-12.

[9] ISHII, H.; TADA, M. -*"Single machine scheduling problem with fuzzy precedence relation"*- European Journal of Operational Research. Vol. 87, 1995. Pp. 284-288.

[10] TURKSEN, I. B. -*"Scheduling system design: Three fuzzy theory approach"*- Fuzzy Information Engineering, Chapter 40

[11] KURODA, M.; WANG, Z. -*"Fuzzy job shop scheduling"*- International Journal of Production Economics. Vol 44, 1996. pp 45-51.

[12] GRABOT, B.; GENESTE, L. -*"Dispatching rules in scheduling: a fuzzy approach"*- International Journal of Production Research. Vol 32, n 4 ,1994. pp 903-915.

[13] CUSTODIO, L. M. M.; SENTIEIRO, J. J. S.; BISPO, C. F. G. -*"Production planning and scheduling using a fuzzy decision system"*- IEEE Transactions on Robotics and Automation. Vol 10, n 2, April 1994. pp 160-168.

[14] DUBOIS, D.; PRADE, H. –"*A Survey of Engineering Applications*"- European Symposium on Computer Aided Process Engineering, pp 373-380. REFERENCIA IMCOMPLETA.

[15] ISHIBUCHI, H.; YAMAMOTO, N.; MURATA, T.; TANAKA, H. –*"Genetic Algorithms and Neighborhood Search Algorithms for Fuzzy Flow-Shop Scheduling Problems"*- Fuzzy Sets and Systems, vol 67, 1994. Pp 81 – 100.

[16] ISHIBUCHI, H.; MURATA, T., LEE, K. H. –*"Formulation of Fuzzy Flowshop Scheduling Problems with Fuzzy Processing Time"*- Proceedings of the 1996 5th International Conference on Fuzzy Systems. Part 1 (of 3), New Orleans LA. Pp. 199–205.

[17] TSUJIMURA, Y.; GEN, M.; KUBOTA, E. –*"Solving Job-Shop Scheduling Problems with Fuzzy Processing Time Using Genetic Algorithms"*- Journal of Japan Society for Fuzzy Systems, vol 7 n 5, 1995. Pp 1073 – 1083.

[18] AMADOPOULUS, G. I.; PAPPIS, C. P. –*"A Fuzzy Linguistic Approach to a Multi-Criteria Sequencing Problem"*- European Journal of Operational Research, vol. 92, 1996. Pp. 628–636.

[19] VIEGAS, J. M. N.; CUSTÓDIO, L. M. M.; PINTO-FERREIRA, C. A. –"*A Constraint Management Approach to Project Job-Shop Scheduling"*- Incomplete reference.

[20] ERSCHELER, J.; ROUBELLAT, F.; VERNHES, J. P. –*"Finding Some Essential Characteristics of the Feasible Solution for a Scheduling Problem"*- Technical Notes on Operations Research, vol. 24(4), July-August 1976, pp. 775-783.

[21] CONWAY, R. W.; MAXWELL, W. L.; MILLER, L. W. – "Theory of Scheduling" – Addison-Wesley Publishing Company, 1967, 294 p.

[22] RARDIN, R. –*"Optimization in Operations Research"*- Prentice Hall, 1998, pp. 609-613.

[23] SLANY, W -*"Scheduling as a fuzzy multi-criteria optimization problem"*- Fuzzy Sets and Systems. Vol 78, 1996. pp 197-222.

[24] RENDER, B. & STAIR, R. M. – *"Quantitative Analysis for Management"*- Prentice-Hall Inc. New Jersey sixth edition, 1997, pp. 522-531.

[25] HOROWITZ, E.; SAHNI, S.; RAJASEKARAN, S. – "*Computer Algorithms C++*" Computer Science Press, New York, 1997, 769 p.

[26] AVRIEL, M.; GOLANY, B. - *"Mathematical Programming for Industrial Engineers"*- Marcel Dekker, Inc., New York, 1996, 637 p.

[27] JOHNSON, S. M. – *"Optimal Two- and Three-Stage Production Schedules with Setup Times Included"* – Naval Research Logistics Quarterly. Vol. 1, 1954. Pp. 61-68.

