Associação Brasileira de Engenharia de Produção - ABEPRO
Universidade Federal de Santa Catarina - UFSC
www.producaoonline.org.br
ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

TENDÊNCIAS DE APLICAÇÕES DA OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS NA PROGRAMAÇÃO DE *JOB-SHOPS*

APPLICATION TRENDS OF ANT COLONY OPTIMIZATION IN JOB-SHOPS SCHEDULING

Felipe Fonseca Tavares de Freitas

Pontifícia Universidade Católica do Paraná – PUCPR
Mestrando em Engenharia de Produção e Sistemas
Programa de Pós-graduação em Engenharia de Produção e Sistemas
Rua Imaculada Conceição, 1155. Parque Tecnológico – Bloco 3 – 2º Andar
Prado Velho – Curitiba – PR, CEP: 80215-901, (41) 3271-2579
fftfreitas@gmail.com

Guilherme Ernani Vieira

Pontifícia Universidade Católica do Paraná – PUCPR
Professor Titular
Programa de Pós-graduação em Engenharia de Produção e Sistemas
Rua Imaculada Conceição, 1155. Parque Tecnológico – Bloco 3 – 2º Andar
Prado Velho – Curitiba – PR, CEP: 80215-901, (41) 3271-2579
gui.vieira@pucpr.br

RESUMO

A programação da produção revela-se como uma atividade que, se bem planejada, otimizada e controlada, gera grandes vantagens competitivas e duradouras para a empresa. Um dos problemas mais complexos de programação da produção ocorre em sistemas do tipo *job-shop* – os quais envolvem otimização combinatória, cuja resolução em tempo computacional aceitável é quase sempre improvável (NP-*hard*). Neste contexto, vários métodos de otimização têm sido pesquisados e desenvolvidos nas últimas décadas, almejando-se planos de produção cada vez melhores, sob tempos de execução computacional gradativamente menores e viáveis para a indústria. Em especial, destaca-se o uso das técnicas de inteligência coletiva que, mimetizando fenômenos biológicos e sociais da natureza, vêm obtendo bons resultados quando aplicadas a problemas do tipo *job shop scheduling* (JSS). Revisando-se a teoria referente à metaheurística de Otimização por Colônia de Formigas (ou *Ant Colony Optimization* - ACO) e suas aplicações em problemas de JSS, este artigo identifica e explica

ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

as principais tendências de pesquisa nessa área, tanto a nível mundial quanto nacional. Como resultados deste estudo, pode-se vislumbrar a hibridização entre ACO e outros algoritmos de otimização e o tratamento de cenários de JSS cada vez mais complexos como as tendências mais relevantes dos trabalhos envolvendo ACO e problemas de programação de *job-shops*.
Palavras-chave: Otimização combinatória. *Job shop scheduling*. Otimização por colônia de formigas. Tendências de pesquisa.

ABSTRACT

The production scheduling function, if well planned, optimized, and controlled, creates an important competitive and lasting advantage to a company. One of the most complex production scheduling problems occurs in job-shop environments – which deal with combinatory optimization and acceptable computer execution time is usually improbable (NP-hard). In this context, several optimization methods have been researched and developed in these last decades aiming at finding better production plans, under computer execution times gradually smaller and hopefully viable for the industry. In particular, swarm intelligence-based techniques have gained greater attention, which by mimicking biological and social nature phenomena have obtained good results when applied to production job shop scheduling problems (JSS). Reviewing the theory about a metaheuristic called Ant Colony Optimization (ACO) and its use in JSS problems, this paper identifies and explains the most important research trends in this area, both in the world and nationally. As a result of this study, one can see the hybrid use of ACO and other optimization algorithms and the solution of more complex JSS scenarios as the most relevant tendencies of works dealing with ACO and job shop scheduling problems.
Key-words: Combinatory optimization. Job shop scheduling. Ant colony optimization. Research trends.

1 INTRODUÇÃO

A competitividade a nível mundial tem afetado as organizações de várias maneiras, sob múltiplos pontos de vista. Nesse contexto, Kunnathur *et al.* (2004) e Heinonen e Pettersson (2007) esclarecem que programas de produção devidamente dimensionados são capazes de aproveitar de modo adequado a capacidade produtiva dos recursos produtivos disponíveis, configurando notável vantagem competitiva a partir da redução de custos de produção por meio de menores níveis de estoques e maiores índices de produtividade.

Assim, dentre os níveis de planejamento da produção identificados em uma empresa, a programação de chão de fábrica ou de curto prazo (*shop scheduling*) destaca-se, quando observados os crescentes esforços de flexibilização e automação exigidos pela dinâmica

ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

produtiva contemporânea, que tornam as atividades de programação da produção cada vez mais complexas (KATHAWALA e ALLEN, 1993).

Segundo Cox *et al.* (1992), *shop scheduling* refere-se à colocação de datas de início e término a operações ou grupos de operações, para indicar quando essas devem ser iniciadas, objetivando-se a data final como restrição de término daquelas. Nesse âmbito da programação da produção, destaca-se o *job shop scheduling problem* (JSSP), um problema de otimização combinatória que se refere à alocação de um grupo de operações aos recursos produtivos de uma fábrica (máquinas, linhas de produção, pessoal/mão-de-obra, ferramentas, etc.), tendo-se usualmente como principal objetivo a atribuição do maior número plausível de tarefas (operações) aos recursos disponíveis, no menor tempo de processamento possível.

Sob essa perspectiva, Dorigo e Stützle (2004) comentam que problemas de otimização combinatória são interessantes porque, mesmo sendo usualmente fáceis de interpretar e declarar, são geralmente difíceis de resolver. Nesse contexto, destacam-se os problemas definidos como NP-*hard*, ou seja, aqueles que não podem ser deterministicamente resolvidos em tempo computacional polinomial. Dentre os muitos problemas dessa ordem, ressaltam-se os problemas de programação da produção, especialmente o JSSP, principalmente pelo fato do mesmo representar um dos mais difíceis problemas de otimização combinatória abordados pela teoria clássica de programação (GAREY e JOHNSON, 1979).

Para solucionar tais problemas, são empregados, segundo Dorigo e Blum (2005), dois grupos de algoritmos, quais sejam: completos e aproximados. Enquanto os primeiros garantem soluções ótimas em tempo hábil, os aproximados abrem mão da otimalidade das soluções em prol de resultados satisfatórios, que possam ser obtidos em tempo computacional viável.

Nesse contexto, uma série de métodos de otimização foi experimentada nas últimas décadas, desde técnicas de programação matemática, tais como programação linear e *branch-and-bound* (Land e Doig, 1960), até sistemas especialistas (Alexander, 1987; Kusiak e Chen, 1988; Jain e Meeran, 1999), algoritmos genéticos (Goldberg, 1989; Ling, 2003), busca *tabu* (Taillard, 1989; Dell Amico e Trubian, 1993; Barnes e Chambers, 1996) e *simulated annealing* (Laarhoven *et al.*, 1992; Sadeh e Nakakuki, 1996; Aydin e Fogarty, 2004).

ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

No entanto, Dorigo *et al.* (1991) apresentam um novo método aproximado de Otimização por Colônia de Formigas (ou *Ant Colony Optimization* - ACO), tratando-se de uma técnica enquadrada em um ramo específico da inteligência artificial (inteligência coletiva ou *swarm intelligence*), inspirada no comportamento social que as formigas apresentam ao buscarem por fontes de alimento para seus ninhos (HUANG e LIAO, 2008). Ao longo de suas jornadas, os animais depositam uma substância química denominada feromônio, capaz de orientar as demais formigas na escolha do menor caminho do ninho até a fonte de alimento identificada. Quanto mais feromônio depositado, menor é o caminho e, conseqüentemente, mais atraídas para essa trajetória serão as formigas vindouras.

Nesse contexto, este artigo visa fazer uma análise sobre as principais tendências do método de otimização por colônia de formigas em problemas de programação da produção de *job-shops*. O restante do artigo encontra-se organizado da seguinte forma: as Seções 2 e 3 introduzem alguns conceitos e definições sobre problemas de otimização combinatória e JSSP; a Seção 4 trata de alguns aspectos teóricos necessários à compreensão da técnica ACO; a Seção 5 apresenta e discute alguns trabalhos acadêmicos relacionados às principais tendências e oportunidades de pesquisa envolvendo ACO e JSSP; e a Seção 6 traz algumas considerações finais e propostas para estudos futuros.

2 PROBLEMAS DE OTIMIZAÇÃO COMBINATÓRIA

Um problema de otimização combinatória pode ser caracterizado a partir da necessidade de se atribuir valores observados em um dado espaço de busca a determinadas variáveis discretas, de modo que uma solução ótima seja identificada em relação a uma função-objetivo conhecida (DORIGO e STÜTZLE, 2004). Papadimitriou e Steiglitz (1982), por sua vez, definem um problema dessa natureza como um conjunto de objetos e uma função-objetivo para a qual, atribuindo-se valores de custo positivos para cada elemento, almeja-se construir soluções capazes de encontrar o objeto de menor valor de custo positivo.

Nesse sentido, nota-se que problemas de otimização despertam tanto o interesse científico quanto o profissional, por se enquadrarem em uma série de eventos observados diariamente por empresas das mais variadas naturezas possíveis. Assim, pode-se vislumbrar uma gama de exemplos que sinalizam a relevância de tais problemas sob as perspectivas

Revista Produção On Line
Associação Brasileira de Engenharia de Produção - ABEPRO
Universidade Federal de Santa Catarina - UFSC
www.producaoonline.org.br
ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

teórica e prática, tais como manipulação de itens em estoque, roteirização de veículos, programação de transporte entre células de manufatura, otimização de operações de máquinas em manufatura, fabricação de *chips*, programação da produção, identificação das menores trajetórias (problema do caixeiro viajante), dentre outros (GOLDBARG e LUNA, 2005).

