



RESOLUÇÃO DE PROBLEMAS DE PRODUÇÃO POR MEIO DA METAHEURÍSTICA PARTICLE SWARM OPTIMIZATION (PSO)

Carolina de A. S. Pinotti

Neida M. P. Volpi

carolpinotti16@gmail.com

neidavolpi@gmail.com

Afiliação:

Universidade Federal do Paraná (UFPR) - Av. Coronel Francisco Heráclito dos Santos, 210 - Jardim das Américas, 81531-900, Curitiba, Paraná, Brasil

Resumo. *O presente trabalho apresenta aplicações da metaheurística Particle Swarm Optimization (PSO) em problemas de produção. Apresenta-se uma proposta de resolução para um problema de Planejamento de Produção (com restrição de capacidade e preservação da produção), comparando-a às resoluções por métodos exatos e também aplicando à problemas de grande porte (dos quais não são conhecidos os valores ótimos). Essas aplicações foram primeiramente comparadas às soluções exatas de problemas de pequeno porte para, na sequência, serem aplicadas a problemas de grande porte.*

Palavras-chave: *Particle Swarm Optimization, Planejamento de Produção, Problemas de Produção, Metaheurísticas*

1 INTRODUÇÃO

O Planejamento de Produção (PP) é um problema encontrado por muitas indústrias no mundo inteiro. Tal problema aplica-se às indústrias que possuem uma demanda de itens a serem produzidos. Essa demanda pode ou não estar sujeita à uma restrição de capacidade (que pode ser tempo, matéria prima, uma combinação de ambos, etc). Para problemas de pequeno porte os melhores resultados serão mais facilmente obtidos. Quando o porte das indústrias é grande, ou o modelo encontrado muito complexo, precisa-se de métodos diferentes de solução do modelo de Programação Linear Mista para conseguir obter um resultado. Para isso existem as metaheurísticas, que são métodos que fornecem uma solução muitas vezes exata ou muito próxima da exata com um trabalho computacional muito menor, dependendo de sua programação.

Problemas de Planejamento de Produção de pequeno porte são rapidamente resolvidos por otimizadores como o LINGO ou CPLEX, por exemplo. Encontra-se, nestes casos, a melhor solução possível do problema, a solução ótima. Quando aumenta-se o tamanho do problema ou acrescentam-se algumas restrições, este problema ficará com uma maior complexidade e muitas vezes estes otimizadores já não conseguem resolvê-lo em um tempo computacional baixo. O uso da metaheurística PSO faz com que os problemas sejam resolvidos com um custo computacional muito menor, ou seja, mesmo problemas grandes são solucionados rapidamente. Porém não pode-se garantir que é a solução ótima, visto que esta muitas vezes ainda não é conhecida.

Para o problema trabalhado, cujo modelo é de Programação Linear Mista, a aplicação da metaheurística PSO traz muitas soluções infactíveis. Isso gera diversos problemas quanto à aplicação, o que muitas vezes faz com que não valha a pena a aplicação dessa metaheurística para esse tipo de problema. Deve-se, portanto, trabalhar para evitar estas infactibilidades, um trabalho que não é simples e sim o problema desta aplicação. O problema que foi trabalhado não só possui restrições de capacidade como também a possibilidade de manter a preparação da máquina para um determinado item de um período a outro, ou seja, trabalhou-se com preservação da preparação.

Fleischmann & Meyr (1997) apresentam uma heurística chamada Greedy-Mod para determinar a quantidade a ser produzida em cada período para um problema de produção. Toledo et al. (2010) apresenta uma aplicação desta heurística Greedy-Mod para problemas de Dimensionamento de Lotes e Programação da Produção, seguido da aplicação de PSO. Chen & Lin (2009) apresentam uma aplicação de PSO modificado em Planejamento de Produção. Para a aplicação do PSO em um problema de Planejamento de Produção com restrição de capacidade e preservação da preparação, utilizou-se várias ideias de cada um deste artigo, modificando um pouco a heurística Greedy-Mod apresentada de forma a ficar adequada a este problema.

