

Desenvolvimento de interface gráfica para o sequenciamento de produção

D. A. Bispo¹, E. Jesus², A. M. O. Júnior², J. A. Pacífico²

¹ Programa de Institucional de Bolsas de Iniciação Científica, Universidade Federal de Sergipe, 49100-000, São Cristóvão-Se, Brasil

² Programa de Pós-graduação em Engenharia Química, Universidade Federal de Sergipe, 49100, Saõ Cstistóvão-Se, Brasil.

edilsons@ufs.br

(Recebido em 24 de maio de 2011; aceito em 27 de julho de 2011)

O trabalho teve como objetivo desenvolver uma interface gráfica para o sistema GAMS (General Algebraic Modeling System) de modo a facilitar o acompanhamento de execução das tarefas planejadas em plantas de polpa de fruta. O uso de interface gráfica permite a visualização de variáveis de produção, por exemplo, do tempo término de cada tarefa. Desta forma, foi implementado modelo de sequenciamento de natureza MILP (Mixed Integer Linear Programming) e resolvido pelo sistema GAMS/CPLEX. O GAMS não possui interface amigável para a saída dos resultados. Assim, foi desenvolvida interface gráfica usando linguagem de programação JAVA para o sistema GAMS.

Palavras-chave: Sequenciamento de produção. Modelagem matemática. Interface gráfica. Otimização.

The study aimed to develop a graphical interface to the system GAMS (General Algebraic Modeling System) to facilitate monitoring of implementation of planned tasks in plants of fruit pulp, allowing the visualization of production variables, for example, the completion time of each task. A template for sequencing nature MILP (Mixed Integer Linear Programming) was implemented, which was solved by the system GAMS / CPLEX. The GAMS does not have friendly interface for output of results. Thus, the interface was developed using graphical programming language JAVA to the GAMS system.

Keywords: Sequencing production. Mathematical modeling. Graphical interface. Optimization.

1. INTRODUÇÃO

No processamento de frutas para a produção de polpas surge a necessidade de sistemas automatizados para gestão da produção, pois diversos sabores de polpas são disponibilizados no mercado e a depender da sequência que as quantidades de polpas são produzidas, têm-se custos de produção elevados em decorrência, por exemplo, do consumo de água e de energia, isto faz com que o sequenciamento das atividades seja fundamental para que se tenha uma produção otimizada.

O sequenciamento de produção em plantas flexíveis (plantas que produzem diferentes produtos por rotas similares) busca responder ao seguinte questionamento: “Qual melhor sequência de produção para manufaturar quantidades de polpas de fruta que não comprometa a data de entrega ao mercado e que leve a custos mínimos de produção?”. A resposta a essa indagação, justifica o investimento em pesquisa na área de otimização de processos produtivos.

Devido ao grande número de polpas a serem processadas, sendo que a produção de cada polpa é uma atividade, a qual é constituída por diversas operações, faz-se necessário a elaboração de um programa de produção que otimize a produção, através do uso de modelos e de uma interface que mostre graficamente o andamento da produção.

Neste trabalho, a interface foi desenvolvida fazendo um link com o software GAMS, o que permitiu rodar o modelo matemático, mostrando simultaneamente os resultados gerados de forma gráfica. Um sistema computacional integrado GAMS/Interface gráfica, tem como objetivo fornecer visão gráfica do programa de produção ótimo.

2. PROCESSO DE PRODUÇÃO DE POLPA DE FRUTAS

Define-se polpa de fruta como o produto não fermentado, não concentrado, não diluído, obtido de frutos polposos, através de processo tecnológico adequado, com um teor mínimo de sólidos totais, proveniente da parte comestível do fruto.

O Brasil é o maior produtor mundial de frutas, mas a produtividade é baixa, quando comparada com aquelas obtidas em países de tecnologia mais avançada, embora tenha aumentado significativamente nos últimos anos. Estimam-se potenciais de produtividade de 60% a 250% superiores às médias nacionais para as principais espécies frutíferas, caso as tecnologias disponíveis sejam adequadamente utilizadas (CARRARO & CUNHA, 1994).