Dorigo e Stützle (2004) e Dorigo e Blum (2005) apresentam uma definição mais formal, que representa problemas de otimização combinatória conforme a notação $\mathcal{P} = (\mathcal{S}, \Omega, f)$, onde:

- $\mathcal{S}$: espaço de busca finito, composto pelas soluções candidatas;
- Ω: conjunto de restrições impostas ao espaço de soluções; e
- f: uma função objetivo.

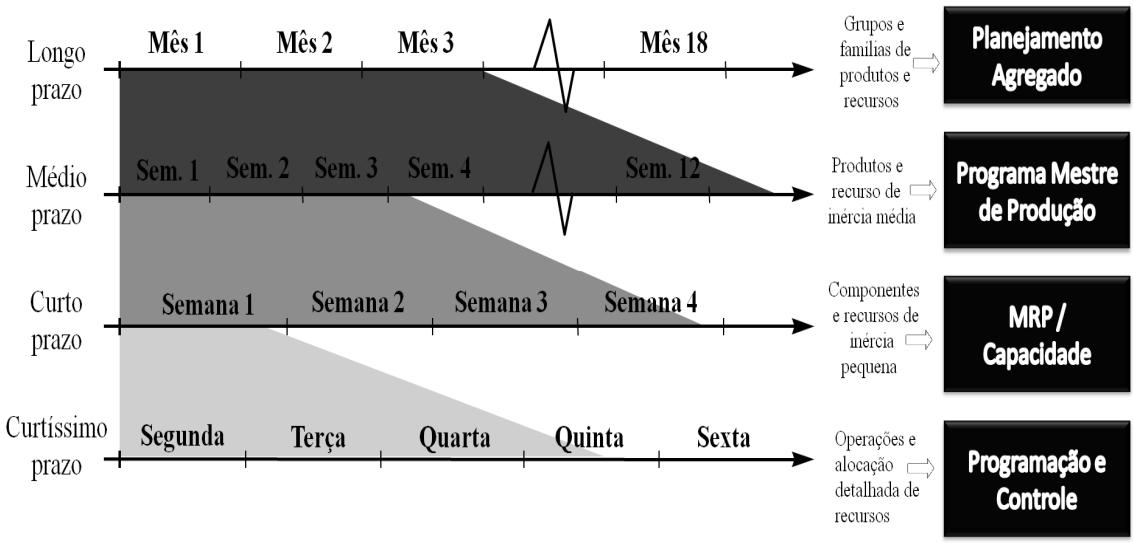
O espaço de busca $\mathcal{S}$ é composto por n variáveis de decisão discretas X_i, sendo os valores do domínio de soluções $v_i^j \in D_i = \left\{v_i^1, \ldots, v_i^{|D_i|}\right\}$, $i = 1, \ldots, n$. Destarte, uma solução $s \in \mathcal{S}$ é caracterizada como viável quando todas as variáveis de decisão do espaço de busca possuem valores do domínio de solução atribuídos, ou seja, quando $X_i = v_i^j$, sendo $i = 1, \ldots, n$ e $j = 1, \ldots, |D_i|$, respeitando-se as restrições expressas em Ω. O objetivo dos métodos de otimização, pois, é encontrar uma solução viável global s^*, tal que: para problemas de minimização, $f(s^*) \leq f(s)$, para toda solução viável $s \in \mathcal{S}$; para problemas de maximização, $f(s^*) \geq f(s)$, para toda solução viável $s \in \mathcal{S}$.

Para resolver tais problemas, são empregados, basicamente, dois tipos de métodos: determinísticos e probabilísticos. Os determinísticos abrangem, fundamentalmente, técnicas alicerçadas em modificações de trajetórias e imposição de penalidades, como meios de se escapar de mínimos locais; enquanto os métodos estocásticos envolvem, essencialmente, decisões probabilísticas de quando a busca deve ou não partir da vizinhança de mínimos (PARSOPOULOS e VRAHATIS, 2002). Em se tratando de problemas do tipo NP-*hard*, como é o caso do JSSP, não existem métodos capazes de resolvê-los em tempo polinomial. Logo, houve um estímulo cada vez maior às pesquisas voltadas para o desenvolvimento de algoritmos probabilísticos, circunstância em que a técnica ACO ganha destaque.

3 *JOB SHOP SCHEDULING*

Associação Brasileira de Engenharia de Produção - ABEPRO
Universidade Federal de Santa Catarina - UFSC
www.producaoonline.org.br
ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

A programação da produção, sujeita hierarquicamente ao plano mestre de produção, é definida por Morton e Pentico (1993) e Pinedo (1995) como o processo de alocação de certos recursos limitados a determinadas tarefas por um dado curto período de tempo (horas e/ou dias), de forma a se produzir os *outputs* previstos no prazo planejado, obedecendo-se, concomitantemente, a um conjunto de restrições de tempo, de relações de precedência e de capacidades produtivas. O posicionamento da programação da produção nos níveis hierárquicos de planejamento e controle da produção pode ser vislumbrado por meio da Figura 1.



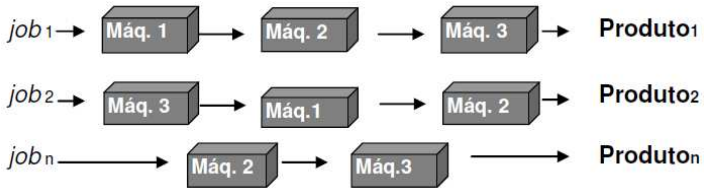
Fonte: Adaptado de Corrêa e Corrêa (2006).

Figura 1: Níveis hierárquicos de Planejamento e Controle da Produção (PCP).

Nesse contexto, Ross *et al.* (2005) destacam uma série de problemas de programação da produção, definidos a partir da relação entre o número de máquinas (recursos), ordens de produção (*jobs*) e operações, e dos diferentes modos como as últimas são programadas. Dentre os cenários identificados, destaca-se o *job shop scheduling* (JSS), interpretado por Fattahi *et al.*(2007) como um problema de programação da produção cuja preocupação é alocar um grupo de ordens de produção com seqüências de operação pré-ordenadas em ambientes dotados de múltiplos recursos produtivos, sendo que cada ordem apresenta um conjunto de restrições tecnológicas que descrevem diferentes roteiros produtivos fixos e previamente conhecidos.

Associação Brasileira de Engenharia de Produção - ABEPRO
Universidade Federal de Santa Catarina - UFSC
www.producaoonline.org.br
ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

Em termos formais, um *job shop* descreve um sistema produtivo composto por *n jobs* e *m* máquinas, sendo cada *job* integrado por um grupo de operações O_{ij} = (i = 1, 2, 3, ..., m; j = 1, 2, 3, ..., n). Cada *job*, por sua vez, apresenta uma seqüência própria de execução de operações nos recursos produtivos disponíveis, conforme ilustra a Figura 2.



Fonte: Bustamante (2007).

Figura 2: Exemplo de cenário produtivo do tipo *job shop scheduling* – JSS.

Sob uma ótica matemática, um cenário clássico de JSS pode ser caracterizado por meio de (ZHANG *et al*., 2006; HEINONEN e PETTERSSON, 2007; ESSAFI *et al*., 2008; XING *et al*., 2009):

- Um conjunto de *n jobs*, sendo $J = \{J_1, J_2, \ldots, J_n\}$;
- Um conjunto de *m* recursos produtivos (máquinas), tal que $M = \{M_1, M_2, \ldots, M_m\}$;
- Cada *job j* é composto por uma seqüência pré-ordenada de $u_{ij} \leq m$ operações, de modo que u_{ij} representa a execução do *job j* na máquina *i*, tal que o conjunto de operações de um sistema $O = \{u_{ij} | i \in [1, m], j \in [1, n]\}$;
- Se uma relação de precedência $u_{ej} \rightarrow u_{fj}$ existir, significa que as operações u_{ej} e u_{fj} pertencem ao mesmo *job j*, sendo que não há máquina *h* que não seja *e* ou *f*, tal que a relação $u_{ej} \rightarrow u_{hj}$ ou $u_{hj} \rightarrow u_{fj}$ exista. Assim, uma vez identificada a relação $u_{ej} \rightarrow u_{fj}$, pode-se afirmar que u_{fj} é uma operação que imediatamente sucede u_{ej};
- Cada operação u_{ij} apresenta um tempo de processamento não-negativo $p_{ij} > 0$, sendo que, uma vez iniciada sua execução na máquina *i*, a mesma não pode ser interrompida e não pode haver processamento simultâneo de outra operação. Logo, se $u_{ej} \rightarrow u_{fj}$, então a operação u_{fj} não pode ser iniciada até que u_{ej} já se encontre finalizada; e
- De maneira complementar, somente um *job* pode ser processado, a cada unidade de tempo, em uma determinada máquina *i*. Logo, iniciado o processamento do *job j* na

Revista Produção On Line
Associação Brasileira de Engenharia de Produção - ABEPRO
Universidade Federal de Santa Catarina - UFSC
www.producaoonline.org.br
ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

máquina *i*, não pode haver um *job k* também em execução na máquina *i*, tal que k ≠ *j*.

