



MINIMIZAÇÃO DO TEMPO TOTAL DE ATRASO NO PROBLEMA DE FLOWSHOP COM BUFFER ZERO ATRAVÉS DE BUSCA TABU

Vinícius Amaral Armentano

Faculdade de Engenharia Elétrica e Computação
Universidade Estadual de Campinas
Caixa Postal 6101 – 13083-970 – Campinas – SP
E-mail: vinicius@densis.fee.unicamp.br

Débora Pretti Ronconi

Departamento de Engenharia de Produção
Escola Politécnica – USP
Av. Prof. Almeida Prado, 128 – Cidade Universitária
05508-900 – São Paulo – SP
E-mail: dronconi@usp.br

Resumo

Trabalhos na área de programação de tarefas mostram que atender a data de entrega dos consumidores é uma preocupação crítica na maioria dos sistemas de manufatura. Enquanto existe considerável pesquisa para minimizar o makespan para o problema de flowshop sem restrição de estoque intermediário, poucos trabalhos abordam o critério do atraso total no ambiente flowshop com buffer zero entre as máquinas. Problemas de seqüenciamento com buffer zero ocorrem em processos de manufatura em série que não possuem espaço para armazenamento intermediário. Neste artigo, pesquisa-se a aplicação de busca tabu para explorar o espaço de soluções deste problema. Estratégias adicionais são avaliadas. Testes computacionais são apresentados e comparações com uma adaptação do algoritmo NEH e com um algoritmo Branch-and-Bound são realizadas.

Palavras-chave: *programação de tarefas, flowshop com bloqueio, busca tabu.*

1. Introdução

Este trabalho considera o problema de programação de tarefas no ambiente *flowshop* com buffer zero entre as máquinas. Este ambiente se caracteriza pelo processamento

de n tarefas por m máquinas na mesma ordem, ou seja, a j -ésima operação de todas as tarefas deve ser sempre feita na máquina j . O tempo de processamento de cada tarefa em cada máquina e as datas de entrega são conhecidos. Neste ambiente não existem espaços de armazenamento

entre as máquinas, logo filas de tarefas esperando no sistema pelas próximas operações não são permitidas. Uma tarefa finalizada em uma máquina bloqueia esta máquina até que a próxima máquina esteja disponível para o processamento.

O problema de *flowshop* com buffer zero pode modelar qualquer outro *flowshop* com restrição na capacidade de armazenamento intermediário. Isto é possível porque uma máquina cujos tempos de processamento de todas as tarefas são zero pode representar o espaço de armazenamento de uma tarefa (McCORMICK *et al.*, 1989). Com o desenvolvimento da manufatura *just-in-time* e dos sistemas de controle *kanban* que mantêm um estoque limitado dentro do sistema, o problema de seqüenciamento com buffer limitado assume grande importância (LEISTEN, 1990; HALL & SRISKANDARAJAH, 1996).

A medida de desempenho considerada neste trabalho é a minimização do atraso total das tarefas. O nível de dificuldade deste problema pode ser estimado por meio de um caso particular cuja resolução é comprovadamente complexa: o problema com uma máquina é *NP-hard* para tal critério (DU & LEUNG, 1990). Critérios de otimalidade envolvendo datas de entrega são de grande importância nos sistemas de manufatura pois pode existir uma série de custos quando uma tarefa é entregue com atraso. Dentre estes podem ser citados: penalidades contratuais, perda de credibilidade resultando em uma boa probabilidade de perder o cliente para algumas, ou para todas as possíveis futuras tarefas, e danos na reputação da empresa que poderão afastar outros clientes (SEN & GUPTA, 1984).

Uma extensa revisão bibliográfica sobre programação de tarefas com o critério do atraso total pode ser encontrada em KOULAMAS (1994) e uma revisão específica para o ambiente *flowshop* com buffer ilimitado é apresentada por KIM (1993). Esse autor avalia várias heurísticas para solucionar este problema e recomenda a utilização de uma adaptação do algoritmo NEH

proposto por NAWAZ *et al.* (1983), dado que este método apresentou bons resultados em um tempo razoável. Outros autores trataram deste critério neste ambiente, como por exemplo, TOWNSEND (1977), OW (1985), RAMAN (1995), ADENSO_DÍAZ (1992), KIM (1995) e ARMENTANO & RONCONI (1999).