[28] STANFIELD, P. M.; KING, R. E.; JOINES, J. A. -*"Scheduling Arrivals to a Production System in a Fuzzy Environment"*- European Journal of Operational Research vol. 93, 1996, pp. 75-87.

[29] McMAHON, G. B.; LIM, C. J. -*"The Two Machine Flow Shop Problem with Arbitrary Precedence Relations"*- European Journal of Operational Research vol. 64, 1993, pp. 249-257.

[30] GIM, B.; CURRY, G. L.; DEUERMEYER, B. L. -*"Two-Machine Flow-Shop Sequencing with Sparse Precedence Constraints"*- Computers Industrial Engineering vol. 26(1), 1994, pp. 173-180.

[31] PATTERSON, J. H.; TALBOT, F. B.; SLOWINSKI, R.; WEGLARZ, J. - *"Computational Experience with a Backtracking Algorithm for Solving a General Class of Precedence and Resource-Constrained Scheduling Problems"*- European Journal of Operational Research vol. 49, 1990, pp. 68-79.

[32] CHENG, J.; KISE, H.; MATSUMOTO, H. –*"A Branch-and-Bound Algorithm with Fuzzy Inference for a Permutation Flow-Shop Scheduling Problem"*- European Journal of Operational Research, vol 96, n 3 Feb 1, 1997. Pp 578-590.

[33] BRAH, S. A.; HUNSUCKER, J. L. - *"Branch and Bound Algorithm for the Flow Shop with Multiple Processors"*- European Journal of Operational Research vol. 51, 1991, pp. 88-99

[34] NAGAR, A.; HERAGU, S. S.; HADDOCK, J. - *"A Branch and Bound Approach for a Two Machine Flowshop Scheduling Problem"*- Journal of the Operational Research Society vol. 46, 1995, pp. 721-734.

[35] CARLIER, J.; PINSON, E. -*"An Algorithm For Solving The Job-Shop Problem"*- Management Science, vol. 35(2), February 1989, pp. 164-176.

[36] CARLIER, J.; PINSON, E. -*"Adjustments of Heads and Tails for the Job-Shop Problem"*- European Journal of Operational Research, vol. 78, 1994, pp. 146-161.

[37] DUBOIS, D.; PRADE, H. -"*Constraint Satisfaction and Decision Under Uncertainty Based on Qualitative Possibility Theory*"- Proceedings of The 1997 IEEE International Conference on Fuzzy Systems, July 1-5, 1997 Barcelona Spain, pp. 23-30.

[38] OW, P. S.; MORTON, T. E. -*"The Single Machine Early/Tardy Problem"*- Management Science, vol. 35(2) February 1989, pp. 177-191.

[39] KAUFMANN, A.; GUPTA, M. M. –"*Fuzzy Mathematical Models in Engineering and Management Science*"- Elsevier Science Publisher Company Inc., New York, 1988, 338 p.

[40] DUBOIS, D.; PRADE, H. –"*Fuzzy Sets and Systems Theory and Applications*"- Academic Press, New York, 1980, 393 p.

[41] ZIMMERMANN, H. J. –"*Fuzzy Set Theory and its Application*"- Kluwer Academic Publishers, Boston, 1996, 435 p.

8 Bibliografia

ALMOND, R. G. – *"Discussion: Fuzzy Logic: Better Science? Or Better Engineering?"* - Technometrics, August 1995, vol. 37(3), pp. 267-270.

AZMI, Z. A. -*"New Fuzzy Approaches by Using Statistical and Mathematical Methodologies in Operations Research"*- The Journal of Fuzzy Mathematics vol. 1(1) 1993, pp. 69-87.

BENSANA, E.; BEL, G.; DUBOIS, D. –*"Opal: A multi-Knowledge Based System for Industrial Job-Shop Scheduling"*- International Journal of Production Research, vol 26, 1988. Pp 795 –819.

BERRAH, L.; MAURIS, G.; FOULLOY, L.; HAURAT, A. -"*A Fuzzy Approach for the Performance Evaluation of Manufacturing Processes*"- Proceedings of The 1997 IEEE International Conference on Fuzzy Systems, July 1-5, 1997 Barcelona Spain, pp. 951-956.

BEZDEK, J. –*"Editorial: Fuzzy Models – What Are They, and Why?"*- IEEE Transactions on Fuzzy Systems, vol 1, n 1, 1993. Pp 1 – 5.