Não são encontradas muitas aplicações de PSO em Problemas de Planejamento de Produção. É mais comumente encontrada a aplicação em problemas como Flowshop e Jobshop. Por isso este trabalho é baseado nos artigos acima descritos, na forma como é aplicado o PSO nestes outros problemas de produção e também como foram aplicadas outras metaheurísticas em problemas de Planejamento de Produção.

2 FORMULAÇÃO DO MODELO MATEMÁTICO PARA O PROBLEMA DE PLANEJAMENTO DE PRODUÇÃO

Considerando n o número de itens a ser produzidos em uma linha de produção que consiste em T períodos, com i , tais que, $i=1,\dots,n$, e t período, tal que $t=1,\dots,T$. Neste problema tem-se que cada item está sujeito a uma demanda, um custo de estoque por período e um tempo de produção. Cada período possui uma capacidade limite. Além disso há um tempo de preparação da máquina (setup), um custo de preparação da máquina e custo de estoque para um determinado item.

O problema a ser trabalhado possui a seguinte notação:

Dados:

n : número de itens do problema

T : número de períodos

Parâmetros:

d_{it} : demanda do item i ao final do período t

b_i : capacidade (em minutos) necessária do recurso da máquina para produzir uma unidade do item i

C_t : capacidade total (em minutos) disponível do recurso da máquina no período t

sp_i : tempo de preparação da máquina para processar o item i

s_i : custo de preparação do item i por período

h_i : custo de estoque do item i por período

I_{i0} : estoque inicial do item i

Algumas variáveis serão definidas para a formulação do problema. A variável y retornará o valor 1 quando a máquina for (ou já estiver) preparada para aquele item i no período t . Se houve produção do item i no período $t-1$ e a máquina ficou preparada para este item até o período t , diz-se que houve preservação da preparação. Então a variável y retornará 1 quando a máquina tiver sido preparada no período ou se teve a preparação mantida do período anterior para esse período. A variável w retorna o valor 1 quando houve essa preservação da preparação e a variável z retorna o valor 1 se o item i tiver sido preparado em um período $t-1$, produzido no período t e a preparação ainda tiver sido mantida até o período $t+1$. Ou seja, no período t o item i foi o único item a ser produzido (pois a preparação da máquina para este item foi mantida até o período $t+1$). Dessa forma, as variáveis serão definidas como a seguir.

Variáveis:

q_{it} : quantidade do item i produzida no período t

I_{it} : estoque do item i no final do período t

$y_{it} : \begin{cases} 1, & \text{se a máquina está preparada para o item } i \text{ no período } t \\ 0, & \text{caso contrário} \end{cases}$

$w_{it} : \begin{cases} 1, & \text{se o estado de preparação para o item } i \text{ continua do período } t-1 \text{ ao período } t \\ 0, & \text{caso contrário} \end{cases}$

$z_t : \begin{cases} 1, & \text{se não existe preparação de itens em } t, \text{ sendo que a preparação de um item} \\ & \text{específico ocorreu em } t-1 \text{ e é mantida até o período } t+1 \\ 0, & \text{caso contrário} \end{cases}$

Formulação do Modelo:

$$\min \sum_{i=1}^n \sum_{t=1}^T (s_i y_{it} + h_i I_{it}) \quad (1)$$

$$I_{it} = I_{i,t-1} + q_{it} - d_{it} \quad i = 1, \dots, n, \quad t = 1, \dots, T \quad (2)$$

$$\sum_{i=1}^n (sp_i y_{it} + b_i q_{it}) \leq C_t \quad t = 1, \dots, T \quad (3)$$

$$\sum_{i=1}^n w_{it} \leq 1 \quad t = 2, \dots, T \quad (4)$$

$$w_{it} \leq y_{i,t-1} + w_{i,t-1} \quad i = 1, \dots, n, \quad t = 1, \dots, T \quad (5)$$

$$w_{i,t+1} + w_{it} \leq 1 + z_t \quad i = 1, \dots, n, \quad t = 2, \dots, T \quad (6)$$