No que diz respeito aos objetivos de se resolver um JSSP, vários podem ser os critérios de desempenho adotados. Ross *et al*. (2005) citam dez parâmetros de desempenho usualmente aplicados em trabalhos científicos, mas esclarecem que esses não são os únicos, existindo uma série de critérios alternativos. Dentre os mais usados em pesquisas científicas, destaca-se o *makespan,* que representa o tempo total transcorrido entre a primeira e a última operações programadas de um sistema. Esse indicador é definido a partir do valor $C_{ij} = C_{kj} + p_{ij}$, voltado à operação u_{ij} e segundo a relação de precedência $u_{kj} \rightarrow u_{ij}$. Destarte, um JSSP que tenha como parâmetro de desempenho o *makespan* pode ser matematicamente formulado da seguinte maneira (ZHANG *et al*., 2006; ARTIGUES e FEILLET, 2008; ARTIGUES *et al*., 2009):

$$Minimizar \quad C_{max} = \max_{todo\ u_{ij} \in O} (C_{ij}) = \max_{u_{kj} \rightarrow u_{ij}} (C_{kj} + p_{ij}) \qquad (1)$$

Sujeito a:

$$C_{ij} \geq C_{kj} + p_{ij}\,, \quad \forall\ u_{kj} \rightarrow u_{ij}\,; \quad i = 1,2,\dots,m \quad e \quad j = 1,2,\dots,n \qquad (2)$$

$$C_{ij} \geq C_{ik} + p_{ij}\,, \quad \forall\ u_{ik} \rightarrow u_{ij}\,; \quad i = 1,2,\dots,m \quad e \quad j = 1,2,\dots,n \qquad (3)$$

$$p_{ij} \geq 0\,, \quad \forall\ u_{ij} \in O \qquad (4)$$

$$t_{ij} \geq 0\,, \quad \forall\ u_{ij} \in O \qquad (5)$$

A função-objetivo é definida a partir da minimização do *makespan* (C_{max}), assim como o tempo de conclusão ou finalização dos *jobs* (1). As restrições de precedência operacional e de processamento são estabelecidas em (2) e (3), respectivamente. Elas declaram não somente que cada *job* possui uma determinada seqüência de operações a ser obedecida, como também que somente um *job* pode ser processado, a cada unidade de tempo, em um mesmo recurso produtivo. As restrições (4) e (5), por sua vez, apenas indicam a não-negatividade dos tempos de processamento (p_{ij}) e de início (t_{ij}) das operações a serem programadas.

Nesse âmbito, JSSPs representam um dos principais objetos de estudo na área de desenvolvimento de métodos de resolução de problemas de otimização combinatória, sendo

ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

expressiva a quantidade de trabalhos acadêmicos que abordam tais eventos (JAIN e MEERAN, 1999; ROSS *et al*., 2005).

4 OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS

A Otimização por Colônia de Formigas (ou *Ant Colony Optimization* – ACO) pode ser interpretada como uma tecnologia de otimização global inspirada no comportamento natural de colônias de formigas, quando empenhadas em encontrar novas fontes de alimentos para seus ninhos. Em termos computacionais, trata-se da modelagem do comportamento de formigas "artificiais", de modo que essas, cooperando entre si, procurem pelas melhores soluções viáveis (melhores fontes de comida) para um dado problema, observadas determinadas restrições (limitações ou peculiaridades impostas à movimentação dos agentes) integradas ao espaço de busca considerado (a pluralidade de caminhos que podem ser explorados pelas formigas) (YING e LIAO, 2004; HUANG e LIAO, 2008; YAGMAHAN e YENISEY, 2008).

Ainda de acordo com os autores supracitados e Beckers *et al.* (1992), as formigas são animais capazes de identificar os menores caminhos entre seus ninhos e as fontes de alimento mais próximas, sem necessitar de quaisquer orientações visuais. Na verdade, a única forma de comunicação existente entre os insetos ocorre por meio de uma essência aromática depositada pelos mesmos durante o processo de busca por novas e boas fontes de comida. A substância liberada, conhecida por feromônio, é percebida pelas demais formigas do ninho que, ao iniciarem suas buscas, terão suas trajetórias orientadas pelo nível de concentração da química secretada.

Quanto maior a quantidade de feromônio depositada em um determinado caminho, mais estimuladas a percorrerem tal curso serão as formigas vindouras, ou seja, maior será a probabilidade de escolha daquele caminho. À medida que as formigas identificam os menores caminhos que levam às melhores fontes de alimento, os percursos do ninho até a fonte e vice-versa ocorrem, naturalmente, de forma mais rápida, uma vez que se trata do menor caminho. Logo, mais rapidamente será acumulada uma quantidade considerável de feromônio, fazendo com que a probabilidade de que futuras formigas escolham tais direções seja maior do que aquelas referentes a caminhos alternativos (ZHANG *et al.*, 2006).

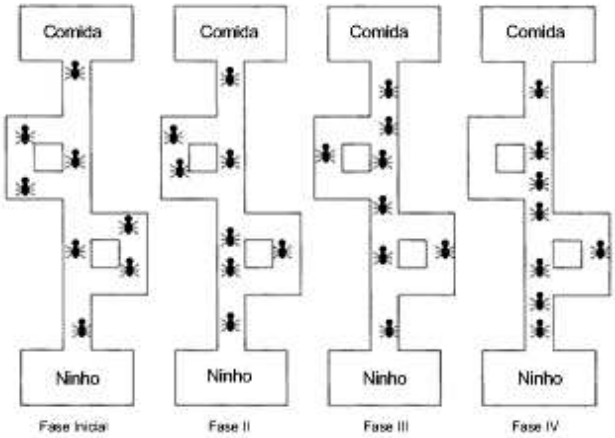
Associação Brasileira de Engenharia de Produção - ABEPRO
Universidade Federal de Santa Catarina - UFSC
www.producaoonline.org.br
ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

4.1 INSPIRAÇÃO BIOLÓGICA DO MÉTODO

Quando as formigas movimentam-se em busca de novas fontes de alimentos, ocorre um processo de auto-organização que, de acordo com Vittori *et al.* (2006), é regido por dois tipos de *feedback*, ambos relacionados à concentração de feromônio secretado pelos agentes: um positivo (intensificação da quantidade de feromônio depositada em um determinado caminho); e um negativo (evaporação gradativa da concentração de feromônio em certas trajetórias). Com base no equilíbrio entre essas duas informações, a colônia torna-se apta a selecionar, progressivamente, as melhores fontes de comida, assim como os menores caminhos para alcançá-las (GOSS *et al.*, 1989; BECKERS *et al.*, 1993).

De modo geral, o processo de busca de uma formiga por novos recursos alimentícios envolve, basicamente, uma constante tomada de decisão, seguida por futura ratificação ou não da escolha realizada anteriormente. Segundo Vittori *et al.* (2006), ao longo de sua caminhada, um formiga depara-se freqüentemente com "bifurcações". Tais bifurcações, naturalmente, exigem que o inseto tome uma decisão sobre qual caminho seguir. Inicialmente, tal escolha é feita de modo aleatório, pois ainda não há reconhecimento químico do espaço de busca.

Com o passar do tempo e realizadas algumas rotinas de busca a fontes de alimentos, passam a existir diferenças de concentração entre as quantidades atualizadas de feromônio entre as trajetórias componentes do espaço de busca. Logo, a tomada de decisão deixa de ser randômica e torna-se guiada probabilisticamente pelos níveis de feromônio percebidos pelas formigas que percorrem as bifurcações do espaço de busca (JOHNSON e ROSSI, 2006).



Fonte: Adaptado de McMullen (2001).

Figura 3: Comportamento das formigas em busca de novas fontes de alimento.

Associação Brasileira de Engenharia de Produção - ABEPRO
Universidade Federal de Santa Catarina - UFSC
www.producaoonline.org.br
ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

Um modelo visual do comportamento exploratório das formigas pode ser vislumbrado por meio da Figura 3. A fase inicial e a fase II descrevem os primeiros esforços de busca por alimentos, quando ainda não existe orientação via feromônio. A fase III, por sua vez, já revela como acontece a intensificação (*feedback* positivo) da melhor trajetória, uma vez que um número maior de formigas percorre um determinado caminho e, conseqüentemente, maior é o depósito de feromônio nessa região de busca. A fase IV, finalmente, além de exibir a intensificação do menor curso, exemplifica a evaporação (*feedback* negativo) dos caminhos alternativos, que não receberam acréscimos de feromônio. Com o tempo, as formigas deixam de ser sensibilizadas e não se movimentam mais por essas trajetórias.

4.2 ACO SOB UMA PERSPECTIVE FORMAL

Tomando-se como referência os trabalhos de Blum e Sampels (2004), Dorigo e Blum (2005) e Blum (2005a), pode-se dizer que a ACO atua, fundamentalmente, em cinco etapas, sendo essas: inicialização do modelo de feromônio e parâmetros de configuração; construção de soluções viáveis; realização de procedimentos de busca local; atualização do modelo de feromônio; e implantação de ações complementares. O pseudo-código exibido na Figura 4 esclarece a dinâmica geral de um algoritmo de ACO.

Início
Passo I – *Inicialização do modelo de feromônio e dos parâmetros de configuração.*
Enquanto (critério de parada não for alcançado) fazer:
Passo II – *construir uma solução viável;*
Passo III – *melhorar a solução por meio de procedimentos de busca local (opcional);*
Passo IV – *atualizar o modelo de feromônio (intensificação e evaporação);*
Passo V – *desempenhar ações complementares.*
Retornar a melhor solução encontrada.
Fim

Figura 4: Pseudo-código geral do método ACO.