BOGARIN, J. A.; EBECKEN, N. F. F. -"*Expert Scheduling of Petroleum Support Vessels for Offshore Applications*"- Proceedings of The 1998 IEEE International Conference on Fuzzy Systems, May 4-9, 1998 Anchorage, Alaska, USA, pp. 1048-1053.

BONISSONE, P. P. – *"Discussion: Fuzzy Logic Control Technology: A Personal Perspective"* – Technometrics, August 1995, vol. 37(3), pp. 262-266.

BOOCH, G. -"*Object Oriented Analysis and Design with Applications*"- Addison-Wesley, California 2nd edition, 1994, 589p.

BOSTAN, B.; YAZICI, A. -"*A Fuzzy Deductive Object Oriented Database Model*"- Proceedings of The 1998 IEEE International Conference on Fuzzy Systems, May 4-9, 1998 Anchorage, Alaska, USA, pp. 1361-1366.

CARLSON, C.; TAVARES, H. M. F.; FORMIGONI, J. R. F. -"*Dealing with Fuzzy Costs in Telecommunications Network Design*"- Proceedings of The 1998 IEEE International Conference on Fuzzy Systems, May 4-9, 1998 Anchorage, Alaska, USA, pp. 634-639.

CHANTRAPORNCHAI, C.; TONGSIMA, S.; SHA, E. H. M. -"*Imprecise Task Scheduling Optimization*"- Proceedings of The 1997 IEEE International Conference on Fuzzy Systems, July 1-5, 1997 Barcelona Spain, pp. 1265-1270.

CHEESEMAN, P. – *"Discussion: Fuzzy Thinking"* - Technometrics, August 1995, vol. 37(3), pp. 282-283.

CHENG, T. C. E.; CHEN, Z. L. –*"Parallel-Machine Scheduling Problem with Earliness and Tardiness Penalties"*- Journal of Operational Research Society, vol 45, n 6, 1994. Pp 685 – 695.

CUSTODIO, L. M. M.; BISPO, C. F. G.; SENTIEIRO, J. J. S. –*"A Hierarchical Fuzzy Control System for Short-Range Planning and Scheduling"*- Proceedings of the Third International Conference on Computer Integrated Manufacturing, R. P. I. Troy, NY, USA. May 1992.

CUSTODIO, L. M. M.; BISPO, C. F. G.; SENTIEIRO, J. J. S. –*"Fuzzy Logic Applied to Production Planning and Scheduling"*- Technical Report ISR/CIM 93 – 1, IST 1993.

CUSTÓDIO, L. M. M.; PINTO-FERREIRA, C. A. –"*Issues and Difficulties in the Design and Control of Manufacturing Systems*"- INTERNET.

DEVEDZIC, G. -"*Fuzzy Sets Based Global Evaluation in Metal Cutting Process Planning*"- Proceedings of The 1998 IEEE International Conference on Fuzzy Systems, May 4-9, 1998 Anchorage, Alaska, USA, pp. 891-896.

DUBOIS D. -*"Fuzzy Knowledge in AI System for Job-Shop Scheduling"*- In Applications of Fuzzy Set Methodologies In Industrial Engineering. Eds Evan, Karwowsky and Wilhem pp 1989. Elseiver, 1989.

DUBOIS, D.; FAGIER, H.; PRADE, H. - *"Propagation and Satisfaction of Flexible Constraints"*- Fuzzy Sets, Neural Networks, and Soft Computing. Edited by Ronald R. Yager and Lotfi A. Zadeh, Van Nostrand Reinhold, New York, 1994, 440 p.

DUBOIS, D.; KONING, J. L. – *"A Decision Engine Based on Rational Aggregation of Heuristic Knowledge"* – Decision Support Systems. N. 11, 1994. Pp. 337-361.

DUBOIS, D.; PRADE, H. –*"A Survey of Belief Revision and Updating Rules in Various Uncertainty Models*"*- International Journal of Intelligent Systems, vol. 9 1994, pp 61 – 100.

DUBOIS, D.; PRADE, H. –*"Possibility Theory, An Approach to Computerized Processing of Uncertainty"*- New York. Plenum Press, 1988.