$$y_{it} + z_t \leq 1 \quad i = 1, \dots, n, \quad t = 1, \dots, T \quad (7)$$

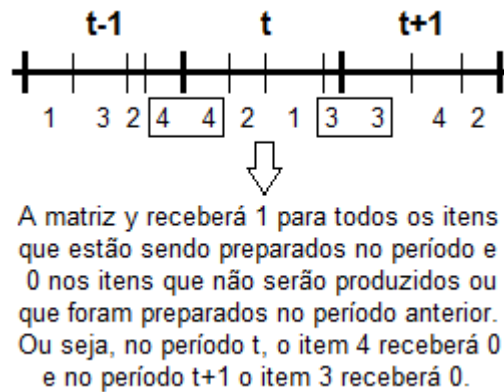
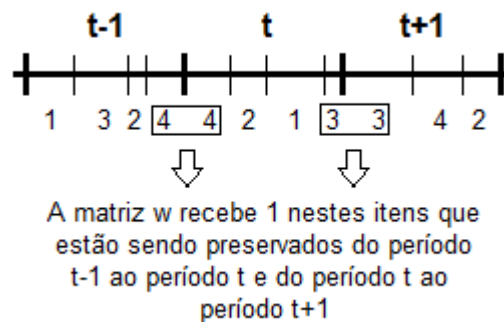
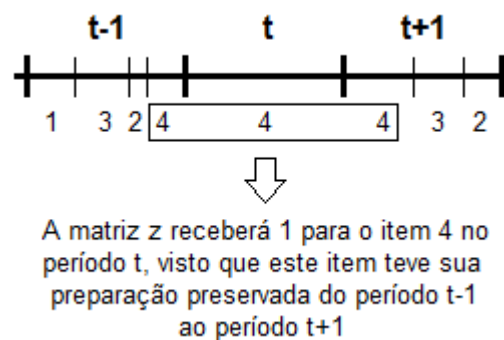
$$q_{it} \leq M_{it} (y_{it} + w_{it}) \quad i = 1, \dots, n, \quad t = 1, \dots, T \quad (8)$$

$$z_t \geq 0 \quad t = 1, \dots, T-1 \quad (9)$$

$$w_{it} \in \{0,1\} \quad i = 1, \dots, n, \quad t = 1, \dots, T \quad (w_{i1} = 0) \quad (10)$$

$$q \in \mathbb{R}_+^{nT}, \quad I \in \mathbb{R}_+^{nT} \quad \text{e} \quad y \in B^{nT} \quad (11)$$

As variáveis y , w e z estão exemplificadas nas Fig. 1, Fig. 2 e Fig. 3. Assim o leitor pode compreender melhor a diferença entre essas três variáveis. A diferença é bem pequena mas importante na montagem do problema. Cada variável é necessária para o estudo desse tipo de problema onde tem-se a preservação da preparação.

Figura 1: Matriz y Figura 2: Matriz w Figura 3: Matriz z

Com relação ao problema, a função objetivo (1) minimiza a soma dos custos de preparação da máquina e custos de estoque para todos os itens i em todos os períodos t . As equações (2) correspondem ao balanceamento do estoque. As restrições (3) limitam a capacidade do período no somatório dos tempos de preparação da máquina (setup) para todos os itens quando há produção de cada item no período e tempo de produção de cada item de acordo com a quantidade de itens a ser produzida no período. (4) garantem que a máquina só vai estar preparada para um item em cada período. As restrições (5), por sua vez, garantem

que se não há preparação do item i no período atual e houve mudança do estado de preparação da máquina para o item i do período anterior para o atual, a máquina também não estará preparada para este mesmo item i no período atual. As restrições (6) garantem que se há preparação de algum item no período t não pode haver um item i que tem sua preparação preservada de $t-1$ a t e também de t a $t+1$. (7) garantem que somente uma das duas coisas pode acontecer: preparação da máquina para o item i no período t e preservação da preparação de itens do período $t-1$ ao período $t+1$. As restrições (8) garantem que se não houve preparação da máquina para o item i no período t e também o estado de preparação do item i não foi preservado do período $t-1$ para o período t , então não pode haver produção naquele período para aquele item i . (9) asseguram que na minimização a variável z_t não assuma valores negativos. (10) define w_{it} como uma variável binária e (11) indica o tipo das demais variáveis.