A inicialização dos valores de feromônio nada mais significa do que a atribuição de um valor baixo e constante de feromônio ($\tau_0 \geq 0$) a todos os caminhos passíveis de visitação pelas formigas. A intenção é estimular uma busca inicialmente aleatória, uma vez que todas as trajetórias possuem a mesma concentração de feromônio. Dorigo *et al.* (1996) realizou experimentos com $\tau_0 = 0$ e $\tau_0 = 1/(d * L_{dd})$, onde d expressa o número de pontos de

Revista Produção On Line
Associação Brasileira de Engenharia de Produção - ABEPRO
Universidade Federal de Santa Catarina - UFSC
www.producaoonline.org.br
ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

decisão ao longo do processo de busca e L_{dd} representa o tamanho da viagem produzida pela heurística do vizinho mais próximo (ROSENKRANTZ *et al.*, 1977). Como resultado da pesquisa, constatou-se que $\tau_0 = 1/(d * L_{dd})$ rendeu melhor desempenho que $\tau_0 = 0$, sendo um potencial valor a ser adotado nessa etapa do método.

A etapa de construção das soluções viáveis de um problema é coordenada pela atuação das formigas artificiais, que podem ser interpretadas como heurísticas construtivas de soluções probabilísticas obtidas por meio do seqüenciamento dos componentes de solução estabelecidos. Toda solução é iniciada por uma construção parcial $s^p = \langle\,\rangle$, sendo essa progressivamente preenchida por novos elementos absorvidos do conjunto de componentes de soluções viáveis $\mathfrak{N}(s^p)$, definido a partir das restrições impostas ao problema e integradas ao mecanismo de construção de soluções viáveis configurado. Nesse sentido, cada componente de solução $c_i^j \in \mathfrak{N}(s^p)$ possui uma determinada probabilidade de ser escolhido como próximo ponto a ser visitado em um procedimento de busca. Tal probabilidade, conhecida como probabilidade ou regra de transição, pode assumir uma gama de funções estocásticas. No entanto, a maioria dos algoritmos de ACO adota a seguinte equação:

$$p(c_i^j | s^p) = \frac{[\tau_i^j]^\alpha \cdot [\eta(c_i^j)]^\beta}{\sum_{c_k^l \in \mathfrak{N}(s^p)} [\tau_k^l]^\alpha \cdot [\eta(c_k^l)]^\beta}, \forall\, c_i^j \in \mathfrak{N}(s^p) \qquad (6)$$

Onde,

- η representa a informação heurística do componente de solução c_i^j;
- τ_i^j expressa a quantidade de feromônio presente em um dado componente de solução;
- α ($\alpha > 0$) significa a importância relativa do valor de feromônio; e
- β ($\beta > 0$) representa a importância relativa da informação heurística.

Em suma, quanto maior a relação entre quantidade de feromônio e informação heurística, maior a probabilidade de seleção de um determinado componente de solução. Uma vez construídas as soluções, pode-se, opcionalmente, aplicar um procedimento complementar de busca local, objetivando-se aprimorar a qualidade dos resultados obtidos com a aplicação

ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

da regra de transição. Apesar de não ser obrigatória, a utilização de algoritmos de busca local vem apresentando bons resultados experimentais, como demonstram Heinonen e Pettersson (2007), Huang e Liao (2008) e Rossi e Boschi (2009), entre outros.

A partir da construção das soluções pelos agentes artificiais e o eventual melhoramento das mesmas por meio de alguma técnica complementar, tem-se início a etapa de atualização do modelo de feromônio a partir das soluções identificadas. Essa etapa tem como objetivos principais valorizar os melhores componentes de solução através de acréscimos em seus valores de feromônio, assim como estimular a exploração de componentes alternativos, uma vez que a evaporação de parte do feromônio depositado faz com que as formigas não tenham suas escolhas prematuramente "fixadas", permitindo com que o espaço de busca seja melhor analisado e impedindo o aprisionamento dos resultados em regiões de mínimos locais.

Muitas regras podem ser adotadas para se atualizar modelos de feromônio. No entanto, conforme salientam Dorigo e Blum (2005), a grande maioria das fórmulas pode ser extraída da seguinte equação:

$$\tau_i^j = (1-\rho)\cdot\tau_i^j + {}^{\rho}/_{\mathfrak{S}_{upd}} \cdot \sum_{\{s\in\mathfrak{S}_{upd}\,|\,c_i^j\in s\}} F(s) \qquad (7)$$

Onde,

- $\rho \in [0,1]$ é a taxa de evaporação de feromônio (*feedback* negativo);
- $F(s)$ representa uma medida qualitativa da função-objetivo a ser otimizada; e
- $\mathfrak{S}_{upd}$ representa o conjunto de combinações possíveis entre os componentes de solução.

O conjunto $\mathfrak{S}_{upd}$, na verdade, envolve um grupo de componentes que pode variar, dependendo das especificações atribuídas ao algoritmo. No Sistema de Colônia de Formigas – SCF (Dorigo e Gambardella, 1997), adota-se $\mathfrak{S}_{upd} \leftarrow s_{bs}$, onde somente a melhor solução obtida ao longo de todas as iterações já efetuadas (s_{bs}) é considerada para fins de atualização da quantidade de feromônio. No Sistema de Formigas MIN-MAX – SFMM (Stützle e Hoos, 2000), por outro lado, aplicam-se $\mathfrak{S}_{upd} \leftarrow s_{bs}$ e $\mathfrak{S}_{upd} \leftarrow argmax\,\{F(s) \mid s \in \mathfrak{S}_{iter}\}$,

ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

circunstância em que apenas a melhor solução da iteração atual sofre atualizações de feromônio.

Finalmente, tem-se a etapa de ações complementares, que abrange uma série de medidas centrais ao método, e que não podem ser executadas individualmente pelas formigas artificiais. Em geral, tais ações incluem algoritmos complementares de busca local e regras específicas de atualização do modelo de feromônio. Contudo, essa etapa pode contemplar quaisquer outras inovações tecnológicas sugeridas ao método ACO desenvolvido.

5 TENDÊNCIAS DE APLICAÇÕES DE ACO EM JSSPs

O primeiro estudo que integrou ACO a JSSPs é atribuído à Colorni *et al.* (1994), aplicando a versão original do método ACO, conhecida por Sistema de Formigas (ou *Ant System* – AS), a cenários do tipo JSS. No entanto, em sua primeira versão, utilizada inicialmente no tratamento de problemas do tipo caixeiro viajante, a técnica ACO mostrou-se pouco satisfatória à resolução de vários problemas de otimização combinatória, impulsionando uma série de pesquisas inovadoras em prol de melhorias ao método.

Nesse contexto, uma gama de variações do algoritmo original foi desenvolvida, sendo que algumas alcançaram resultados tão promissores que acabaram por se distinguirem das demais, recebendo denominações próprias. Dentre as mais conhecidas, destacam-se o Sistema de Formigas Elitista – SFE (Dorigo *et al.*, 1996), o Sistema de Formigas Ranqueado – SFR (Bullnheimer *et al.*, 1999), o *Framework* Hipercúbico – FH (Blum e Dorigo, 2004), o SFMM e SCF, ambos já comentados preteritamente.

Com base em tais variações, novas possibilidades de estudo foram identificadas, destacando-se as pesquisas de Blum (2002) e Zhang *et al.* (2006), em que métodos inspirados nas variações SFMM e SCF, respectivamente, foram testados junto a cenários de JSS. Essas pesquisas revelam que o desempenho das técnicas experimentadas é muito satisfatório, inclusive quando comparado a outros algoritmos de otimização. Apesar de cada variação do método ACO apresentar suas peculiaridades e, naturalmente, ser mais adequada para determinados tipos de problema, Stützle e Dorigo (2002) esclarecem que as técnicas que estabelecem níveis positivos mínimos para os valores de feromônio alcançam melhores resultados, pois evitam com que soluções “zeradas” sejam produzidas. Tais métodos

ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

constituem uma classe de algoritmos conhecida por $ACO_{\tau_{min}}$, sendo que SFMM e SCF fazem parte de tal grupo, constituindo um dos motivos que as tornam duas das técnicas de ACO mais bem sucedidas na resolução de JSSPs.

Pesquisas como a de Figlali *et al.* (2009) também abordam a ACO aplicada em JSSPs, porém sob uma perspectiva distinta. Nesse estudo, os autores investigam a influência das variáveis básicas de controle de um método ACO (número de formigas - *m*, número de viagens - *t*, taxa de evaporação - ρ, importância relativa de feromônio – α, e importância relativa da informação heurística - β), partindo da otimização de JSSPs segundo o critério de desempenho *makespan*. Adotando-se Análise de Variância (ANOVA) em cinco grupos de experimentos com características distintas relacionadas aos parâmetros e aos próprios cenários de JSS, os autores recorrem, inicialmente, a uma técnica de planejamento experimental conhecida como planejamento fatorial completo, adotando 2^5 replicações e, posteriormente, testam o planejamento fatorial fracionário 2^{5-2}, comparando seus resultados.

A pesquisa constatou que o método ACO apresenta melhor desempenho com altos valores de *m* e *t* e baixos valores de β, sendo que não foi verificada tendência explícita em relação às variáveis α e ρ. No que diz respeito aos dois tipos de planejamento fatorial utilizados, observou-se a equivalência estatística entre ambos, sendo os resultados registrados por ambas as abordagens praticamente os mesmos. Isso mostra que, para determinados cenários, o planejamento fatorial fracionário pode gerar resultados de modo mais rápido e compacto, sem comprometimento da validade estatística dos mesmos.

Não obstante, mesmo quando consideradas as variações de ACO desenvolvidas com sucesso nos últimos anos, o método ainda apresenta resultados inferiores àqueles obtidos pelos algoritmos que representam o estado da arte em otimização de JSSPs (DORIGO e STÜTZLE, 2004). Nesse sentido, uma iniciativa de pesquisa que vem se destacando é a hibridização entre ACO e outras técnicas de otimização, tais como algoritmos de programação matemática e busca local.