DUBOIS, D.; PRADE, H. –*"What Are Fuzzy Rules and How to Use Them"*- Fuzzy Sets and Systems vol 84, 1996 pp 169 – 185.

EHLERS, E. M.; VAN RENSBURG, E. –*"An Object Oriented Manufacturing Scheduling Approach"*- IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, vol 26 n 1, January 1996. Pp 17 – 26.

EHLERS, E. M.; VAN RENSBURG, E.; VAN DER MERWE, R. L. – *"Intelligent Scheduling in the Manufacturing Environment"* – Computers and Industrial Engineering. Vol. 27, n. 1-4, 1994. Pp. 31-34.

ESCODA, I.; ORTEGA, A.; SANZ, A., HERMES, A. -"*Demand Forecast by Neuro-Fuzzy Techniques*"- Proceedings of The 1997 IEEE International Conference on Fuzzy Systems, July 1-5, 1997 Barcelona Spain, pp. 1381-1386.

EWACHA, K.; RIVAL, I. -*"Permutation Schedule for Flow Shops with Precedence Constraints"*- Operations Research, vol. 38(6), November-December 1990, pp. 1135-1139.

FORTEMPS, P. -"*Job Shop Scheduling With Imprecise Duration: A Fuzzy Approach*"- IEEE Transactions on Fuzzy Systems, vol. 5(4), November 1997, pp. 557-569.

GERSTORFER E.; HELLENDOORN, H. -"*On The Role of Fuzzy Logic in Production Planning"*- Proceedings of The 1997 IEEE International Conference on Fuzzy Systems, July 1-5, 1997 Barcelona Spain, pp. 1271-1275.

GERSTORFER, E.; HOLLATZ, J. -"*Strategic Production Planning with Neuro-Fuzzy Systems*"- Proceedings of The 1998 IEEE International Conference on Fuzzy Systems, May 4-9, 1998 Anchorage, Alaska, USA, pp. 1206-1210.

GOMES, R.; PACHECO, R.; MARTINS A.; WEBER, R. BARCIA, R. -"*Product Pricing Decision Support Fuzzy Systems Through the Internet*"- Proceedings of The 1998 IEEE International Conference on Fuzzy Systems, May 4-9, 1998 Anchorage, Alaska, USA, pp. 1664-1669.

GRABOT, B. et al *–"A multi-Heuristic Scheduling: Three Approach to Tune Compromises"*- Journal of Intelligent Manufacturing, vol 5, 1994. Pp 303 – 313.

GRABOT, B.; GENESTE, L. *–" Tuning of Fuzzy Rules for Multiobjective Scheduling"*- Fuzzy Information Engineering, Chapter 41.

HAPKE, M.; JASZKIEWICZ, A.; SLOWINSKI, R. -"*Fuzzy Project Scheduling with Multiple Criteria*"- Proceedings of The 1997 IEEE International Conference on Fuzzy Systems, July 1-5, 1997 Barcelona Spain, pp. 1277-1282.

HAPKE, M.; KOMINEK, P.; SLOWINSKI, R. – *"FPS. Fuzzy Project Scheduling"* – Institute of Computing Science, Poznañ University of Technology. Fps_eng.html at www.cs.put.poznan.pl.

HISDAL, E. - *"The Philosophical Issues Raised by Fuzzy Set Theory"*- Fuzzy Sets and Systems vol. 25, 1988, pp. 349-356.

HUBER, K. P.; BERTHOLD, M. R. -"*Applications of Fuzzy Graphs For Metamodeling*"- Proceedings of The 1998 IEEE International Conference on Fuzzy Systems, May 4-9, 1998 Anchorage, Alaska, USA, pp. 640-644.

ISHII, H.; TADA, M.; MASUDA, T. –*"Two Scheduling Problems with Fuzzy Due-Dates"*- Fuzzy Sets and Systems, vol 46, 1992. Pp 339 – 347.

KANDEL, A. –"*Fuzzy Expert Systems*"- CRC Press, Boca Raton, Ca. 1992.

KANDEL, A., MARTINS, A., PACHECO R. – *"Discussion: On the Very Real Distinction Between Fuzzy and Statistical Methods"* - Technometrics, August 1995, vol. 37(3), pp. 276-281.