3 HEURÍSTICA GREEDY-MOD

Uma solução para o modelo (1) - (11) será da forma $q = \begin{bmatrix} q_{11} & \cdots & q_{1T} \\ \vdots & \ddots & \vdots \\ q_{n1} & \cdots & q_{nT} \end{bmatrix}_{n \times T} = (q_{it})_{n \times T}$,

onde q_{it} é a quantidade produzida do item i no período t . Essa matriz está associada à matriz

solução $y = \begin{bmatrix} y_{11} & \cdots & y_{1T} \\ \vdots & \ddots & \vdots \\ y_{n1} & \cdots & y_{nT} \end{bmatrix}_{n \times T} = (y_{it})_{n \times T}$, onde y_{it} será 1 quando a máquina estiver

preparada para o item i no período t e 0 caso contrário. Neste trabalho a heurística Greedy-Mod que será explicada a seguir será aplicada nesta matriz y .

A Heurística Greedy-Mod, proposta por Fleischmann & Meyr (1997) monta a solução do problema de forma construtiva, iniciando no último período e terminando no primeiro período. Consiste em preencher as quantidades a serem produzidas com base no valor máximo da razão entre custo de estoque e tempo de produção, de forma a utilizar a capacidade disponível de cada período primeiramente nos itens que custará mais caro estocar. Se a capacidade não satisfizer a demanda, produzirá um produto que tem um custo de estoque mais barato em relação ao de produção, de forma a minimizar os custos. Esta heurística leva em conta micropériosodos e também não possui preservação da preparação. Portanto é necessária uma adaptação da mesma de forma a satisfazer tais restrições. Inicialmente será considerado sem a preservação da preparação, mas retirando os micropériosodos, contando somente períodos inteiros. Além disso, o Greedy-Mod contava com número de lotes por macropériodo. Como não será trabalhado com micropériodo, essa informação será retirada. Será considerado que, em um período que há produção, o mínimo a ser produzido é um e definindo a variável x_{it} e $\bar{C}_{it}$ por

$$x_{it} = \begin{cases} 1, & \text{se houve produção do item } i \text{ no período } t \\ 0, & \text{caso contrário} \end{cases}, \quad (12)$$

$\bar{C}_{it}$ = capacidade disponível em todos os períodos anteriores ao período t se o item i estiver sendo produzido neste período.

Dessa forma, temos que $\bar{C}_{it}$ é dada pela equação

$$\bar{C}_{it} = \left(\sum_{\tau=1}^{t-1} x_{it} \cdot C_{\tau} \right), \quad i=1, \dots, n; \quad t=1, \dots, T \quad (13)$$

Considerando essas variáveis que foram adicionadas e também as já definidas anteriormente, tem-se que o algoritmo do Greedy-Mod é dado pela tabela 3.1.

A primeira parte da heurística faz com que todas as quantidades sejam nulas e a demanda acumulada seja igual ao somatório das demandas de todos os períodos anteriores, para cada i . Na segunda parte o algoritmo vai completar cada quantidade com o mínimo a ser produzido, que determinamos em 1. Dessa forma, se há produção naquele período, também haverá uma quantidade a ser produzida. Caso não haja produção no período, esta quantidade será nula. Com isso atualiza-se também a capacidade e a demanda acumulada (demanda que ainda resta para ser atendida). A terceira parte, por sua vez, é a parte que Fleischmann & Meyr (1997) incluíram no algoritmo Greedy-Sim para construir o Greedy-Mod. Neste caso, se a demanda for maior do que a quantidade que pode ser produzida utilizando toda a capacidade dos períodos anteriores, coloca-se um valor mínimo de produção neste período de forma que consiga satisfazer a demanda posteriormente. Novamente atualiza a quantidade de produção, capacidade e demanda acumulada. Na última parte, através da razão $\frac{h_i}{b_i}$, o algoritmo

determina qual item é menos vantajoso estocar e altera sua quantidade de produção por primeiro. Aquele item que gera um menor prejuízo no caso de estocagem é deixado para ser avaliado por último e, inclusive, pode acabar nem sendo produzido por conta do limite de capacidade. Assim o algoritmo estará minimizando o custo de produção e estoque, fazendo com que seja produzido no período o que estocar nos traz um custo muito alto.