A hibridização pode ocorrer de várias formas, envolvendo os mais diferentes métodos conhecidos. De modo geral, pode-se aplicar hibridismo de duas maneiras: a complementar, onde um algoritmo atua sobre os resultados gerados por outro, refinando-os e melhorando o

ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

desempenho da otimização, como é o caso dos trabalhos de Heinonen e Pettersson (2007), Yagmahan e Yenisey (2008), Huang e Liao (2008), e Atighehchian *et al.* (2009).

Essas pesquisas exemplificam muito bem as vantagens em se adotar métodos híbridos, onde as características de dois ou mais algoritmos são incorporadas ao processo de otimização, resultando em desempenhos superiores aos apresentados pelas técnicas isoladamente consideradas. Vale ressaltar ainda que essa é uma vasta área de pesquisa, uma vez que existem diversas possibilidades de hibridização. Os trabalhos supracitados, por exemplo, atuam com hibridizações entre ACO e busca *tabu*, e ACO e otimização não-linear.

A segunda forma de hibridização acontece sob uma perspectiva simultânea, situação em que um mesmo domínio de soluções é tratado concomitantemente por dois ou mais métodos de otimização, podendo-se configurar posturas colaborativas ou concorrentes entre os algoritmos trabalhados. Tseng e Chen (2006) estudam o desempenho de um método híbrido, composto por ACO, algoritmo genético e um procedimento complementar de busca local, experimentado na resolução de problemas de programação de projetos com restrições de recursos.

Além de alcançar resultados satisfatórios, os autores constatam que a combinação de algumas características inerentes aos métodos verificados apresenta-se como principal razão da superior eficiência da hibridização, quando comparada com cada técnica isoladamente avaliada. Nessa mesma linha de pesquisa, Rossi e Boschi (2009) realizam estudo inovador, ao tratarem um JSSP flexível por meio de ACO e algoritmos genéticos, configurados para atuarem simultaneamente de maneira colaborativa, ou seja, trocando informações entre suas respectivas populações. O método híbrido proposto alcançou resultados muito positivos, superando a eficiência dos algoritmos individualmente testados.

No que diz respeito às novas fronteiras de pesquisa envolvendo ACO e JSSP, pode-se observar dois tipos de tendências, uma relacionada aos princípios teóricos que cerceiam o desenvolvimento de novos algoritmos híbridos de ACO, e outra referente aos cenários de JSS analisados.

5.1 HIBRIDIZAÇÃO ENTRE ACO E OUTROS MÉTODOS DE OTIMIZAÇÃO

ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

No âmbito das hibridizações, Blum (2005a) esclarece que, nos últimos anos, muitos estudos investiram na hibridização entre ACO e outros algoritmos de otimização, destacando-se procedimentos de busca local. Porém, grande parte desses estudos passa a gerar resultados insatisfatórios quando problemas muito complexos ou com um número alto de restrições são analisados. Logo, o autor sugere a adoção de técnicas de inteligência artificial e pesquisa operacional clássicas, partindo-se do princípio de que a natureza construtiva de ACO adéqüa-se promissoramente a estruturas de árvores de busca, de forma análoga a métodos como *beam search* e programação de restrições.

Pesquisas como as desempenhadas por Sabuncuoglu e Bayiz (1999) e Valente e Alves (2005) preocupam-se em experimentar e investigar a eficiência da técnica *beam search* junto a problemas de programação da produção dos tipos JSS e *single machine scheduling* (SMS), respectivamente. Avaliando indicadores de desempenho tais como *makespan*, atraso total médio e custos médios de atraso e antecipação de *jobs* finalizados, os trabalhos alcançam resultados promissores, comprovando a adequabilidade da técnica para problemas de otimização dessa ordem.

Blum (2005b), por outro lado, desenvolve um método híbrido, incorporando características de busca em árvore típicas da técnica *beam search* ao mecanismo de construção de soluções probabilísticas de ACO. Ao aplicar o algoritmo construído (denominado pelo autor de *Beam*-ACO) em cenários de programação da produção do tipo *open shop scheduling* (OSS), a pesquisa não somente confirmou a eficiência da hibridização entre *beam search* e ACO, como também identificou que, em se tratando de OSS, o método revelou-se como o melhor dentre os já experimentados, representando o estado da arte na resolução de OSS.

Coello (2002), por outro lado, aborda um conjunto de técnicas de programação de restrições e suas relações com alguns métodos evolutivos, incluindo o algoritmo ACO. Buscando avaliar a eficiência de seis das técnicas de programação de restrições trabalhadas em sua pesquisa, o autor as implanta em um algoritmo genético e testa sua eficiência, comparando-o com outros métodos de otimização baseados em programação matemática e algoritmos genéticos também. Apesar de inconclusivos, os resultados mostraram-se promissores, estimulando novos estudos na área.

ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

No que diz respeito a problemas de programação da produção, Trentesaux *et al.* (2001) e Yun e Gen (2002) desenvolvem e testam alguns algoritmos baseados em programação de restrições voltados a problemas do tipo JSS e SMS, respectivamente. Os resultados obtidos por tais estudos assinalam boas oportunidades de pesquisa envolvendo programação de restrições em cenários de programação da produção.

5.2 TENDÊNCIAS DE APLICAÇÃO DE ACO EM JSSP

Tratando-se de tendências de pesquisa em JSS, os estudos mais recentes procuram otimizar cenários cada vez mais complexos, especialmente em termos de flexibilidade de roteiros de produção, paralelismo de recursos produtivos e adoção de tempos de *setup* dependentes e múltiplos critérios de desempenho. Uma solução ótima, em se tratando de problemas com um único objetivo de desempenho, geralmente é bem definida. Não obstante, o mesmo não pode ser dito sobre situações onde dois ou mais critérios são identificados, uma vez que passa a existir um conjunto de soluções conciliatórias ótimas, o qual se apresenta para um processo de tomada de decisão onde, geralmente, é necessária alguma compensação (*trade-off*).

Roteiros de produção flexíveis e máquinas atuando em paralelo, analogamente, também caracterizam maiores dificuldades à resolução de JSSPs. Isto porque, de modo geral, pode-se dizer que JSSPs dessa natureza podem ser desmembrados em dois sub-problemas: o primeiro, de designação das operações a cada máquina pertencente a um grupo de equipamentos com características de processamento semelhantes; e outro, de seqüenciamento de tais operações nos respectivos recursos produtivos onde foram previamente alocadas (ROSSI e BOSCHI, 2009). Tal conjuntura produtiva caracteriza cenários de programação da produção mais complexos e maleáveis, uma vez que a disponibilidade de máquinas análogas divididas em grupos aumenta as possibilidades de alocação e seqüenciamento de operações.

De acordo com Rossi e Dini (2007), quando as máquinas disponíveis são agrupadas segundo funcionalidades análogas (capacidades equivalentes de processamento de operações), trata-se de um *flexible job shop scheduling problem* (FJSSP), com máquinas paralelas ou replicadas (FJSSP-PM). Formalmente, um FJSSP-PM pode ser caracterizado como *n jobs* que devem ser processados por *m* grupos de máquinas, cada um contendo m_j recursos paralelos

Associação Brasileira de Engenharia de Produção - ABEPRO
Universidade Federal de Santa Catarina - UFSC
www.producaoonline.org.br
ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

dotados de configurações operacionais análogas (j = 1, 2, ..., m). O valor $k = min_{j\ =\ 1,\ 2,\ ...,\ m}\ m_j$ representa o grau da capacidade de paralelização de um determinado sistema produtivo.

Quando os grupos de máquinas envolvidos apresentam o mesmo número de equipamentos, ou seja, $k = m_j$, defini-se o cenário como FJSSP-*k*PM. Cada *job j* deve ser processado de acordo com as restrições de dependência apresentadas por suas operações O_{ijr}, sendo esta a *r*-ésima operação do *job j* a ser processada em uma única máquina do grupo *i* de recursos. Cada operação apresenta um tempo de processamento t_{ijr} e $st(O_{ijr})$ e $t(O_{ijr})$ expressam, respectivamente, o tempo de início e término de uma operação.

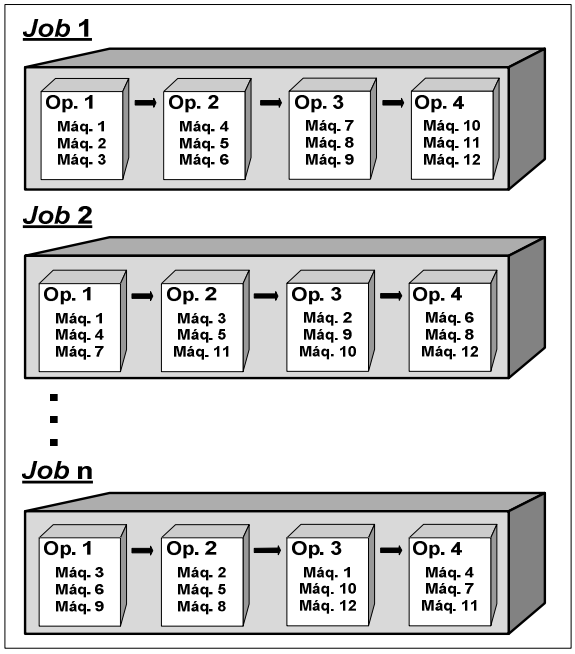


Figura 5: Exemplo esquemático de um FJSSP-*k*PM.