KARLOF, J. K.; WANG, W. -*"Bilevel Programming Applied to the Flow Shop Scheduling Problem"*- Computers Operations Research vol. 23(5), 1996, pp. 443-451.

KERR, R.; WALKER, R. –*"A Job-Shop Scheduling System Based on Fuzzy Arithmetic"*- Proceedings 3rd International Conference on Expert Systems and The Leading Edge in Production and Operation Management. Pp 433-450. Hilton Head Isl., South Car. 1989.

KUBOTA, N.; ARAKAWA, T.; FUKUDA, T.; SHIMOJIMA, K. -"*Fuzzy Manufacturing Scheduling by Virus-Evolutionary Genetic Algorithm in Self-Organizing Manufacturing System*"- Proceedings of The 1997 IEEE International Conference on Fuzzy Systems, July 1-5, 1997 Barcelona Spain, pp. 1283-1288.

LAM, S. S.; CAI, X. -"*Minimizing Earliness and Tardiness of Job Completions about a Fuzzy Due Date*"- Proceedings of The 1998 IEEE International Conference on Fuzzy Systems, May 4-9, 1998 Anchorage, Alaska, USA, pp. 869-872.

LAVIOLETTE, M.; SEAMAN, J. W.; BARRETT, J. D.; WOODALL, W. H. – *"A Probabilistic and Statistical View of Fuzzy Methods"* – Technometrics. August, vol. 37, n. 3, 1995. Pp. 249-261.

LAVIOLETTE, M.; SEAMAN, J. W.; BARRETT, J. D.; WOODALL, W. H. – *"Reply"* - Technometrics, August 1995, vol. 37(3), pp. 287-292.

LI, R. K.; SHYU, Y. T.; ADIGA, S. –*"A Heuristic Rescheduling Algorithm for Computer-Based Production Scheduling Systems"*- International Journal of Production Research, vol 31, n 8, 1993. Pp 1815 – 1826.

LIM, C. J.; MAcMAHON, G. B. -*"The Three Machine Flow-Shop Problem With Arbitrary Precedence Relations"*- European Journal of Operational Research vol. 78, 1994, pp. 216-223.

LOPES, O. C. –*"Notas de aula Programação e Sequenciamento"*- Programa de Pós-Graduação em Engenharia de Produção e Sistemas. Universidade Federal de Santa Catarina, capítulo 2. 1997.

LÓPEZ, D. R.; JIMÉNEZ, C. J.; BATURONE, I.; BARRIGA, A.; SOLANO, S. S. - "*Xfuzzy: A Design Environment for Fuzzy Systems*"- Proceedings of The 1998 IEEE International Conference on Fuzzy Systems, May 4-9, 1998 Anchorage, Alaska, USA, pp. 1060-1065.

LORENZ, M. -"*Object Oriented Software Development. A Practical Guide*"- Prentice-Hall, New Jersey, 1993, 227p.

MAMDANI, E. H. –*"Applications of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis"*- IEEE transactions Comput.. Vol 26 1977. pp 1182 – 1191.

MASNATA, A.; SETTINERI, L. -"*An Application of Fuzzy Clustering to Cellular Manufacturing* "- International Journal Of Production Research, vol. 35(4), 1997, pp 1077 - 1094.

MESEGUER, P.; LARROSA, J. -"*Solving Fuzzy Constraint Satisfaction Problems*"- Proceedings of The 1997 IEEE International Conference on Fuzzy Systems, July 1-5, 1997 Barcelona Spain, pp. 1233-1238.

MIYAMOTO, S. -"*Indexed Rough Approximations and Generalized Possibility Theory*"- Proceedings of The 1998 IEEE International Conference on Fuzzy Systems, May 4-9, 1998 Anchorage, Alaska, USA, pp. 791-795.

MONTAZERI, M.; VAN VASSENHOVE, L. N. –*"Analysis of Scheduling Rules for a FMS"*- International Journal of Production Research, vol 28, n 4. 1990. Pp 785 – 802.

MURATA, T.; ISHIBUCHI, H.; LEE, K. H. -"*Reformulation of Various Non-Fuzzy Scheduling Problems Using the Concept of Fuzzy Due-Date*"- Proceedings of The 1997 IEEE International Conference on Fuzzy Systems, July 1-5, 1997 Barcelona Spain, pp. 447-452.