O congelamento de polpa de fruta é um método de conservação que preserva as características da fruta e permite seu consumo nos períodos de entressafra. Esse processo possibilita ao produtor uma alternativa para a utilização de frutas que não atendam ao padrão de comercialização do produto *in natura*, ou cujos preços não sejam compensadores. (EMBRAPA, 2005)

O uso de frutas na forma de polpa congelada proporciona, também, a possibilidade de utilização de frutas pouco conhecidas, como as provenientes do Cerrado e das regiões Norte e Nordeste, que já despertam interesse no mercado externo. (*Op. Cit.*, 2005)

A produção de polpa de fruta congelada, antes concentrada somente na Região Nordeste, já se expandiu por todo o território nacional. É um segmento que, apesar de englobar grandes indústrias, está caracterizado pela presença de micro e pequenas empresas.

O processamento de polpas evita desperdícios e minimiza perdas que ocorrem com frequência durante a comercialização do produto *in natura*, agregando assim valor econômico a fruta.

Muitas vezes, pelo desconhecimento das boas práticas de fabricação, o processamento da polpa de fruta é conduzido sem os cuidados de higiene necessários, o que compromete a qualidade do produto final. Além, dessas boas práticas, faz-se necessário o uso de tecnologia, assim como sistemas computacionais que auxiliem a produção.

O fluxograma 1 abaixo descreve as etapas, de modo sucinto, do processo de produção da polpa de fruta.

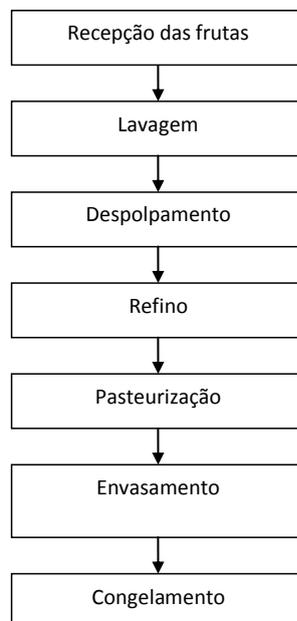


Figura 1 – Etapas do processo de produção de polpa de fruta

A programação de produção em plantas flexíveis, a exemplo de indústrias de produção de polpa, em regra, objetiva apresentar um programa de execução de tarefas, o qual otimize

determinadas função objetivo. Nesta direção, têm-se os trabalhos de Bing (1955); de Becceneri *et al.* (2006) e de Barros e Moccllin (2008).

Segundo Morais e Moccellin (2010),

A programação de produção desempenha um importante papel no auxílio ao cumprimento dos objetivos estratégicos da empresa e sua gestão da produção, na maioria das situações é conduzida de forma sistemática. A responsabilidade pelo planejamento, execução e controle das atividades realizadas no sistema de produção, ou seja, pela operacionalização do sistema, é do Planejamento e Controle da Produção (PCP).

Diversos modelos matemáticos têm sido usados na otimização da produção em diversos sistemas de manufatura (BECCENERI *et al.*, 2006; BARROS E MOCCLLIN, 2008; FERREIRA *et al.*, 2008).

Com o avanço da informática, alterou-se significativamente o modelo de controle de produção nas grandes indústrias, que passaram a adotar sistemas computacionais que melhor disponibiliza a informação em todos os seus setores, seguindo uma tendência de busca pela qualidade total na indústria (DEMING, 1990; CAMPOS, 1992; CROSBY, 1992). No entanto, esta realidade ainda não chegou às microempresas.

Quando se aborda a otimização de produção de qualquer indústria, de qualquer natureza, os seguintes objetivos são buscados: manufaturar produtos com melhor qualidade e aumentar a produtividade com menor custo unitário do produto.

O processo de manufatura assistida por computador, conhecida por CAM (Computer Aided Manufacturing) significa usar sistemas computacionais que auxiliem os engenheiros na tomada de decisão (KAI e KAH, 1997). O principal propósito da CAM é criar um processo de produção veloz e componentes com dimensões mais precisos de algum material para que seja usado só a quantidade de material requerido (minimizando o desperdício). A CAM é uma ferramenta de programação que permite manufaturar modelos físicos e ajuda também na tomada de decisão, plano operacional.