A metaheurística busca tabu tem sido utilizada com sucesso no problema de *flowshop* com o critério de minimização do *makespan*. A primeira proposta com este enfoque foi apresentada por WIDMER & HERTZ (1989). Outro trabalho interessante, destacando-se o extensivo estudo para a definição dos componentes da busca tabu, foi realizado por TAILLARD (1990) que comparou os resultados do seu algoritmo com uma implementação eficiente do algoritmo NEH e concluiu que a busca tabu obtém melhores resultados. REEVES (1993) destaca em seu trabalho o balanceamento que deve existir entre a diversificação da busca no espaço de soluções e a exploração mais detalhada de certas regiões. O desempenho deste algoritmo superou o proposto por TAILLARD. Recentemente, NOWICKI & SMUTNICKI (1996) propuseram uma heurística baseada em busca tabu que utiliza a idéia de blocos de tarefas sugerida por GRABOWSKI *et al.* (1983). Testes computacionais relatados pelos autores mostraram que o método proposto possui um desempenho melhor, em termos de qualidade e tempo, que outros métodos conhecidos da literatura.

Existem poucas aplicações de busca tabu específicas para minimizar o atraso total. ADENSO_DÍAZ (1992) trata o problema do *flowshop* proporcional e propõe uma redução de duas vizinhanças geradas por movimentos normalmente utilizados: o de inserção e o de troca. No estudo realizado por KIM, a adaptação da busca tabu proposta por WIDMER & HERTZ (1989) apresenta, em média, os melhores resultados com um alto tempo computacional. Porém, em 25% dos problemas o desempenho da busca tabu foi inferior ao de outras heurísticas, inclusive à adaptação do algoritmo NEH. ARMENTANO & RONCONI (1999) também

desenvolveram uma heurística baseada em busca tabu e mostraram que o algoritmo proposto supera o algoritmo NEH em 94% dos problemas testados.

A pesquisa sobre *flowshop* com buffer zero e limitado com o critério do atraso é limitada e a maioria dos estudos considera o critério de minimização do *makespan*. Por tal motivo, apresenta-se a seguir uma pequena revisão considerando este critério. Uma boa revisão deste ambiente pode ser encontrada em HALL & SRISKANDARAJAH (1996). Estes autores provam que o problema de *flowshop* com três máquinas e buffer zero é *NP-hard in the strong sense*. REDDI & RAMAMOORTHY (1972) mostraram que para duas máquinas este problema pode ser reduzido a um caso especial do caixeiro viajante, que pode ser resolvido em tempo polinomial pelo algoritmo de GILMORE & GOMORY (1964). LEVNER (1969) e SUHAMI & MAH (1981) apresentam algoritmos *Branch-and-Bound* que tratam especificamente do ambiente *flowshop* com buffer zero. LEVNER mostra que calcular o *makespan* de um *flowshop* é equivalente a encontrar o caminho mais longo em um grafo direcionado. Resultados computacionais não são apresentados, mas o autor afirma ter encontrado soluções ótimas para problemas com $n \leq 15$. SUHAMI & MAH propõem um algoritmo *Branch-and-Bound* heurístico que utiliza um parâmetro empírico que ajusta o valor do limitante inferior. Para problemas com $m = 3$ e $6 \leq n \leq 10$, este algoritmo usualmente encontra soluções a 1% do valor ótimo. McCORMICK *et al.* (1989) desenvolveram uma heurística conhecida por *Profile Fitting* na qual procura-se seqüenciar primeiro as tarefas que acarretam uma menor soma dos tempos de ociosidade e de bloqueio nas máquinas. Uma abordagem mais ampla é apresentada por LEISTEN (1990) que propôs duas heurísticas que tratam o problema de *flowshop* de permutação e não-permutação com buffer finito e ilimitado. O principal objetivo destas heurísticas é maximizar o uso dos buffers e minimizar o bloqueio das máquinas. LEISTEN

compara estas heurísticas e outras conhecidas na literatura para buffer ilimitado e conclui que a heurística proposta por NAWAZ *et al.* (1983) possui um desempenho superior.

Neste trabalho apresenta-se uma heurística baseada em busca tabu para o critério de minimização do atraso total das tarefas. Escolheu-se esta metaheurística por seu excelente desempenho no problema de *flowshop* com buffer ilimitado. A inclusão de estratégias de diversificação, intensificação e redução de vizinhança é analisada. Uma comparação com uma adaptação do algoritmo NEH é realizada para avaliar o desempenho da heurística em problemas de média e grande dimensão. Para problemas de pequena dimensão as soluções da heurística são comparadas com as soluções provenientes do algoritmo *Branch-and-Bound* desenvolvido em RONCONI (1997) e RONCONI & ARMENTANO (2000).

Este artigo é organizado da seguinte forma. A próxima seção contém uma descrição do problema. A seção 3 apresenta os componentes da versão básica da busca tabu. A seção 4 mostra os testes computacionais com a versão básica, e a inclusão da diversificação, intensificação e redução da vizinhança. A última seção resume os principais resultados deste artigo.