Esta heurística não está adaptada para preservação da preparação, ela somente distribui as quantidades a serem produzidas conforme o custo de estoque, tempo de produção, demanda e capacidade do período. Para adaptá-la algumas mudanças teriam que ser feitas, para que em cada período que haja preservação o tempo de preparação não seja computado.

Tabela 1: Greedy-Mod

----- Primeira Parte -----
Inicie:
$q_{it} = 0; \forall i, t$
$D_i = \sum_{t=1}^T d_{it}; \forall i$
----- Segunda Parte -----
Para $t = T$ até 1 faça:
Para todo i com $(x_{it} > 0)$ faça:

$$q_{it} = x_{it}$$

$$C_t = C_t - b_i q_{it}$$

$$D_i = D_i - \min \left\{ q_{it} ; D_i - \sum_{\tau=1}^{t-1} d_{i\tau} \right\}$$

Se $C_t < 0$ então pare;

----- Terceira Parte -----

Para todo i com $(x_{it} > 0)$ faça:

$$\text{Se } \left(D_i > \frac{\bar{C}_{it}}{b_i} \right) \text{ então}$$

$$\text{diff} := \min \left\{ D_i - \frac{\bar{C}_{it}}{b_i} ; \frac{C_t}{b_i} ; D_i - \sum_{\tau=1}^{t-1} d_{i\tau} \right\}$$

$$q_{it} = q_{it} + \text{diff}$$

$$C_t = C_t - \text{diff} \cdot b_i$$

$$D_i = D_i - \text{diff}$$

----- Quarta Parte -----

Para todo i com $(x_{it} > 0)$ na ordem de decrescimento de $\frac{h_i}{b_i}$ faça:

$$\text{diff} := \min \left\{ D_i - \sum_{\tau=1}^{t-1} d_{i\tau} ; \frac{C_t}{b_i} \right\}$$

$$q_{it} = q_{it} + \text{diff}$$

$$C_t = C_t - \text{diff} \cdot b_i$$

$$D_i = D_i - \text{diff}$$

Fim.

4 OTIMIZAÇÃO POR PARTICLE SWARM OPTIMIZATION

4.1 Introdução

A técnica de otimização por Particle Swarm Optimization (PSO) é uma metaheurística criada há 20 anos, por James Kennedy, Russell Eberhart e Frank Heppner. Este método analisa o comportamento de enxames, o comportamento social de pássaros e peixes, e adapta à matemática na forma de uma metaheurística de população. Baseando-se na individualidade e sociabilidade, cada partícula possui um peso para se espelhar em suas melhores posições e também nas melhores posições do grupo inteiro, fazendo com que a otimização seja mais precisa e não caia diretamente em mínimos locais como aconteceria se fosse baseada somente na individualidade. Já a sociabilidade gera diversificação.

A velocidade de uma partícula é dada por três fatores: termo de inércia, termo cognitivo e termo de aprendizado social. Ao atualizar uma partícula deve-se avaliar se a solução encontrada é melhor ou pior que a anterior (melhor solução da partícula é chamada de $pbest$) e também se é melhor ou pior que a melhor solução encontrada até o momento (melhor solução de todas as partículas é chamada de $gbest$). Para fazer essa avaliação deve-se calcular o fitness da solução. No caso do problema de Planejamento de Produção, o fitness é a própria função objetivo. Em outros problemas avalia-se se deve maximizar ou minimizar o objetivo. Este objetivo será o fitness. Cada termo da velocidade da partícula determina um fator na equação da velocidade. O primeiro termo é dado pela inércia, o segundo é o termo cognitivo e o terceiro é o termo de aprendizado social. A atualização da velocidade pode ser feita pela seguinte equação:

$$v_i' = v_i w + \eta_1 \cdot rand_1 \cdot (pbest_i - x_i) + \eta_2 \cdot rand_2 \cdot (gbest_i - x_i) \quad (14)$$