3. MODELO MATEMÁTICO DE SEQUENCIAMENTO DE PRODUÇÃO

O modelo proposto visa a otimização da função objetivo *makespan* (tempo de execução de todas as tarefas), dada pela equação 1.

$$Mk \geq TE(i, p) \quad \forall i \in T, \exists pm \in P \quad (1)$$

Onde:

Mk: makespan;

TE(i,pm): tempo de termino de uma tarefa do conjunto T no ultimo processador do conjunto de processadores P;

pm: último processador

As duas principais restrições de um modelo de programação de produção é que garante a não sobreposição de operações (não-coexistência de operações em determinado equipamento) e a restrição de precedência tecnológica, que determina o ordenamento das diversas tarefas pelos diversos processadores. A relação de precedência é dada pelas expressões 2 e 3.

$$TE(i, p) = TS(i, p) + TP(i, p) \quad \forall i \in T; p \in P \quad (2)$$

$$TS(i, p+1) \geq TE(i, p) \quad \forall i \in T; p \in P \quad (3)$$

Onde:

T: conjunto de tarefas;

P: conjunto de processadores;

TE(i,p): tempo de termino de uma tarefa do conjunto T em um processador do conjunto p;

TS(i,p): tempo de início de uma tarefa em um processador;

TP(i,p): tempo de processamento de uma tarefa em um processador.

Para a não sobreposição de tarefas no processador p, tem-se as equações 4, 5 e 6:

$$TS(t, p) - TE(i, p) \geq U * (1 - Z(i, t, p)) \quad \forall i \in T; p \in P; \forall t \in TL \quad (4)$$

$$TS(t, p) - TE(i, p) \leq U * Z(i, t, p) \quad \forall i \in T; p \in P; \forall t \in TL \quad (5)$$

$$Z(i, t, p) = 1 - Z(i, t, p) \quad \forall i \in T; p \in P; \forall t \in TL \quad (6)$$

Onde:

T, TL: conjunto de tarefas;

P: conjunto de processadores;

U: Valor escalar grande de controle;

TE(i,p): tempo de termino de uma tarefa do conjunto T em um processador do conjunto p;

TS(i,p): tempo de início de uma tarefa em um processador;

Z(i,t,p): variável inteira (0 ou 1) de precedência de tarefas no processador.

4. INTERFACE GRÁFICA PROPOSTA E RESULTADOS

Segundo Deitel (2005),

Uma interface gráfica (graphical user interface – GUI) apresenta um mecanismo amigável ao usuário para interagir com um aplicativo. Uma GUI (pronuncia-se ‘gui’) dá ao aplicativo uma ‘aparência’ e um ‘comportamento’ distintos. Fornecer aos diferentes aplicativos componentes de interface com o usuário consistentes e intuitivos permite, de certa maneira, que ele se familiarize com um aplicativo, para que possa aprende-lo mais rapidamente e utiliza-lo com mais produtividade.

Neste trabalho, a interface permitirá a entrada de dados para o modelo de programação de produção, a saída da carta do programa de produção (carta de Gantt).

A linguagem de programação a ser empregada no interfaciamento com o software GAMS foi JAVA. Foi desenvolvida uma interface amigável utilizando como apoio as bibliotecas JFreeChart e JCommon, que disponibilizam a estrutura para montagem do gráfico em barras.

A linguagem JAVA foi desenvolvida em 1990, pela SUN Microsystems. É uma linguagem de alto nível orientada a objetos e multi-plataforma, roda em qualquer dispositivo independentemente do sistema operacional. Para utilizar essa linguagem são necessárias as ferramentas JDK (Java development Kit), que é um pacote com as principais classes e um compilador, e uma IDE (Integrated Development Environment), um ambiente integrado para desenvolvimento de software, este pode ser substituído por um bloco de notas. O JDK utilizado no trabalho foi a versão 6 update 23 e a IDE foi o Eclipse.

Programas JAVA consistem em partes chamadas de classes. As classes incluem partes chamadas de métodos que realizam tarefas e retornam informações ao concluir. Nas bibliotecas de classe JAVA (APIs – application programming interface). Existem ricas coleções de classes, as quais podem ser oferecidas pelos fornecedores de compilador ou por fornecedores de softwares independentes (ISV – independente software vendors).

Ao executar o modelo de seqüenciamento no GAMS, há um retorno de um arquivo no formato “.txt” que possui os valores das variáveis TS (Tempo de Início), TE (Tempo de Término) e P (Processador) para cada Tarefa T do sequenciamento. A classe LerArquivo.java

tem como papel capturar o arquivo .txt e formatá-lo transformando os dados em informação visual (gráficos).