NEGOITA, C.V. –"*Expert Systems and Fuzzy Systems*"- Benjamin/Cummings, Menlo Park, Ca. 1985.

PAN, C. H.; - *"A Study of Integer Programming Formulations for Scheduling Problems"*- International Journal of Systems Science, vol. 28(1), 1997, pp. 33-41.

PAN, Y.; KLIR, G. J.; YUAN, B. -*"Bayesian Inference Based on Fuzzy Probabilities"*- Proceedings of IEEE 5th International Fuzzy Systems, New Orleans, LA, USA, September 8-11, 1996, pp. 1693-1699.

PAWLAK, Z. -"*Granularity of Knowledge, Indiscernibility and Rough Sets*"- Proceedings of The 1998 IEEE International Conference on Fuzzy Systems, May 4-9, 1998 Anchorage, Alaska, USA, pp. 106-110.

PINEDO, M. –*"Scheduling Theory, Algorithms and Systems"*- New Jersey, Prentice-Hall, 1995. Chapters 1, 2, 6 and 7.

PIRAMUTHU, S.; RAMAN, N.; SHAW, M. J. –*"Learning Based Scheduling in a Flexible Manufacturing Flow Line"*- IEEE Transactions on Engineering Management vol 41, n 2, May 1994. Pp 172 – 182.

PRADE, H. M. –*"Using Fuzzy Set Theory in a Scheduling Problem: A case Study"*- Fuzzy Sets and Systems, vol 2, 1979. Pp 153 – 165.

RINNKS, D. B. –*"A Heuristic Approach to Aggregate Production Scheduling Using Linguistic Variables"*- In Methodology and Applications in Fuzzy Sets and Possibility Theory, Recent Developments. Ed. R.R. Yager. Pp 562 – 581. New York: Pergamon Press, 1982.

ROMERO, P. R.; YAMAKAMI, A. -*"Fuzzy Number Ordering Methods Applied to Workpiece Scheduling Problem with Imprecise Parameters*"- Proceedings of The 1997 IEEE International Conference on Fuzzy Systems, July 1-5, 1997 Barcelona Spain, pp. 421-426.

ROUSSEEUW, P. J. – *"Discussion: Fuzzy Clustering At the Intersection"* - Technometrics, August 1995, vol. 37(3), pp. 283-286.

SARKAR, M.; YEGNANARAYANA, B. -"*Rough-Fuzzy Membership Functions*"- Proceedings of The 1998 IEEE International Conference on Fuzzy Systems, May 4-9, 1998 Anchorage, Alaska, USA, pp. 796-801.

SARKAR, U. K.; CHAKRABARTI, P. P.; GHOSE, S.; DE SARKAR, S. C. -*"Effective Use of Memory in Iterative Deepening Search"*- Information Processing Letters vol. 42, 1992, pp. 47-52.

SEN, T.; GUPTA, S. –*"A State-of-Art Survey of Static Scheduling Research Involving Due Dates"*- OMEGA The International Journal of Management Science, vol 12 n 1, 1984. Pp 63 – 76.

SLANY, W. –"*Fuzzy Expert System to Predict Maintenance Intervals in a Continuous Caster*"- CPC-93 International Conference on Computerized Control in Steel Plants, Seoul, Korea, Nov. 1993. INTERNET.

SLANY, W. –*"Fuzzy Scheduling"*- Ph.D. Thesis, C Doppler Laboratory for Expert Systems, Austria, 1995.

SLANY, W.; STARY, C.; DORN, J. –"*Vague Data Management in Production Process Scheduling Applied to High-Grade Steelmaking*"- Proceedings of the First International Conference on Artificial Intelligence Planning Systems. Morgan Kaufmann Publishers, Inc., San Mateo, Ca. 1992.

SUBASIC, P.; NAKATSUYAMA, M. -"*A New Representational Framework for Fuzzy Sets*"- Proceedings of The 1997 IEEE International Conference on Fuzzy Systems, July 1-5, 1997 Barcelona Spain, pp. 1601-1606.

SUGENO, M.; YASUKAWA, T. –*"A Fuzzy Logic-Based Approach to Qualitative Modeling"*- IEEE Transactions of Fuzzy Systems, vol 1, n1, 1993. Pp 1 – 24.

