Volume : 2 | Issue : 4 | April 2013 • ISSN No 2277 - 8160

Research Paper | Mathematics

Dynamic Optimization in Production Scheduling of Batch Chemical Industry

Mr. M. Devarajan — Assistant Professor, Department of Mathematics, SACS MAVMM, Engineering College, Madurai 625 301, Tamilnadu, India.

Dr. Mohamed Sheriff — Associate Professor, Department of Mathematics, H.K.RH.College, Uthamapalayam, Theni District, Tamilnadu State, India.

ABSTRACT

Dynamic optimization of batch processes chemical industries has more attention in recent years because of growing contest, it is a accepted choice for reducing cost of production , quality improvement , and meeting safety necessities and environmental regulations. Since the models currently available in agrochemical industry are poor and face a large sum of uncertainty, standard model-based optimization techniques are by and large in effective, and optimization methods need to be depending on measurements. A new framework is also presented, where important characteristics of the optimal solution that are invariant under uncertainty are identified and improve the feedback control scheme. Thus, optimality is achieved by tracking. When only batch-end measurements are available, the proposed method leads naturally to an efficient batch-to-batch optimization scheme. The approaches are analyzed with data available from the Lanxess chemical industry and explain its importance of improving the chemicals process.

KEYWORDS: Agrochemical, Optimization, Production, Reducing, Simulation, Techniques.

1. Introduction

Scheduling is an important tool for manufacturing and engineering, where it can have a major impact on the productivity of a process. In manufacturing, the purpose of scheduling is to minimize the production time and costs, by telling a production facility when to make, with which staff, and on which equipment. Production scheduling aims to maximize the efficiency of the operation and reduce costs.

Production scheduling tools greatly outperform older manual scheduling methods. These provide the production scheduler with powerful graphical interfaces which can be used to visually optimize real-time work loads in various stages of production, and pattern recognition allows the software to automatically create scheduling opportunities which might not be apparent without this view into the data. For example, an airline might wish to minimize the number of airport gates required for its aircraft, in order to reduce costs, and scheduling software can allow the planners to see how this can be done, by analyzing time tables, aircraft usage, or the flow of passengers.

Companies use backward and forward scheduling to altlocate plant and machinery resources, plan human resources, plan production processes and purchase materials. Forward scheduling is planning the tasks from the date resources become available to determine the shipping date or the due date.

Backward scheduling is planning the tasks from the due date or required-by date to determine the start date and/or any changes in capacity required.

- ✓ The benefits of production scheduling include:
- ✓ Process change-over reduction
- ✓ Inventory reduction, leveling
- ✓ Reduced scheduling effort
- ✓ Increased production efficiency
- ✓ Labor load leveling
- ✓ Accurate delivery date quotes

Batch production scheduling is the practice of planning and scheduling of batch manufacturing processes. See Batch production. Although, scheduling may apply to traditionally continuous processes, such as refining, it is especially important for batch processes such as those for pharmaceutical active ingredients, biotechnology processes and many specialty chemical processes. Batch production scheduling shares some concepts and techniques with finite capacity scheduling which has been applied to many manufacturing problems.The specific issues of scheduling batch manufacturing processes have generated considerable industrial and academic interest.

Scheduling in the Batch Processing Environment. A batch process can be described in terms of a recipe which comprises a bill of materials and operating instructions which describe how to make the product In mathematics, computer science, and economics, dynamic programming is a method for solving complex problems by breaking them down into simpler subproblems. It is applicable to problems exhibiting the properties of overlapping sub problems [1] and optimal substructure (described below). When applicable, the method takes far less time than naive methods.

The key idea behind dynamic programming is quite simple. In general, to solve a given problem, we need to solve different parts of the problem (subproblems), and then combine the solutions of the subproblems to reach an overall solution. Often, many of these subproblems are really the same. The dynamic programming approach seeks to solve each subproblem only once, thus reducing the number of computations: once the solution to a given subproblem has been computed, it is stored or "memo-ized": the next time the same solution is needed, it is simply looked up. This approach is especially useful when the number of repeating subproblems grows exponentially as a function of the size of the input.