2. Descrição do Problema

O ambiente abordado neste trabalho se caracteriza pelo processamento de tarefas em linha, ou seja n tarefas que são processadas por m máquinas na mesma ordem. Além disso, não existem espaços de armazenamento entre as máquinas, logo filas de tarefas esperando no sistema pelas próximas operações não são permitidas. O número de tarefas e máquinas, o tempo de processamento das operações, e a data de entrega das tarefas são conhecidos. Considera-se que todas as tarefas estão disponíveis para processamento no instante zero; os tempos de preparação das máquinas são independentes da seqüência e estão incluídos nos tempos de processamento; as máquinas estão continuamente

disponíveis, e as operações não podem ser interrompidas. Este problema pode ser representado pela notação $\alpha | \beta | \gamma$, largamente utilizada para descrever problemas de programação de tarefas, onde α representa a configuração das máquinas, β indica características especiais das tarefas e dos recursos, e γ define o critério de otimização utilizado. Logo, o problema tratado pode ser descrito por $Fm | blocking | \sum T_j$, ou seja, trata-se do problema de minimizar a soma dos atrasos no ambiente *flowshop* com m máquinas sem estoque intermediário. O número de alternativas para solucionar este problema é $(n!)$ e os programas correspondentes são chamados de programas de permutação.

Apresentam-se a seguir as fórmulas recursivas para avaliar o instante de término de cada tarefa em cada máquina (PINEDO, 1995). Seja $1, 2, \dots, i, \dots, n$ uma seqüência de tarefas, onde i representa a i -ésima tarefa, $M_1, M_2, \dots, M_k, \dots, M_m$ a seqüência ordenada de máquinas, p_{ik} o tempo de processamento da tarefa i na máquina k , e d_i a data de entrega da tarefa i . Seja D_{ik} o instante de partida da tarefa i da máquina k . Claramente, $D_{ik} \geq C_{ik}$ onde C_{ik} é o instante de término da tarefa i na máquina k . O instante de tempo no qual a tarefa i inicia o seu processamento na primeira máquina é representado por D_{i0} . Os instantes de partida de cada tarefa de cada máquina são dados pelas seguintes expressões:

$$D_{i0} = 0$$

$$D_{1k} = \sum_{q=1}^k p_{1q} \quad k = 1, \dots, m-1$$

$$D_{i0} = D_{i-1,1} \quad i = 2, \dots, n$$

$$D_{ik} = \max(D_{i,k-1} + p_{ik}, D_{i-1,k+1}) \\ i = 2, \dots, n \quad k = 1, \dots, m-1$$

$$D_{im} = D_{i,m-1} + p_{im} \quad i = 1, \dots, n$$

Na recursão acima, primeiro são calculados os instantes de partida da primeira tarefa em

todas as máquinas, em seguida os da segunda tarefa, até a última tarefa. Dado que $C_{im} = D_{im}$ o atraso total é dado por:

$$T = \sum_{i=1}^n \max(D_{im} - d_i, 0).$$

3. Heurística Proposta

A heurística consiste de uma busca em vizinhança orientada por busca tabu. Essa técnica é descrita por GLOVER & LAGUNA (1997) e objetiva guiar a busca na exploração do espaço de soluções de um problema além de um ótimo local. Inicialmente, utilizou-se a busca tabu somente com a memória de curto prazo. Denominou-se esta versão da heurística como a versão básica. Uma das principais características desta memória é armazenar os atributos de soluções já visitadas recentemente. Regras de proibição são associadas com estes atributos para proibir a ocorrência de alguns movimentos por um certo número de iterações. Esta característica tem o propósito de evitar ciclos e visitar diferentes regiões do espaço de soluções.

Adaptações do algoritmo NEH e algumas regras de despacho (também conhecidas como regras de liberação) foram avaliadas para gerar a solução inicial para a busca tabu. Os melhores resultados foram obtidos pela utilização da regra proposta por ARMENTANO & RONCONI (1999) como ponto de partida para o esquema de inserção utilizado pelo algoritmo NEH. A associação desta regra (LB) com o algoritmo NEH será denominada por NEH adaptado. Inicialmente, este algoritmo gera uma lista de prioridades entre as tarefas ordenado-as em ordem não decrescente da seguinte medida:

$$\text{LB:} \quad d_i - \sum_j p_{ij}.$$

As duas tarefas com a maior prioridade são selecionadas da lista, e as duas seqüências possíveis para estas são geradas. Estas são avaliadas de acordo com a medida de desempenho e a melhor seqüência parcial é escolhida. As

posições relativas destas duas tarefas não se modificam durante todo algoritmo. A seguir, a tarefa com a terceira maior prioridade é selecionada e três seqüências são geradas ao se posicionar esta tarefa no início, no meio, e no fim da seqüência gerada no passo anterior. A melhor seqüência parcial encontrada irá fixar as posições relativas destas três tarefas no resto do algoritmo. Este processo é repetido até que todas as tarefas sejam fixadas e a seqüência final encontrada. Maiores detalhes sobre o algoritmo NEH original para minimizar o *makespan* podem ser encontrados em NAVAZ *et al.* (1983).