Retomando o exemplo de JSSP apresentado pela Figura 2, pode-se vislumbrar, por meio da Figura 5, como ele poderia ser formulado, caso fossem consideradas as diretrizes supracitadas, com especial atenção para a flexibilidade de roteiros produtivos e a disposição de múltiplos recursos de produção em paralelo.

Em relação a tempos de *setup*, estudos recentes vêm ratificando a importância de se considerar a existência e dependência entre tempos de *setup* para a otimização da programação da produção. Kim e Bobrowski (1994) chegam a estabelecer a relevância do tratamento de tempos de *setup* em três pontos básicos, quais seriam: tempos de *setup* são parte

ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

integrante do tempo de fluxo que afeta a taxa de saída de um sistema produtivo; o custo da unidade de tempo de *setup* é geralmente maior do que o custo unitário de tempo de processamento, pois engloba desde a paralisação de maquinário até a utilização de mão-de-obra especializada; e a realização de atividades de *setup* usualmente incorre na requisição de serviços técnicos com alto nível de especialização, sendo esses, por vezes, limitados.

Tratando-se mais particularmente da dependência entre tempos de *setup*, Cheng e Chen (1994) esclarecem que tempos de *setup* devem ser definidos como independentes somente se o tempo necessário para se ajustar os equipamentos de uma máquina ou grupo de recursos do processamento de um *job j* para um *job k* for o mesmo requisitado à execução do *job k* sem haver precedência alguma de *jobs*. Caso contrário, ou seja, se o tempo de preparação de *j* para *k* diferir do tempo de um outro *job g* para *k*, então existe dependência entre tempos de *setup* e essas devem ser consideradas à ocasião da otimização da programação da produção.

Nesse sentido, muitos métodos de otimização desenvolvidos nas últimas décadas assumem tempos de *setup* independentes, circunstância em que os mesmo são automaticamente incorporados aos tempos de processamento, ou irrelevantes, ocasião em que eles são ignorados. Porém, tais abordagens nem sempre representam satisfatoriamente situações reais de programação da produção. Ambientes produtivos relacionados a atividades de impressão, têxteis, químicas e metalúrgicas, por exemplo, apresentam tempos de *setup* significativos que dependem diretamente da seqüência de operações executada, não podendo ser negligenciados (RAJENDRAN e ZIEGLER, 2003).

Pesquisas como as realizadas em Barros e Moccellin (2004) e Moccellin e Nagano (2007) evidenciam a atenção prestada ao tratamento dos tempos de *setup* em cenários de programação da produção. Apesar de ambos os estudos abordarem o problema de programação da produção do tipo *flow shop* permutacional (FSP), o aspecto de maior relevância é a inclusão de tempos de *setup* na análise de otimização dos cenários abordados.

Barros e Moccellin (2004) investigaram a programação FSP, considerando tempos de *setup* assimétricos e dependentes da seqüência de operações observada. O estudo adota uma lógica de resolução diferenciada, pois, apesar de ter como medida de desempenho a duração total da programação (*makespan*), o foco do processo resolutivo é a identificação e otimização

ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

do somatório de tempos de *setup*, via *simulated annealing*, do recurso gargalo do sistema verificado. Além de registrar a flutuação do recurso gargalo ao longo do processo de otimização, a pesquisa também comprovou a complexidade atribuída à programação da produção quando observados tempos de *setup* dependentes.

Moccellin e Nagano (2007) efetuam análise mais abrangente, realizando experimentos computacionais no intuito de avaliar numericamente o grau de equivalência dos espaços de soluções entre o problema de programação da produção FSP com tempos de *setup* e o problema cíclico assimétrico do caixeiro viajante. Por meio da análise supracitada, os autores constataram que, quando se opera com tempos de *setup* separados dos tempos de processamento em máquina, pode-se calcular um valor de limitante superior do tempo de máquina parada entre sua preparação (*setup*) e a execução de uma tarefa.

Kim e Brobowski (1997), por sua vez, realizam pesquisa direcionada à avaliação dos impactos proporcionados por variações de tempos de *setup* dependentes em problemas de programação da produção. Baseados nas pesquisas de Wilbrecht e Prescott (1969) e Deane e Yang (1992), que comprovaram que as implicações de tempos de *setup* na performance da programação de produção são maiores do que aquelas provocadas pelos tempos de processamento, especialmente em ambientes com tempos de *setup* dependentes, os autores investigaram o impacto das variações de *setup* no desempenho da programação da produção e de determinadas regras de seqüenciamento consideradas, segundo um planejamento fatorial completo aplicado a três fatores, estruturado da seguinte maneira: variação de *setup* (cinco níveis), regras de seqüenciamento (quatro níveis) e o rigor das datas de entrega (dois níveis).

Como resultados da pesquisa, as regras de seqüenciamento que consideravam as variações nos tempos de *setup* mostraram desempenho estatisticamente superior as demais. De modo geral, a pesquisa demonstra que o desempenho de um ambiente de programação sofre deterioração à medida que variações de *setup* são consideradas. Quanto maiores os valores de variação e mais rigorosos os prazos de entrega (caso esses existam e sejam considerados para efeitos de programação), mais importante torna-se a utilização de regras de seqüenciamento adequadas que observem os tempos de *setup* existentes.

5.3 TENDÊNCIAS DE ACO E JSSP NO BRASIL

ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

Observando particularmente as pesquisas nacionais envolvendo o método ACO e problemas de programação da produção, nota-se que a produção científica ainda é incipiente. De acordo com Neto e Filho (2008), que realizam um levantamento bibliográfico dos estudos nacionais envolvendo a técnica ACO, o método vem sendo gradativamente mais pesquisado e experimentado em problemas de otimização combinatória, especialmente aqueles dos tipos roteirização de veículos e caixeiro viajante. Ainda assim, os autores registram que, quando comparado com outras metaheurísticas, tais como algoritmos genéticos, *simulated annealing* e busca *tabu*, o método ACO ainda é pouco explorado, representando aproximadamente 6,15% dos trabalhos apresentados em eventos nacionais e 0% dos artigos registrados na base de dados *Scielo* (http://www.scielo.br/).

Algumas pesquisas nacionais ratificam consistentemente as tendências de pesquisa envolvendo ACO no Brasil. Lorenzoni *et al.* (2006) desenvolvem e aplicam um método de otimização híbrido combinando ACO e *simulated annealing* à resolução de 536 instâncias de uma classe de problemas de escalonamento com restrições de recursos e múltiplos modos de processamento, obtendo resultados promissores ao comparar a técnica construída a outras aplicadas a mesma classe de problemas e já apresentadas pela literatura.

Minikovski e Vieira (2008), por sua vez, realizam estudo direcionado ao problema de planejamento mestre da produção, no intuito de otimizá-lo, via ACO, com base na qualidade dos planos mestres gerados e no tempo computacional requerido, também comparando o desempenho do algoritmo desenvolvido com o método de programação matemática *branch-and-bound* e uma variante hibridizada entre esse e ACO.

Analisando-se oito cenários de complexidades distintas e explorando os recursos estatísticos de planejamento fatorial completo 2^k e ANOVA, a pesquisa conseguiu demonstrar que as técnicas ACO hibridizada e *branch-and-bound* são superiores à ACO pura em termos de qualidade das soluções obtidas, apresentando diferenças estatisticamente relevantes entre suas médias amostrais. Em se tratando do tempo computacional, o mesmo foi considerado estatisticamente equivalente para todas.

Tratando-se mais especificamente do método ACO aplicado em cenários de programação da produção do tipo JSS, Santos (2008) desenvolve um algoritmo ACO voltado à programação da produção para trás, em sistemas produtivos com um único estágio de

ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

processamento, recursos produtivos paralelos e roteiros operacionais flexíveis. Por meio do planejamento fatorial completo 2^k e ANOVA, a pesquisa demonstrou que, dentre os seis parâmetros de configuração do método ACO investigados, somente um não se mostrou relevante ao desempenho da técnica. Além disso, comprovou-se estatisticamente que, sob o ponto de vista do *makespan*, as técnicas experimentadas (ACO e *branch-and-bound*) apresentaram eficiência equivalente. No que diz respeito ao tempo computacional, verificou-se que o algoritmo ACO é mais vantajoso quando comparado ao outro método considerado.

6 CONSIDERAÇÕES FINAIS

Os mais diversos tipos de problemas de otimização despertam a curiosidade científica há muitos anos. Nesse contexto, aqueles classificados como NP-*hard* destacam-se não somente pela complexidade de se encontrar soluções ótimas em tempo computacional hábil, mas também e, principalmente, pelo fato dos mesmos representarem uma parcela considerável dos problemas identificados em nosso dia-a-dia. A programação da produção é um bom exemplo de problema de otimização não-polinomial continuamente observado em ambiente organizacional, tornando-se ainda mais importante quando considerada a atual realidade competitiva definida pela globalização.

Constituindo um problema com certo grau de complexidade, o JSSP foi arduamente estudado pela comunidade científica, sendo várias as técnicas de otimização já experimentadas, desde métodos de programação matemática até algoritmos evolutivos de abordagem estocástica. Mais recentemente, as metaheurísticas de inteligência coletiva, fundamentadas na modelagem e representação de um grupo de condutas sociais praticado por determinados animais, vêm recebendo destaque, sendo ACO um dos métodos mais aplicados junto a JSSPs.

Diante da revisão teórica apresentada, pôde-se constatar que as pesquisas desenvolvidas nessa área de estudo sinalizam duas tendências muito claras, quais seriam: análise da eficiência de algoritmos de otimização fundamentados em ACO, porém hibridizados com outras técnicas, especialmente aquelas de caráter construtivo, relacionadas à inteligência artificial e pesquisa operacional; e investigação de cenários de JSS mais complexos, caracterizados por aspectos tais como flexibilidade de roteiros de produção,

ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

compartilhamento de recursos produtivos e tempos de *setup* dependentes do seqüenciamento das operações.