Com a atualização da velocidade pode-se atualizar a solução, da forma:

$$x_i' = x_i + v_i \quad (15)$$

Lembrando que:

η_1 e η_2 são parâmetros

$rand_1$ e $rand_2$ são valores randômicos entre 0 e 1

$pbest_i$ é a melhor posição que a partícula i já esteve

$gbest_i$ é a melhor posição que algum vizinho de i já esteve

w é o peso da inércia

As aplicações do PSO são imensas em problemas de sequenciamento (Flow Shop e Job Shop) e também roteamento de veículos, problema do Caixeiro Viajante, entre outros que determinam uma sequência a ser seguida. Essas aplicações utilizam o PSO contínuo e a velocidade seria troca de posições. No caso de Planejamento de Produção, em que não há um sequenciamento, a aplicação já deve ser alterada para um PSO Discreto, onde é mais utilizado o vetor binário como solução. Neste trabalho a velocidade assumirá valores entre -1 e 1.

4.2 O algoritmo utilizado

A Figura 4 traz um fluxograma que mostra como o algoritmo funciona e abaixo está explicado cada detalhe.

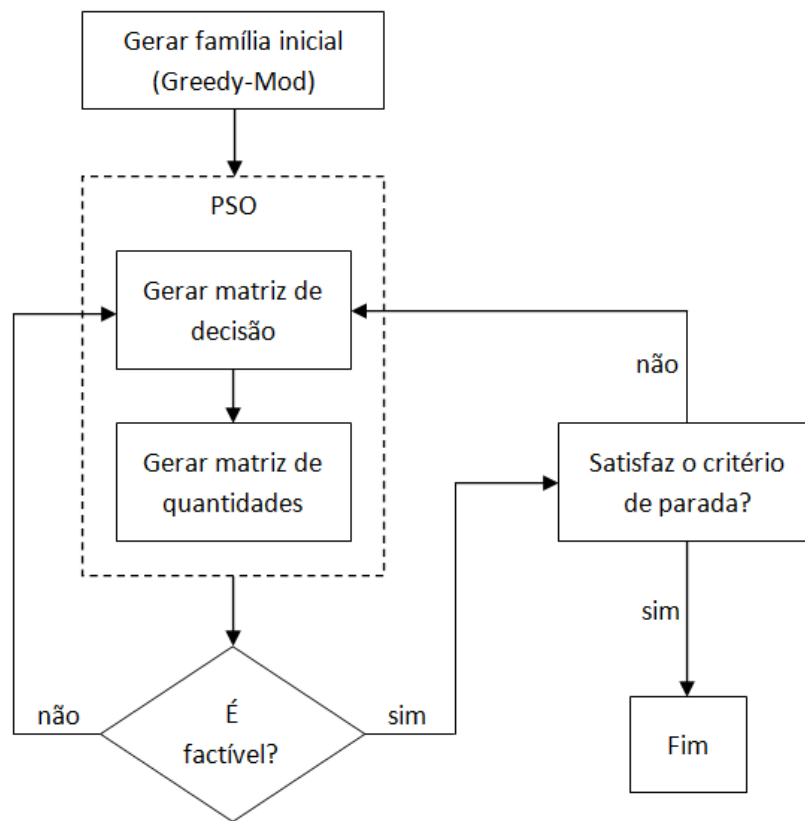


Figura 4: Fluxograma

A população inicial será gerada a partir de quatro matrizes diferentes. Essas quatro maneiras foram baseadas na heurística Greedy-Mod, com algumas modificações. A primeira delas é preenchida de trás para frente (forward), observando o maior custo de estoque de cada item. A segunda preenchida da frente para trás (backward), observando o menor custo de estoque. Ou seja, essas duas matrizes dão preferência de produção àqueles itens que possuem um maior custo de estoque. A terceira matriz é preenchida de trás para frente (forward), observando o maior custo de setup e maior custo de estoque. A quarta também observa o maior custo de setup, mas agora com o menor custo de estoque e é preenchida da frente para trás (backward). Essas duas últimas matrizes considerarão a preservação da preparação, enquanto as duas primeiras ainda não. A partir de cada matriz já é gerada a matriz de decisão, que vai ser a matriz movimentada pela metaheurística. As entradas da matriz de decisão serão -1 no caso de não produção daquele item em um determinado período, 0 caso houve produção mas a máquina já esteve preparada (houve preservação da preparação) e 1 caso houve produção e houve também a preparação da máquina.