Busca tabu requer a definição dos seus principais componentes, tais como o tipo de movimento que gera a vizinhança, o método de seleção da próxima solução, a característica (atributo) a ser armazenada de cada solução, a regra de proibição, o número de iterações durante as quais um atributo continua proibido (duração tabu), o critério de liberação da proibição tabu (critério de aspiração) e o critério de parada. Para determiná-los, diversos testes foram realizados. A combinação de componentes que apresentou os melhores resultados é descrita a seguir.

Em relação à estratégia de geração de vizinhança, observou-se que o movimento de inserção de uma tarefa *i* em todas as posições da seqüência obteve os melhores resultados, gerando uma vizinhança de tamanho $(n-1)^2$. Selecionou-se como a próxima solução a melhor solução encontrada na vizinhança. O atributo de cada solução armazenado foi a tarefa *i* escolhida para ser inserida, e como regra de proibição a não seleção desta tarefa para inserção.

A duração tabu é definida dinamicamente, ou seja, a cada 20 iterações gera-se um novo valor para a mesma com distribuição uniforme dentro do intervalo $[0,25n; 0,85n]$, e a cada iteração é feito um ajuste neste valor. Este ajuste é realizado de acordo com o valor da solução selecionada com relação à solução que gerou o movimento. Caso o valor da solução selecionada seja maior que o da solução atual, a duração tabu associada à solução selecionada é aumentada de

uma unidade. Caso o valor da solução selecionada seja menor que o da solução atual, a duração é diminuída.

Como critério de aspiração definiu-se que um movimento pode ser liberado da sua condição tabu caso a solução gerada possua um valor melhor que a melhor solução gerada até o momento. O critério de parada selecionado foi o tempo de CPU. A Tabela 1 apresenta o tempo permitido para cada tamanho de problema. Estes tempos foram estabelecidos experimentalmente utilizando-se o computador descrito na próxima seção.

Uma breve descrição da heurística é apresentada a seguir.

- *Passo 1:* Obtenha a solução inicial (x_0) e a sua correspondente soma dos atrasos ($c(x_0)$) pelo algoritmo NEH adaptado. Defina um vetor com *n* posições (Proíbe), que armazenará a duração tabu (inicialmente zero). Faça $k=0$, $x^*=x_0$ e $c^*=c(x_0)$ (x^* é a melhor solução encontrada até o momento e c^* é o valor da função objetivo para esta solução).
- *Passo 2:* Na vizinhança de x_k , encontre um movimento de inserção da tarefa *i* que minimize a função objetivo e que não esteja proibido (i.e., $Proíbe[i] \leq k$) ou que satisfaça o critério de aspiração. Caso nenhum movimento tenha sido selecionado eleja o movimento tabu que acarrete o menor valor da função objetivo. Execute o movimento selecionado e obtenha x_{k+1} .
- *Passo 3:* Se $c(x_{k+1}) < c^*$, faça $c^* = c(x_{k+1})$ e $x^* = x_{k+1}$. Faça $k = k + 1$.
- *Passo 4:* Atualize a duração tabu (*dt*) de forma dinâmica conforme descrito anteriormente. Faça $Proíbe[tarefa\ selecionada\ para\ inserção] = k + dt$.
- *Passo 5:* Se o tempo de execução é menor ou igual que o tempo de parada e c^* for diferente de zero então retorne ao Passo 2.

Na próxima seção, apresentam-se os resultados computacionais referentes à aplicação da heurística na versão básica, com diversificação, intensificação e redução de vizinhança.

Tabela 1 – Tempo permitido para cada dimensão.

Dimensão (nxm)	Tempo (segundos)
20 x 5	25
20 x 10	50
20 x 20	90
50 x 5	200
50 x 10	500
50 x 20	1000

4. Resultados Computacionais

4.1 Características da Implementação

Os programas computacionais foram escritos em linguagem C e os testes realizados em uma estação de trabalho SUN SPARC classic com 50MHz de *clock*, 48Mb RAM. Os dados de entrada foram gerados conforme sugerido por TAILLARD (1993). Para cada uma das seis dimensões do problema (Tabela 1) gerou-se dez diferentes matrizes de tempos de processamento e, para cada uma destas foram selecionados quatro cenários. O tempo de processamento é uniformemente distribuído entre 1 e 99. Cada cenário representa uma diferente geração de datas de entrega, que são uniformemente distribuídas entre $P(1 - T - R/2)$ e $P(1 - T + R/2)$ (POTTS & VAN WASSENHOVE 1982), onde T e R representam o fator de atraso e a faixa de dispersão, respectivamente, e P é um limitante inferior do *makespan* no ambiente *flowshop* com buffer ilimitado definido por:

$$P = \max \left\{ \max_{1 \leq k \leq m} \left\{ \sum_{i=1}^n p_{ik} + \min_i \sum_{q=1}^{k-1} p_{iq} + \min_i \sum_{q=k+1}^m p_{iq} \right\}, \max_i \sum_{k=1}^m p_{ik} \right\}.$$