SUN, K. T. -"*A Two Dimensional Fuzzy Ranking Approach to Job Scheduling Problems*"- Proceedings of The 1998 IEEE International Conference on Fuzzy Systems, May 4-9, 1998 Anchorage, Alaska, USA, pp. 873-878.

TAN, S.; ZHANG, L.; VANDEWALLE, J. -"*On The Learning of Min-Max Fuzzy Systems*"- Proceedings of The 1997 IEEE International Conference on Fuzzy Systems, July 1-5, 1997 Barcelona Spain, pp. 1581-1584.

TANAK, H.; LEE, H.; GUO, P. -"*Possibility Data Analysis with Rough Sets Concept*"- Proceedings of The 1998 IEEE International Conference on Fuzzy Systems, May 4-9, 1998 Anchorage, Alaska, USA, pp. 117-122.

TORRA, V. -"*Fuzzy Sets as the Aggregation of Weighted Observations*"- Proceedings of The 1997 IEEE International Conference on Fuzzy Systems, July 1-5, 1997 Barcelona Spain, pp. 1333-1337.

TURKSEN, I. B. et al –*"Fuzzy Expert System Shell for Scheduling"*- Proceedings, SPIE 1993. Boston Massachusetts. Vol 2061 ????. pp 308-319.

TURKSEN, I. B.; ULGURAY, D.; WANG, Q. –*"Hierarchical Scheduling Based on Approximate Reasoning – A comparison with ISIS*"*- Fuzzy Sets and Systems, vol 46, 1992. Pp 349 –371.

TURKSEN, I. B.; YURTSEVER, T. –*"Fuzzy Logic Expert System Scheduler"*- Proceedings, 2nd International Conference on Fuzzy Logic and Neural Networks. Iizuka, Japan, 1992. Pp 371 – 374.

UMANO, M.; IMADA, T.; HATONO, I.; TAMURA, H. -"*Fuzzy Object Oriented Databases and Implementation of Its SQL-type Data Manipulation Language*"- Proceedings of The 1998 IEEE International Conference on Fuzzy Systems, May 4-9, 1998 Anchorage, Alaska, USA, pp. 1344-1349.

VIEGAS, J. M. N.; CUSTÓDIO, L. M. M.; PINTO-FERREIRA, C. A. –"*CROSS: A Combined Relation-Operation Based Scheduling System*"- INTERNET.

WANG, H. G.; ROODA, J. E. –*"Fuzzy Control in a Single Scheduling Machine System"*- Accepted for publication in the Journal Production Planning & Control, vol 7, n 6, 1996.

WANG, P. Z.; LUI, H. C.; ZHANG, X. H.; ZHANG, H. M.; XU, W. -*"Win-Win Strategy for Probability and Fuzzy Mathematics"*- The Journal of Fuzzy Mathematics vol. 1(1) 1993, pp. 223-231.

WATANABE, T. –*"Job Shop Scheduling Using a Fuzzy Logic in a Computer Integrated Manufacturing Environment"*- 5th International Conference on Systems Research, Informatics (Germany: Baden-Baden), 1990. Pp 150 – 158.

WELLMAN, M. P.; FORD, M.; LARSON, K. –"*Path Planning Under Time-Dependent Uncertainty*"- In Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI-95), Montreal, Quebec, Canada, August 18-20, 1995. INTERNET.

WURMAN, P. R.; WELLMAN, M. P. –"*Optimal Factory Scheduling Using Stochastic Dominance A**"- In Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence (UAI-96), Portland, OR, USA, August 1996. INTERNET.

YAGER, R. R. -*"Fuzzy Sets, Probabilities, and Decision"*- Journal of Cybernetics vol. 10, 1980, pp. 1-18.

YU, L.; SHIH, H. M.; SEKIGUCHI, T. -"*Dynamic Selection of Dispatching Rules by Fuzzy Inference*"- Proceedings of The 1998 IEEE International Conference on Fuzzy Systems, May 4-9, 1998 Anchorage, Alaska, USA, pp. 979-984.

ZADEH, L. A. – *"Discussion: Probability Theory and Fuzzy Logic Are Complementary Rather Than Competitive"* - Technometrics, August 1995, vol. 37(3), pp.271-276.

ANEXO