Analisando especificamente o contexto de pesquisas envolvendo ACO e JSSP no Brasil, percebe-se que, apesar de uma queda sutil em 2007, a quantidade de trabalhos nessa área vem crescendo significativamente nos últimos anos, contudo em proporção relativamente menor do que o volume de trabalhos publicados a nível mundial. Apesar da atenção progressivamente maior dispensada ao método ACO, sua aplicação na otimização de problemas de programação da produção ainda não representa um foco proeminente das pesquisas desenvolvidas no país, apontando uma grande oportunidade para estudos futuros.

Nesse contexto, muitas outras oportunidades de pesquisa são identificadas, além daquelas já mencionadas ao longo deste trabalho. Torna-se interessante, por exemplo, avaliar até que ponto a hibridização entre diferentes métodos de otimização apresenta-se como medida positiva no tratamento de outros tipos de problemas, inclusive de outros ambientes de *shop scheduling*, tais como *single machine shop* (SMS), *parallel machine shop* (PMS), *flow shop scheduling* (FSS) e *open shop scheduling* (OSS).

Outra potencial iniciativa de pesquisa é a verificação das principais abordagens de configuração adotadas no desenvolvimento de outros métodos de otimização igualmente experimentados em JSSPs. Assim, pode-se não somente observar se as mesmas tendências percebidas em relação à ACO também estão presentes na construção de outros algoritmos testados junto ao JSSP, como também, conhecendo os fundamentos teóricos e práticos de outras técnicas, refletir sobre possibilidades mais vantajosas de hibridização envolvendo ACO, conjugando as principais características e tendências dos métodos de otimização hibridizados. Destarte, não apenas o conhecimento sobre ACO é amadurecido como também sobre as demais técnicas de otimização conhecidas e, não menos relevante, sobre o problema de programação da produção do tipo JSS e suas peculiaridades.

REFERÊNCIAS

ALEXANDER, S. M. Expert system for the selection of scheduling rules in a job-shop. **Computers & Industrial Engineering**, Vol. 12, n. 3, p. 167-171, 1987.

ARTIGUES, C.; FEILLET, D. A branch and bound method for the job-shop problem with sequence-dependent *setup* times. **Annals of Operations Research**, Vol. 159, n. 1, p. 135-159, 2008.

ARTIGUES, C.; GENDREAU, M.; ROUSSEAU, L; VERGNAUD, A. Solving an integrated employee timetabling and job-shop scheduling problem via hybrid branch-and-bound. **Computers & Operations Research**, Vol. 36, n. 8, p. 2330-2340, 2009.

ATIGHEHCHIAN, A.; BIJARI, M.; TARKESH, H. A novel hybrid algorithm for scheduling steel-making continuous casting production. **Computers & Operations Research**, Vol. 36, n. 8, p. 2450-2461, 2009.

AYDIN, M. E.; FOGARTY, T. C. A simulated annealing algorithm for multi-agent systems: a job-shop scheduling application. **Journal of Intelligent Manufacturing**, Vol. 15, n. 6, p. 805-814, 2004.

BARNES, J. W.; CHAMBERS, J. B. **Flexible job shop scheduling by tabu search**. Technical report series ORP 96/09. Department of Mechanical Engineering, University of Texas at Austin, 1996.

BARROS, A. D. DE; MOCCELLIN, J. V. Análise da flutuação do gargalo em flow shop permutacional com tempos de *setup* assimétricos e dependentes da seqüência. **Gestão & Produção**, Vol. 11, n. 1, p. 101-108, 2004.

BECKERS, R.; DENEUBOURG, J. L.; GOSS, S. Modulation of trail laying in the ant Lasius niger (Hymenoptera: Formicidae) and its role in the collective selection of a food source. **Journal of Insect Behavior**. Vol. 6, n. 6, p. 751–759, 1993.

BECKERS, R.; DENEUBOURG, J. L.; GOSS, S. Trails and U-turns in the selection of the shortest path by the ant Lasius niger. **Journal of Theoretical Biology**, Vol. 159, n. 4, p. 397–415, 1992.

BLUM, C. Ant colony optimization: introduction and recent trends. **Physics of Life Reviews**, Vol. 2, n. 4, p. 353-373, 2005a.

BLUM, C. Beam-ACO — hybridizing ant colony optimization with beam search: an application to open shop scheduling. **Computers & Operations Research**, Vol. 32, n. 6, p. 1565-1591, 2005b.

BLUM, C. **Metaheuristics for group shop scheduling**. 2002. 80 f. Thesis (Diplome D'etudes Approfondies) - Université Libre de Bruxelles, Brussels, 2002.

BLUM, C.; DORIGO, M. The Hyper-Cube Framework for Ant Colony Optimization. **IEEE Transactions on Systems, Man, and Cybernetics – Part B**, Vol. 34, n. 2, p. 1161-1172, 2004.

BLUM, C.; SAMPELS, M. An ant colony optimization algorithm for shop scheduling problems. **Journal of Mathematical Modelling and Algorithms**, Vol. 3, n. 3, p. 285-308, 2004.

BULLNHEIMER, B.; HARTL, R. F.; STRAUSS, C. A new rank-based version of the ant system: a computational study. **Central European Journal for Operations Research and Economics**, Vol. 7, n. 1, p. 25-38, 1999.

BUSTAMANTE, L. M. **Minimização do custo de antecipação e atraso para o problema de seqüenciamento de uma máquina com tempo de preparação dependente da seqüência**: aplicação em uma usina siderúrgica. 2007. 86 f. Dissertação (Mestrado em Engenharia de Produção) – Universidade Federal de Minas Gerais, Belo Horizonte, 2007.

CHENG, T. C. E.; CHEN, Z.-L. Parallel machine scheduling with batch *setup* times. **Operations Research**, Vol. 42, n. 6, p. 1171-1174, 1994.

COELLO, C. A. C. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. **Computer Methods in Applied Mechanics and Engineering**, Vol. 191, n. 11-12, p. 1245-1287, 2002.

ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

COLORNI, A.; DORIGO, M.; MANIEZZO, V.; TRUBIAN, M. Ant system for job-shop scheduling. **JORBEL – Belgian Journal of Operations Research, Statistics and Computer Science**, Vol. 34, n. 1, p. 39-53, 1994.

CORRÊA, H. L.; CORRÊA, C. A. **Administração de produção e operações**: manufatura e serviços – uma abordagem estratégica. 2. ed. São Paulo, SP: Atlas, 2006.

COX, J. F.; BLACKSTONE, J. H., Jr.; SPENCER, M. S. **APICS Dictionary, american production and inventory control society**. Falls Church, VA: APICS, 1992.

DEANE, R. H.; YANG, J. Product mix selection and closed manufacturing cell flow time performance. **International Journal of Production Economics**, Vol. 28, n. 2, p. 157-169, 1992.

DELL AMICO, M.; TRUBIAN, M. Applying tabu search to the job shop scheduling problem. **Annals of Operations Research**, Vol. 41, n. 1-4, p. 231-252, 1993.

DORIGO, M.; BLUM, C. Ant colony optimization theory: a survey. **Theoretical Computer Science**, Vol. 344, n. 2-3, p. 243-278, 2005.

DORIGO, M.; GAMBARDELLA, L. M. Ant colony system: a cooperative learning approach to the traveling salesman problem. **IEEE Transactions on Evolutionary Computation**, Vol. 1, n. 1, p. 53-66, 1997.

DORIGO, M.; MANIEZZO, V.; COLORNI, A. **Ant System**: an autocatalytic optimizing process. Relatório Técnico, 91-016, revisado. Itália: Universidade de Milão, 1991.

DORIGO, M.; MANIEZZO, V.; COLORNI, A. Ant system: optimization by a colony of cooperating agents. **IEEE Transactions on Systems, Man, and Cybernetics – Part B**: Cybernetics, Vol. 26, n. 1, p. 29-41, 1996.

DORIGO, M.; STÜTZLE, T. **Ant Colony Optimization**. Cambridge: MIT Press, 2004.

ESSAFI, I.; MATI, Y.; DAUZÈRE-PÉRÈS, S. A genetic local search algorithm for minimizing total weighted tardiness in the job-shop scheduling problem. **Computers & Operations Research**, Vol. 35, n. 8, p. 2599-2616, 2008.

FATTAHI, P.; MEHRABAD, M. S.; JOLAI, F. Mathematical modeling and heuristic approaches to flexible job shop scheduling problems. **Journal of Intelligent Manufacturing**, Vol. 18, n. 3, p. 331-342, 2007.

FIGLALI, N; ÖZKALE, C; ENGIN, O.; FIGLALI, A. Investigation of ant system parameter interactions by using design of experiments for job-shop scheduling problems. **Computers & Industrial Engineering**, Vol. 56, n. 2, p. 538-559, 2009.

GAREY, M. R.; JOHNSON, D. S. **Computers and intractability**: a guide to the theory of np-completeness. New York: W. H. Freeman e Co, 1979.

GOLDBARG, M. C.; LUNA, H. P. L. **Otimização combinatória e programação linear**. 2. ed. Rio de Janeiro, RJ: Elsevier, 2005.

GOLDBERG, D. E. **Genetic algorithms in search, optimization, and machine learning**. Reading, MA: Addison-Wesley, 1989.