A partir da variação de T e R foram obtidos os seguintes cenários:

- *cenário 1*: baixo fator de atraso (T=0,2) e pequena faixa de datas de entrega (R=0,6)
- *cenário 2*: baixo fator de atraso (T=0,2) e ampla faixa de datas de entrega (R=1,2)
- *cenário 3*: alto fator de atraso (T=0,4) e pequena faixa de datas de entrega (R=0,6)
- *cenário 4*: alto fator de atraso (T=0,4) e ampla faixa de datas de entrega (R=1,2)

Cada combinação de tamanho e cenário foi denominada por classe. A avaliação da heurística é realizada considerando a melhoria percentual obtida em relação à adaptação do algoritmo NEH:

$$\text{Melhoria} = \frac{T_{\text{NEH}} - T_{\text{tabu}}}{T_{\text{NEH}}} \times 100,$$

onde T_{NEH} e T_{tabu} são os valores do atraso total obtidos pelos algoritmos. No cálculo da média da porcentagem de melhoria não foram considerados os problemas onde a busca tabu ou o algoritmo NEH adaptado obtiveram a soma dos atrasos zero. Isto porque a melhoria da busca tabu seria de 100% independente do valor do algoritmo NEH adaptado. Os números em negrito na tabela indicam as classes nas quais a Busca Tabu encontrou somas dos atrasos com valor zero e o algoritmo NEH adaptado não.

4.2 Versão Básica

A Tabela 2 apresenta a porcentagem média de melhoria em cada classe composta de 10

Tabela 2 – Melhoria percentual da busca tabu em relação ao algoritmo NEH adaptado.

Dimensão (n x m)	Cenário				Média por dimensão
	1	2	3	4	
20x5	53,9	37,4	19,7	19,0	32,5
20x10	19,5	25,0	11,2	8,8	16,1
20x20	9,6	6,8	5,6	4,8	6,7
50x5	67,6	91,1	30,9	27,6	54,3
50x10	42,5	61,8	19,9	16,2	35,1
50x20	31,0	26,2	10,3	7,7	18,8
Média	37,4	41,4	16,3	14,0	

problemas. Neste teste a heurística na sua versão básica obteve uma média de melhoria sobre o algoritmo NEH adaptado de 27,3%. Pode ser notado que nos cenários 1 e 2, os quais possuem um menor fator de atraso, a heurística apresenta, em média, maiores melhorias que as apresentadas no cenário 3 e 4.

Observa-se também, considerando a última coluna da Tabela 2, que quanto maior o número de tarefas (n), mantendo-se o número de máquinas fixo (m), maior também é a média de melhoria por dimensão. Com n maior, maior o número de soluções possíveis, tornando a solução do algoritmo NEH adaptado, proveniente de uma enumeração parcial, mais pobre. Tal fato proporciona uma oportunidade para um maior ganho da busca tabu sobre este algoritmo.

4.3 Estratégias de Diversificação e Intensificação

Estas estratégias utilizam uma memória de longo prazo, *i.e.*, de acordo com a história da busca estas tentam identificar o quão frequentemente algumas regiões foram visitadas, características das soluções com os melhores valores da função objetivo, ou outras características que levem a uma nova forma de exploração do espaço de soluções. A diversificação possui o propósito de guiar a heurística a novas regiões, enquanto a intensificação objetiva selecionar

regiões onde a busca poderia ser mais aprofundada.

As estratégias de diversificação e intensificação, e a sua associação, propostas por ARMENTANO & RONCONI (1999) para o problema de *flowshop* com buffer ilimitado, tiveram suas aplicações avaliadas para este problema. Os melhores resultados foram encontrados com uma estratégia similar à utilizada por esses autores, ou seja, inicialmente aplica-se a diversificação, e ao fim da busca a intensificação. A diversificação utilizada neste trabalho é baseada na frequência de ocorrência de cada tarefa em cada posição; se uma tarefa ocupa com maior frequência uma certa posição, um movimento que a leve para esta posição será penalizado. A intensificação armazena as cinco melhores soluções encontradas durante a busca, suas respectivas memórias de curto prazo e também o melhor movimento selecionado a partir de cada uma destas. Posteriormente, retorna-se a estas soluções de forma a explorar suas vizinhanças de forma mais detalhada. De cada uma das soluções seleciona-se agora o segundo melhor movimento. Desta forma, espera-se que ao partir de uma destas boas soluções, a busca siga por um caminho diferente do anterior.

Esta versão utilizando a estratégia de diversificação e intensificação citada anteriormente foi avaliada no mesmo tempo computacional da versão básica. De acordo com GLOVER &

Tabela 3 – Melhoria percentual de busca tabu com diversificação e intensificação em relação ao algoritmo NEH adaptado.