Uma solução para o modelo dado pelas equações de (1) - (11), como já descrita anteriormente, será da forma $q = \begin{bmatrix} q_{11} & \cdots & q_{1T} \\ \vdots & \ddots & \vdots \\ q_{n1} & \cdots & q_{nT} \end{bmatrix}_{n \times T} = (q_{it})_{n \times T}$, onde q é a matriz das quantidades produzidas de cada item i em cada período t . À essa matriz tem-se a matriz y associada, que descreve a decisão de produção. A variável y_{it} assumirá -1 caso o item i não

esteja sendo produzido no período t ; 0 caso esteja sendo produzido, mas a preparação da máquina foi preservada do período $t-1$ (ou seja, não foi preciso preparar a máquina no período t para produzir o item i) e 1 caso esteja sendo produzido e a máquina foi preparada para a

produção do item i no período t . A matriz y terá a forma $y = \begin{bmatrix} y_{11} & \dots & y_{1T} \\ \vdots & \ddots & \vdots \\ y_{n1} & \dots & y_{nT} \end{bmatrix}_{n \times T} = (y_{it})_{n \times T}$.

A matriz q de quantidades não pode ser a base para a aplicação da metaheurística pois gera muita infactibilidade. A maneira como foi tentado contornar o problema de infactibilidade foi utilizar a matriz de decisão.

A movimentação será realizada de acordo com o resultado da entrada da matriz de velocidades. Caso a entrada resulte em um valor menor que um valor negativo θ , se a entrada for 1 ou 0, ela será trocada para -1. Caso a entrada resulte em um valor maior que θ e menor que um valor positivo α , será trocada para 0. Caso resulte em um valor maior que α , será trocada para 1. O valor θ é um valor mais próximo de -1 do que α de 1 para que as infactibilidades sejam evitadas.

Após feita a atualização da velocidade deve-se encontrar novamente as matrizes y e q . Após a aplicação do PSO deve-se ser verificado se a matriz não tem preservação da produção no primeiro período e também preservação de um mesmo item por dois períodos consecutivos caso tenha produção de outro item no mesmo período.

Por exemplo, a matriz de decisão pode ser da forma $y = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & -1 & 1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$, ou seja, o item

1 é preparado e produzido nos períodos 1 e 2, e somente produzido (sem preparação pois esta já foi preservada do período 2) nos períodos 3 e 4. O item 2 é preparado e produzido nos períodos 1 e 4, somente produzido (sem preparação) no período 2 e não é produzido no período 3. O item 3 é preparado e produzido nos períodos 1 e 4 e não produzido nos períodos

2 e 3. A matriz $y = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 1 \\ 1 & -1 & 1 & 1 \end{bmatrix}$, por exemplo, deve ser movimentada pois o item 1 no

período 1 não poderia ter preservação, sendo que a máquina não está preparada para nenhum item no primeiro período. Dessa forma, a entrada deveria ser 1 (ou -1 caso não haja demanda). Também não pode haver dois zeros em períodos consecutivos caso tenha outro item sendo produzido. Ou seja, nos períodos 3 e 4 o item 1 foi preservado mas o item 3 foi produzido no período 3. Isso não pode acontecer pois se o item 3 foi produzido, deve-se haver preparação da máquina para o item 1 e não a preservação da preparação novamente. Neste caso também deve-se mudar a matriz de decisão de forma a evitar o problema.

A matriz de quantidades será gerada com o LINGO. Será retornado ao otimizador os valores das matrizes y , w e z , de forma que o modelo fique com somente variáveis reais, não tenha mais variáveis inteiras e/ou mistas. Assim a solução encontrada será a matriz de quantidades que apresenta uma melhor função objetivo para aquela matriz de decisão enviada ao otimizador.