GOSS, S.; ARON, S.; DENEUBOURG, J. L.; PASTEELS, J. M. Self-organized shortcuts in the Argentine ant. **Naturwissenschaften**, Vol. 76, n. 12, p. 579–581, 1989.

ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

HEINONEN, J.; PETTERSSON, F. Hybrid ant colony optimization and visibility studies applied to a job-shop scheduling problem. **Applied Mathematics and Computation**, Vol. 187, n. 2, p. 989-998, 2007.

HUANG, K.-L.; LIAO, C.-J. Ant colony optimization combined with taboo search for the job shop scheduling problem. **Computers & Operations Research**, Vol. 35, n. 4, p. 1030-1046, 2008.

JAIN, A. S.; MEERAN, S. Deterministic job-shop scheduling: past, present and future. **European Journal of Operational Research**, Vol. 113, n. 2, p. 390-434, 1999.

JOHNSON, K.; ROSSI, L. F. A mathematical and experimental study of ant foraging trail dynamics. **Journal of Theoretical Biology**, Vol. 241, n. 2, p. 360-369, 2006.

KATHAWALA, Y.; ALLEN, W. R. Expert systems and job shop scheduling. **International Journal of Operations and Production Management**, Vol. 13, n. 2, p. 23-35, 1993.

KIM, S. C.; BOBROWSKI, P. M. Impact of sequence-dependent *setup* time on job shop scheduling performance. **International Journal of Production Research**, Vol. 32, n. 7, p. 1503-1520, 1994.

KIM, S. C.; BOBROWSKI, P. M. Scheduling jobs with uncertain *setup* times and sequence dependency. **Omega**, Vol. 25, n. 4, p. 437-447, 1997.

KUNNATHUR, A. S.; SUNDARARAGHAVAN, P. S.; SAMPATH, S. Dynamic rescheduling using a simulation-based expert system. **Journal of Manufacturing Technology Management**, Vol. 15, n. 2, p. 199-212, 2004.

KUSIAK, A.; CHEN, M. Expert systems for planning and scheduling manufacturing systems. **European Journal of Operational Research**, Vol. 34, n. 2, p. 113-130, 1988.

LAARHOVEN, P. J. M.; AARTS, E. H. L.; LENSTRA, J. K. Job shop scheduling by simulated annealing. **Operations Research**, Vol. 40, n. 1, p. 113-125, 1992.

LAND, A. H.; DOIG, A. G. An automatic method for solving discrete programming problems. **Econometrica**, Vol. 28, n. 3, p. 497-520, 1960.

LING, W. **Shop scheduling with genetic algorithm**. Beijing: Tsinghua University e Springer Press, 2003.

LORENZONI, L. L.; AHONEN, H. T.; ALVARENGA, A. G. DE. Um algoritmo híbrido baseado em colônia de formigas e recozimento simulado para problemas de escalonamento com restrição de recursos e múltiplos modos de processamento. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 26., 2006, Fortaleza. **Anais...** Fortaleza: ABEPRO, 2006, p. 1-9.

MCMULLEN, P. R. An ant colony optimization approach to addressing a JIT sequencing problem with multiple objectives. **Artificial Intelligence in Engineering**, Vol. 15, n. 3, p. 309-317, 2001.

MINIKOVSKI, L. J.; VIEIRA, G. E. Otimização do planejamento mestre da produção por meta-heurística colônia de formigas. **Revista INGEPRO - Inovação, Gestão e Produção**, Vol. 1, n. 2, p. 1-15, 2009.

MOCCELLIN, J. V.; NAGANO, M. S. Uma propriedade estrutural do problema de programação da produção flow shop permutacional com tempos de *setup*. **Pesquisa Operacional**, Vol. 27, n. 3, p. 487-515, 2007.

MORTON, T.; PENTICO, D. **Heuristic scheduling systems**. New York, NY: John Wiley e Sons, 1993.

ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

NETO, R. F. T.; FILHO, M. G. A aplicação no Brasil da meta-heurística de colônia de formigas em problemas de Engenharia de Produção: caminhos traçados e oportunidades. In: SIMPÓSIO DE ENGENHARIA DE PRODUÇÃO, 15., 2008, Bauru. **Anais...** Bauru: UNESP, 2008, p. 1-12.

PAPADIMITRIOU, C. H.; STEIGLITZ, K. **Combinatorial optimization**: algorithms and complexity. New York, NY: Dover Publications Inc., 1982.

PARSOPOULOS, K. E.; VRAHATIS, M. N. Recent approaches to global optimization problems through particle swarm optimization. **Natural Computing**, Vol. 1, n. 2-3, p. 235-306, 2002.

PINEDO, M. **Scheduling theory, algorithms, and systems**. Englewood Cliffs, NJ: Prentice-Hall, 1995.

RAJENDRAN, C.; ZIEGLER, H. Scheduling to minimize the sum of weighted flow time and weighted tardiness of jobs in a flowshop with sequence-dependent *setup* times. **European Journal of Operational Research**, Vol. 149, n. 3, p. 513-522, 2003.

ROSENKRANTZ, D. J.; STEARNS, R. E.; LEWIS, P. M. An analysis of several heuristics for the traveling salesman problem. **SIAM Journal of Computing**, Vol. 6, n. 3, p. 563-581, 1977.

ROSS, P.; HART, E.; CORNE, D. Evolutionary scheduling: a review. **Genetic Programming and Evolvable Machines**, Vol. 6, n. 2, p. 191-220, 2005.

ROSSI, A.; BOSCHI, E. A hybrid heuristic to solve the parallel machines job-shop scheduling problem. **Advances in Engineering Software**, Vol. 40, n. 2, p. 118-127, 2009.

ROSSI, A.; DINI, G. Flexible job-shop scheduling with routing flexibility and separable *setup* times using ant colony optimisation method. **Robotics and Computer-Integrated Manufacturing**, Vol. 23, n. 5, p. 503-516, 2007.

SABUNCUOGLU, I.; BAYIZ, M. Job shop scheduling with beam search. **European Journal of Operational Research**, Vol. 118, n. 2, p. 390-412, 1999.

SADEH, N. M.; NAKAKUKI, Y. Focused simulated annealing search: an application to job shop scheduling. **Annals of Operations Research**, Vol. 63, n. 1, p. 77-103, 1996.

SANTOS, L. P. DOS. **Análise da otimização da programação de produção para trás em sistemas mono-estágio por colônia de formigas e sua comparação com branch and bound**. 2008. 109 f. Dissertação (Mestrado em Engenharia de Produção e Sistemas) – Pontifícia Universidade Católica do Paraná, Curitiba, 2008.

STÜTZLE, T.; DORIGO, M. A short convergence proof for a class of ACO algorithms. **IEEE Transactions on Evolutionary Computation**, Vol. 6, n. 4, p. 358-365, 2002.

STÜTZLE, T.; HOOS, H. H. MAX–MIN Ant System. **Future Generation Computer Systems**, Vol. 16, n. 9, p. 889–914, 2000.

TAILLARD, E. **Parallel tabu search technique for the jobshop scheduling problem**. Internal Report ORWP 89/11. Departemente de Mathematiques, Ecole Polytechnique Federale de Lausanne, Lausanne, 1989.

TRENTESAUX, D.; PESIN, P.; TAHON, C. Comparison of constraint logic programming and distributed problem solving: a case study for interactive, efficient and practicable job-shop scheduling. **Computers & Industrial Engineering**, Vol. 39, n. 1-2, p. 187-211, 2001.

TSENG, L.-Y.; CHEN, S.-C. A hybrid metaheuristic for the resource-constrained project scheduling problem. **European Journal of Operational Research**, Vol. 175, n. 2, p. 707-721, 2006.

ISSN 1676 - 1901 / Vol. X/ Num.I/ 2010

VALENTE, J. M. S.; ALVES, R. A. F. S. Filtered and recovering beam search algorithms for the early/tardy scheduling problem with no idle time. **Computers & Industrial Engineering**, Vol. 48, n. 2, p. 363-375, 2005.

VITTORI, K.; TALBOT, G.; GAUTRAIS, J.; FOURCASSIÉ, V.; ARAÚJO, A. F. R.; THERAULAZ, G. Path efficiency of ant foraging trails in an artificial network. **Journal of Theoretical Biology**, Vol. 239, n. 4, p. 507-515, 2006.

WILBRECHT, J. K.; PRESCOTT, W. B. The influence of *setup* time on job shop performance. **Management Science**, Vol. 16, n. 4, p. B-274-B-280, 1969.

XING, L.-N.; CHEN, Y.-W.; YANG, K.-W. Multi-objective flexible job shop schedule: design and evaluation by simulation modeling. **Applied Soft Computing**, Vol. 9, n. 1, p. 362-376, 2009.

YAGMAHAN, B.; YENISEY, M. M. Ant colony optimization for multi-objective flow shop scheduling problem. **Computers & Industrial Engineering**, Vol. 54, n. 3, p. 411-420, 2008.

YING, K.-C.; LIAO, C.-J. An ant colony system for permutation flow-shop sequencing. **Computers & Operations Research**, Vol. 31, n. 5, p. 791-801, 2004.

YUN, Y.-S.; GEN, M. Advanced scheduling problem using constraint programming techniques in SCM environment. **Computers & Industrial Engineering**, Vol. 43, n. 1-2, p. 213-229, 2002.

ZHANG, J.; HU, X.; TAN, X.; ZHONG, J. H.; HUANG, Q. Implementation of an ant colony optimization technique for job shop scheduling problem. **Transactions of the Institute of Measurement and Control**, Vol. 28, n. 1, p. 93-108, 2006.

Artigo recebido em 27/08/2009 e aceito para publicação em 12/03/2010