Dimensão (n x m)	Cenários			
	1	2	3	4
20x5	55,5*	37,6*	19,7	19,0
20x10	20,8*	24,5	11,3*	8,9*
20x20	9,4	6,8	5,7*	4,8
50x5	67,2	91,1	31,0*	27,6
50x10	42,3	63,6*	20,5*	15,9
50x20	31,1*	25,9	10,0	7,8*
Média	37,7*	41,6*	16,4*	14,0

Tabela 4 – Comparações entre a versão básica e a versão com diversificação e intensificação.

Sucessos da diver. + inten.	Melhoria % média	Maior %	Sucessos da básica	Melhoria % média	Maior %	Empates
53	3,0	41,0	39	2,2	14,4	148

LAGUNA (1997), o uso destas estratégias não requer necessariamente um tempo longo de execução para que seus benefícios se tornem claros. Foram realizados diversos testes e os melhores resultados foram obtidos a partir das seguintes combinações para cada cenário:

- cenário 1: diversificação no início da busca e nos últimos 10% do tempo de execução aplicou-se intensificação;
- cenário 2: somente diversificação;
- cenários 3 e 4: somente intensificação nos últimos 25% do tempo de execução.

Os resultados desta versão encontram-se na Tabela 3. Os asteriscos indicam as classes de problemas onde a mesma superou a média de melhoria obtida pela versão básica.

A utilização destas estratégias acarretou um aumento de melhoria em 45,8% das classes de problema, manteve a melhoria em 25% das classes, e melhorou a média dos cenários 1, 2 e 3 em relação à versão básica.

Com o intuito de reforçar a análise anterior, apresenta-se na Tabela 4 a comparação desta

versão com a versão básica através do número de sucessos obtidos nos 240 problemas. Esta técnica de avaliação é a mesma adotada por WIDMER & HERTZ (1989) e MOCCELLIN (1995). Considera-se sucesso quando a implementação analisada obtém o menor valor da soma dos atrasos. A coluna denominada “Maior %” denota o maior ganho de uma implementação sobre a outra.

Nota-se que a versão com o melhor desempenho é a associação da diversificação com a intensificação que não só apresenta o maior número de sucessos em relação à versão básica, mas que também possui nestes problemas uma melhor média de melhoria. Esta versão apresenta um ganho no maior caso de 41,0% sobre a versão básica.

4.4 Redução da Vizinhança

Com o objetivo de reduzir o esforço computacional associado à avaliação da vizinhança de inserção, o comportamento da busca foi analisado

Tabela 5 – Melhoria percentual da busca tabu com redução de vizinhança em relação ao algoritmo NEH adaptado em um tempo 40% menor.

Dimensão (n x m)	Cenários			
	1	2	3	4
20x5	53,4	36,9	19,6	18,8
20x10	18,6	24,4	10,9	9,0
20x20	9,2	6,7	5,4	4,5
50x5	64,3	90,8	30,6	26,9
50x10	40,4	63,6	19,5	15,3
50x20	28,9	25,6	10,4	7,1
Média	35,8	41,3	16,1	13,6

para identificar uma redução no seu tamanho que não acarretasse uma perda significativa na qualidade das soluções. Observou-se que, na maioria dos problemas, as primeiras inserções eram realizadas em posições afastadas da posição atual da tarefa, enquanto que próximo de ótimos locais somente inserções próximas eram realizadas. Por tal motivo um corte cíclico na vizinhança foi gerado de tal forma a permitir inserções distantes no começo de um ciclo e de acordo com o avanço das iterações esta distância era reduzida. Quando se atinge n iterações, o processo de corte é reiniciado. A relação entre a distância máxima permitida a cada iteração e o número de iterações que apresentou melhor desempenho é dada por:

$$\text{Distância máxima} = - \text{número de iterações} + n$$

Quando a distância máxima permitida é zero, seleciona-se o movimento tabu com o melhor valor. Para avaliar a eficiência desta estratégia em tempos menores, a mesma foi avaliada com um tempo 40% menor que o apresentado na Tabela 1. Os resultados são apresentados na Tabela 5.

A Tabela 5 mostra que existe uma queda na qualidade das soluções em relação à versão básica, porém dada a redução de tempo obtida, esta queda pode ser considerada desprezível.

Logo, o uso da estratégia de redução de vizinhança é válido para um tempo 40% menor.