É feito isso (aplicação do PSO e movimentação da matriz para evitar infactibilidades) até o critério de parada. Nesse problema o critério de parada foi definido por um número máximo

de iterações. Ao final, o melhor valor encontrado para a função objetivo (fitness) e a solução correspondente serão os valores a serem considerados como solução do problema.

Para estimar a confiabilidade da solução pode-se resolver o mesmo problema diversas vezes e calcular o percentual de acerto (para um problema com o valor ótimo conhecido). Quando tiver um percentual de acerto alto (considerando-se alto um percentual de acerto acima de 80%), pode-se aumentar o tamanho do problema de forma a encontrar uma solução boa (não necessariamente o ótimo, mas melhor ou igual à solução encontrada pelo otimizador utilizado).

A seguir está detalhado o passo a passo da programação do algoritmo.

Tabela 2: Programação do algoritmo

1. (Uma única vez)

Para a solução inicial:

- Definir a população inicial.
- Utilizar o Greedy-Mod para encontrar a matriz q .
- Modificar a matriz y de forma que seja a matriz de decisão correspondente à matriz de quantidades encontradas (com -1 onde não houve produção) - sem considerar a preservação da preparação.
- Gerar a população inicial trocando algumas entradas da matriz y que valiam 1 por 0, de forma a preservar a preparação daquele determinado item. (Serão geradas p matrizes, onde p é o número de indivíduos na população).

2. (Repete enquanto não satisfizer o critério de parada)

Para a movimentação de cada indivíduo*:

- Atualizar a velocidade.
- Atualizar a posição da partícula.
- Atualizar as matrizes y e q .
- Atualizar a função objetivo (fitness do indivíduo).
- Salva a solução atual, a melhor de todas (gbest) e a melhor do indivíduo (pbest).

4.3 Resultados Computacionais

A aplicação descrita foi programada em Visual Studio Professional 2013, Versão 12.0.30723.00 Update 3, em uma CPU com Intel Core i5 com duas placas de 2.27 GHz e 6,00 GB de memória RAM. O otimizador com o qual foi feita a comparação é um LINGO Win64 Versão 13.0.2.17 (2012).

A Tabela 2 demonstra o resultado computacional obtido até o momento para problemas pequenos e também um pouco maiores, para um problema de planejamento de produção com preservação da preparação da máquina (para a produção de um determinado item) de um período a outro.

Há uma dificuldade quando tem-se a preservação da preparação e isto gera muita infactibilidade para o problema. Dessa forma os resultados não foram positivos quando considerada a preservação. Ainda é estudado um meio de considerar a preservação da preparação e obter resultados melhores.

Tabela 3: resultados computacionais

Tamanho do Problema		Otimizador LINGO		PSO (Visual Basic)			
		Melhor solução (em um limite de 30')	Tempo (seg)	Melhor solução			Tempo (seg)
				Tamanho da população	Número de iterações	Melhor solução encontrada	
Itens	3	1585	0,0001	3	1000	1655	4
Períodos	4						
Itens	5	4805	1	5	5000	5105	78
Períodos	7						
Itens	7	14180	1800	7	15000	14910	300
Períodos	10						

Agradecimentos

Agradeço à CAPES pelo financiamento dos meus estudos e por me proporcionar a chance de estar cada vez adquirindo mais conhecimento no meu ramo, com a possibilidade de continuar a estudar após o término da graduação.

REFERENCES

Arenales, M., Armentano, V., Morabito, R., Yanasse, H., 2007. *Pesquisa Operacional*. Elsevier.

Chen, Y., Lin, J., 2009. A modified particle swarm optimization for production planning problems in the TFT Array process. *Expert Systems with Applications*, n. 36, pp. 12264–12271.

Fleischmann, B., Meyr, H., 1997. The general lotsizing and scheduling problem. *OR Spektrum*, n. 19, pp. 11-21.

Toledo, C. F. M., Lafitá, T. F. de Q., Arantes, M. da S., Oliveira, R. R. R. & França, P. M., 2010. Otimização por enxame de partículas aplicada ao problema geral de dimensionamento de lotes e programação da produção. *XXX Encontro Nacional de Engenharia de Produção*.