A Main.java é a responsável por criar a interface gráfica, é nela que está presente as bibliotecas de construção de Gráficos descritas acima. Nela são instanciados processadores e tarefas com seus respectivos tempos de acordo com as informações coletadas e formatadas do arquivo.txt.

Durante o tempo de execução da aplicação, uma rotina é acionada. Essa rotina faz com que o GAMS seja executado e após a execução é retornado o arquivo .txt que é utilizado para construção da interface com descrito anteriormente. O resultado da execução é o gráfico em forma de barras (barChart), que possui os processadores as tarefas e legenda colorida para melhor visualização.

Foi desenvolvido um meio de comunicação entre o GAMS e a interface de forma automática, sem a intervenção do usuário.

A interface foi dividida em três módulos, como podemos observar na figura 2, propiciando ao usuário uma melhor usabilidade. No primeiro módulo, o usuário entra com os dados que serão utilizados para a construção do gráfico. O segundo irá executar todo o processamento dos dados, é nele que será gerado o arquivo .txt como já foi ressaltado. No terceiro será exibido o gráfico, como mostrado na figura 3.

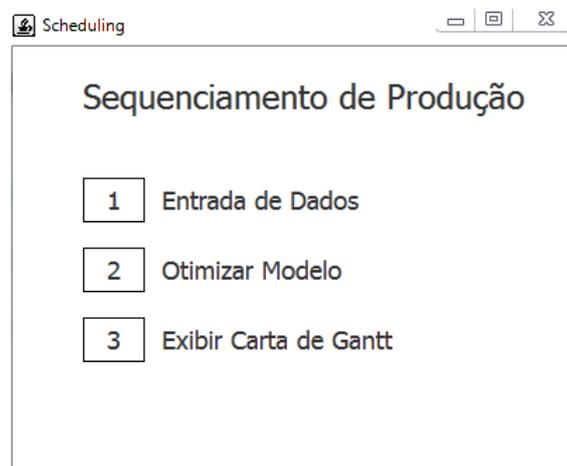


Figura 2 – Interface dividida em três módulos

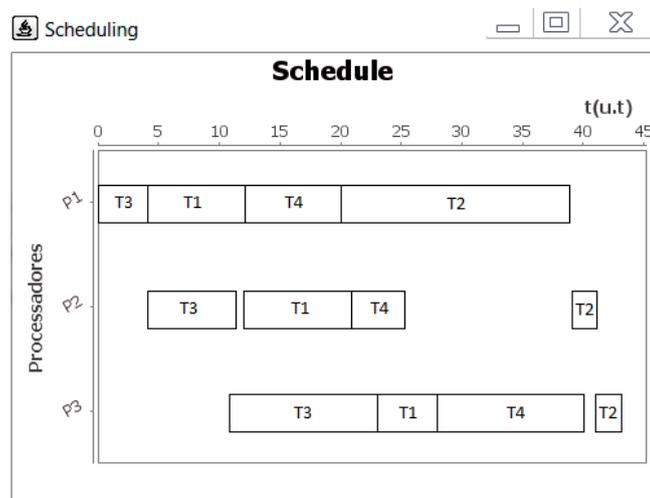


Figura 3 – Resultado do Sequenciamento

A figura3 mostra o programa de produção (Schedule) gerado pela interface após a execução do modelo.

Os dados de entradas (tempos de processamento) do modelo GAMS serão fornecidos conforme Quadro 1. Estes tempos serão fornecidos ao modelo pela interface gráfica.

Quadro 1 – Tempos de processamento

	P1	P2	P3	P4	P5
T1	8	9	5	4	23
T2	4	4	7	12	2
T3	7	7	12	5	16
T4	3	5	12	5	7
T5	8	3	6	8	10
T6	1	8	4	2	16
T7	12	5	6	5	9

O modelo ao ser executado pelo GAMS fornece como saída um arquivo no formato txt mostrando os tempos de início e término de cada tarefa em cada máquina (figura 4).

```

t1( 14.0000, 22.0000, P1 )
t1( 22.0000, 31.0000, P2 )
t1( 39.0000, 44.0000, P3 )
t2( 0.0000, 4.0000, P1 )
t2( 4.0000, 8.0000, P2 )
t2( 8.0000, 15.0000, P3 )
t3( 7.0000, 14.0000, P1 )
t3( 14.0000, 21.0000, P2 )
t3( 27.0000, 39.0000, P3 )
t4( 4.0000, 7.0000, P1 )
t4( 8.0000, 13.0000, P2 )
t4( 15.0000, 27.0000, P3 )