4.5 Comparações com o Algoritmo *Branch-and-Bound*

Com o objetivo de avaliar melhor a heurística proposta realizou-se uma comparação dos resultados obtidos pela mesma com os resultados ótimos obtidos pelo algoritmo *Branch-and-Bound* proposto em RONCONI (1997) e RONCONI & ARMENTANO (2000). Dada a dificuldade em se obter soluções ótimas para problemas com as dimensões do conjunto original de testes, gerou-se um novo conjunto com dimensões menores para a realização desta avaliação. Este conjunto possui 27 diferentes dimensões, para cada dimensão foram geradas cinco matrizes de processamento, e para cada uma destas 4 cenários, ou seja, 540 problemas. Os testes foram realizados em uma estação de trabalho SUN ULTRA 60 com 360 MHz de *clock* e 512 Mb de RAM. O tempo de execução do algoritmo *Branch-and-Bound* foi limitado em 1 hora. A tolerância utilizada foi de 0,5%. Como solução inicial utilizou-se a versão básica da heurística.

Neste novo conjunto-teste, a versão básica da heurística encontrou soluções ótimas em 530

problemas, a versão com redução de vizinhança e tempo 40% menor encontrou valores ótimos em 519 problemas, enquanto a versão utilizando diversificação e intensificação encontrou soluções ótimas em 529 problemas. O algoritmo NEH adaptado encontrou soluções ótimas somente em 49 problemas, ficando em média nos outros problemas a 311% do valor ótimo.

5. Conclusões

O objetivo deste trabalho foi aplicar a metaheurística busca tabu para minimizar o atraso total no ambiente *flowshop* com buffer zero. A inclusão de estratégias de diversificação, intensificação e redução de vizinhanças foram avaliadas. Comparando todas as versões implementadas neste trabalho conclui-se que a versão básica produz excelentes resultados, além de ser mais facilmente implementada. Deve ser destacado que, nos 540 problemas de dimensão reduzida testados, esta versão superou o algoritmo NEH adaptado em 490. Além disso, esta versão obteve a solução ótima em 98% destes problemas.

A incorporação da diversificação e intensificação apresentou a capacidade de melhorar o desempenho da busca tabu em alguns problemas com o mesmo tempo de execução. Embora esta estratégia tenha obtido resultados positivos, não deve ser desconsiderado que para estabelecer os parâmetros da mesma foram necessários inúmeros testes. Dado que as melhorias médias foram

pequenas, é questionável a aplicação de tais estratégias em problemas com características similares ao *flowshop*, ou seja, problemas com acesso a vizinhanças ricas como a gerada pelo movimento de inserção. Acredita-se que possivelmente estas estratégias possuam um desempenho mais relevante em problemas com vizinhanças mais restritas.

A estratégia de redução da vizinhança é simples de ser implementada além de apresentar bons resultados em um tempo menor. Os resultados mostraram que esta versão é uma boa opção.

Futuras pesquisas poderiam ser realizadas no sentido de estender as idéias apresentadas neste artigo para o problema de *flowshop* com atrasos ponderados e *flowshop* com buffer limitado. Outra abordagem futura interessante seria examinar os resultados obtidos por uma análise estatística, mais especificamente empregando-se técnicas de delineamento de experimentos, com o objetivo de observar a influência dos parâmetros utilizados na busca.

Agradecimentos

Os autores são gratos aos revisores anônimos pelos seus úteis comentários e sugestões. Esta pesquisa teve o apoio financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e da Fundação Coordenação de Aperfeiçoamento de Pessoal de Ensino Superior (CAPES).

Referências Bibliográficas

- ADENSO_DÍAZ, B.:** "Restricted Neighborhood in the Tabu Search for the Flowshop Problem". *European Journal of Operational Research*, v.62, p.27-37, 1992.
- ARMENTANO V.A. & RONCONI, D.P.:** "Tabu Search for Total Tardiness Minimization in Flowshop Scheduling Problems". *Computers and Operations Research*, v.26, p.219-235, 1999.
- DU, J. & LEUNG, JY-T.:** "Minimizing total tardiness on one machine is NP-hard". *Mathematics of Operations Research*, v.15, p.483-495, 1990.
- GILMORE, P.C. & GOMORY, R.E.:** "Sequencing a One State Variable Machine: A Solvable Case of the Traveling Salesman Problem". *Operational Research*, v.12, p.655-665, 1964.
- GLOVER, F.W. & LAGUNA, M.:** *Tabu Search*, Kluwer Academic Publishers, Boston, 1997.
- GRABOWSKI, J.; SKUBALSKA, E. & SMUTNICKI, C.:** "On Flow Shop Scheduling with Release and Due Dates to Minimize Maximum Lateness". *Journal of the Operational Research Society*, v.34, p.615-620, 1983.

- HALL, N.G. & SRISKANDARAJAH, C.:** “A Survey of Machine Scheduling Problems with Blocking and No-wait in Process”. *Operations Research*, v.44, p.510-525, 1996.
- KIM, Y-D.:** “Heuristics for Flowshop Scheduling Problems Minimizing Mean Tardiness”. *Journal of the Operational Research Society*, v.44, p.19-28, 1993.
- KIM, Y-D.:** “Minimizing Mean Tardiness on Flowshops”. *European Journal of Operational Research*, v.85, p.541-555, 1995.
- KOULAMAS, C.:** “The Total Tardiness Problem: Review and Extensions”. *Operations Research*, v.42, p.1025-1041, 1994.
- LEISTEN, R.:** “Flowshop Sequencing Problems with Limited Buffer Storage”. *International Journal of Production Research*, v.28, p.2085-2100, 1990.
- LEVNER, E.M.:** “Optimal Planning of Parts Machining on a Number of Machines”. *Automation and Remote Control*, v.12, p.1972-1978, 1969.
- MOCCELLIN, J.V.:** “A New Heuristic Method for the Permutation Flow Shop Scheduling Problem”. *Journal of the Operational Research Society*, v.46, p.883-886, 1995.
- MCCORMICK, S.T.; PINEDO, M.L.; SHENKER, S. & WOLF, B.:** “Sequencing in an Assembly Line with Blocking to Minimize Cycle Time”. *Operations Research*, v.37, p.925-936, 1989.
- NAWAZ, M.; ENSCORE, E.E. & HAM, I.:** “A Heuristic Algorithm for the m-Machine, n-Job Flow-shop Sequencing Problem”. *OMEGA*, v.11, p.91-95, 1983.
- NOWICKI, E. & SMUTNICKI, C.:** “A Fast Tabu Search Algorithm for the Flow Shop Problem”. *European Journal of Operational Research*, v.91, p.160-175, 1996.
- OW, P.S.:** “Focused Scheduling in Proportional Flowshops”. *Management Science*, v.31, p.852-869, 1985.
- PINEDO, M.:** *Scheduling: theory, algorithms, and systems*, Prentice-Hall, New Jersey, 1995.
- POTTS, C.N. & VAN WASSENHOVE, L.N.:** “A Decomposition Algorithm for the Single Machine Total Tardiness Problem”. *Operations Research Letters*, v.1, p.177-181, 1982.
- RAMAN, N.:** “Minimum Tardiness Scheduling in Flow Shops: Construction and Evaluation of Alternative Solution Approaches”. *Journal of Operations Management*, v.12, p.131-151, 1995.
- REDDI, S.S. & RAMAMOORTHY, C.V.:** “On the Flowshop Sequencing Problem with No Wait in Process”. *Operational Research Quarterly*, v.23, p.323-330, 1972.
- REEVES, C.R.:** “Improving the Efficiency of Tabu Search for Machine Sequencing Problems”. *Journal of the Operational Research Society*, v.44, p.375-382, 1993.
- RONCONI, D.P.:** *Uma Contribuição para o Estudo do Problema de Flowshop para Minimizar a Soma dos Atrasos*. Campinas. Dissertação (Doutorado) – Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, 1997.
- RONCONI, D.P. & ARMENTANO, V.A.:** “Lower Bounding Schemes for Flowshops with Blocking In-Process”, submetido para publicação, 2000.
- SEN, T. & GUPTA, S.K.:** “A State-of-Art Survey of Static Scheduling Research Involving Due Date”. *OMEGA*, v.12, p.63-76, 1984.
- SUHAMI, I. & MAH, R.S.H.:** “An Implicit Enumeration Scheme for the Flowshop Problem with No Intermediate Storage”. *Computers and Chemical Engineering*, v.5, p.83-91, 1981.
- TAILLARD, E.:** “Some efficient heuristic methods for the flow shop sequencing problem”. *European Journal of Operational Research*, v.47, p.65-74, 1990.
- TAILLARD, E.:** “Benchmarks for basic scheduling problems”. *European Journal of Operational Research*, v.64, p.278-285, 1993.
- TOWNSEND, W.:** “Sequencing n Jobs on m Machines to Minimize Maximum Tardiness: a Branch-and-Bound Solution”. *Management Science*, v.23, p.1016-1019, 1977.
- WIDMER, M. & HERTZ, A.:** “A New Heuristic Method for the Flow Shop Sequencing Problem”. *European Journal of Operational Research*, v.41, p.186-193, 1989.

TOTAL TARDINESS MINIMIZATION IN A FLOWSHOP WITH BLOCKING IN-PROCESS USING TABU SEARCH***Abstract***

Surveys of production scheduling show that meeting customer due dates is a critical concern in most manufacturing systems. While there is considerable research to minimize the makespan in a flowshop with no storage constraints, very little work has dealt with job total tardiness criterion in a flowshop with blocking in-process. Scheduling problems with blocking arise in serial manufacturing processes where no intermediate buffer storage is available. In this paper we investigate the application of tabu search to this problem in order to explore the solution space. Additional strategies are evaluated. Computational tests are presented and comparisons with the NEH algorithm and with a Branch-and-Bound algorithm are made.

Key words: scheduling, flowshop with blocking, tabu search.