```

Figura 4 - Dados de saída do modelo

Depois de processar os dados de entrada, um arquivo .txt é criado quando o comando file out/out.txt; é executado. Depois de criado, os dados processados são inseridos no formato acima, onde para cada tarefa é informado o tempo de início e o tempo de termino em cada processador. A interface gráfica lê o arquivo formato “txt” para gerar o gráfico de sequenciamento.

5. CONCLUSÃO

Foi desenvolvida uma interface gráfica utilizando linguagem de programação JAVA, como podemos observar na figura 3, proporcionando ao usuário facilidade de interação com a aplicação.

O modelo MLIP foi implementado no software GAMS, utilizando as expressões de 1 a 6. Assim, ao final do processamento do GAMS têm-se uma solução ótima para os dados de entrada fornecidos através da interface.

O meio de comunicação entre o GAMS e a interface foi desenvolvido utilizando como base a plataforma JAVA, com isso foi possível utilizar toda eficácia do GAMS no processamento unido a interface amigável que o JAVA propicia, de forma automática sem a intervenção do usuário.

Como retorno do processamento da aplicação, é exibido um gráfico de fácil entendimento que simplifica a interpretação dos dados, como podemos observar na figura 2.

A aplicação foi desenvolvida visando desempenho na solução do problema e fácil interação com o usuário.

-
1. BARROS, Alexandre Damas e MOCCELLIN, João Vitor Análise da flutuação do gargalo em flow shop permutacional com tempos de setup assimétricos e dependentes da sequência. **Gest. Prod.**, Abr 2008, vol.11, no.1, p.101-108. ISSN 0104-530X
 2. BECCENERI, J. C.; YANASSE, H. H.; SOMA, N. Y. A method for solving the minimisation of the maximum number of open stacks problem within a cutting process. **Computers and Operations Research**, v. 31, n. 14, p. 2315-2332, 2006.
 3. BING, Wu Xiau. **Programa de produção e do transporte em sistemas flexíveis de manufatura**. Tese de doutorado. 1995. Tese (Doutorado em Engenharia Elétrica). Programa de Pós-graduação em Engenharia Elétrica. Universidade Estadual de Campinas, 1995.
 4. BROOKE,A.; KENDRICK,D.; MEERAUS,A.; RAMAN,R. **GAMS-A user's guide** Washington: GAMS Development Corporation 276p, 1998
 5. CAMPOS, Vicente Falconi. **TQC – Controle da qualidade total**, 8. ed. Belo Horizonte: DG, 1992.
 6. CARRARO, A.F.; CUNHA, M.M. **Manual de exportação de frutas**. Brasília: MAARA / SDR / FRUPEX / IICA, 1994. 254p.
 7. CROSBY, Philip B., **Qualidade sem lágrima: a arte da gerência descomplicada**. Rio de Janeiro: José Olímpio, 1992.
 8. DEITEL, H.M., **Java: como programar**, 6. ed. São Paulo Pearson Prentice Hall, 2005.
 9. DEMING, W. E., **Qualidade: a revolução da administração**. Ed Marques Saraiva, 1990.
 10. EMBRAPA. Ministério da Agricultura, Pecuária e Abastecimento. **Polpa de fruta congelada**. Agroindústria familiar. Coleção, 2005.
 11. FERREIRA, Deisemara; MORABITO, Reinaldo e RANGEL, Socorro. Um modelo de otimização inteira mista e heurísticas *relax and fix* para a programação da produção de fábricas de refrigerantes de pequeno porte. **Prod.** [online]. 2008, vol. 18, no. 1, pp. 76-88. ISSN 0103-6513.
 12. KAI, C.Chee Kai; KAH F. **Rapid Prototyping: Principles an Applications and Manufacturing**. John Wiley and Sons, 1997
 13. MORAIS, M. de F. ; MOCCELLIN, J. V. Métodos heurísticos construtivos para redução do estoque em processo em ambientes de produção *flow shop* híbridos com tempos de *setup* dependentes da sequência. **Gest. Prod.** vol.17 no.2 São Carlos 2010.