Dynamic programming in mathematical optimization

In terms of mathematical optimization, dynamic programming usually refers to simplifying a decision by breaking it down into a sequence of decision steps over time. This is done by defining a sequence of value functions V1, V2, ..., Vn, with an argument y representing the state of the system at times i from 1 to n. The definition of Vn(y) is the value obtained in state y at the last time n. The values Vi at earlier times i = n −1, n − 2, ..., 2, 1 can be found by working backwards, using a recursive relationship called the Bellman equation. For i = 2, ..., n, Vi−1 at any state y is calculated from Vi by maximizing a simple function (usually the sum) of the gain from decision i − 1 and the function Vi at the new state of the system if this decision is made. Since Vi has already been calculated for the needed states, the above operation yields Vi−1 for those states. Finally, V1 at the initial state of the system is the value of the optimal solution. The optimal values of the decision variables can be recovered, one by one, by tracking back the calculations already performed.

Real time information hese guidelines include complete descriptions of the fonts, spacing, and related information for producing your proceedings manuscripts. Please follow them and if you have any questions, contact us through http://enathisky.org./ieicsms2013/

2. Dynamic programming in mathematical optimization

In terms of mathematical optimization, dynamic programming usually

refers to simplifying a decision by breaking it down into a sequence of decision steps over time. This is done by defining a sequence of value functions V1, V2, ..., Vn, with an argument y representing the state of the system at times i from 1 to n. The definition of Vn(y) is the value obtained in state y at the last time n. The values Vi at earlier times i = n −1, n − 2, ..., 2, 1 can be found by working backwards, using a recursive relationship called the Bellman equation. For i = 2, ..., n, Vi−1 at any state y is calculated from Vi by maximizing a simple function (usually the sum) of the gain from decision i − 1 and the function Vi at the new state of the system if this decision is made. Since Vi has already been calculated for the needed states, the above operation yields Vi−1 for those states. Finally, V1 at the initial state of the system is the value of the optimal solution. The optimal values of the decision variables can be recovered, one by one, by tracking back the calculations already performed.

3. Batch chemical

The Batch chemical plants are classified as either multiproduct or multipurpose (Rippin, 1983). Multiproduct batch chemical plants are analogous to flow shops in discrete manufacturing where all products follow the same sequence of operations (a single process path). Multipurpose batch chemical plants are analogous to job shops with alternative routing in discrete manufacturing where products may follow different process paths. The focus of this paper is on multipurpose batch chemical plants.

4. Abut LANXESS

LANXESS is a major supplier of phosphorous compounds. Starting from elemental phosphorous, LANXESS produces a variety of derivatives which can be used in numerous application fields such as flame retardance or as intermediates. Excellent upstream integration, experience of many decades and modern plants provide a basis for economic production and top-level quality. The product portfolio of LANXESS includes the phosphorous-based product lines Disflamoll and Levagard, which offer highly efficient fire protection and other advantages such as easy processability or high elasticity.

Under the proprietary name Bayfomox, LANXESS offers a reactive polyurethane system that is used to manufacture extremely flame-retardant articles for the prevention of fire spreading Phosphorous Chemicals

LANXESS is one of the main producers of organophosphorous compounds which can be used as:

- synthetic chemicals
- flame retardants
- water treatment additives
- specialties
- lubricant and oil additives

Our broad product range is the result of a committed cooperation between Research and Development, Marketing and Manufacturing. LANXESS phosphorous compounds are manufactured to the stringent DIN ISO 9001.

Synthesis Chemicals

LANXESS is one of the world's leading producers of Phosphorous trichloride (PCl3) and of its downstream products alkyl phosphites such as Dimethyl phosphite (DMPI), Diethyl phosphite (DEPI) and Triethyl phosphite (TEPI) to name some of the most important derivatives.

These products together with Phosphorous oxychloride (P0Cl3) are of high purity and may therefore be used in different chemical reactions

Flame Retardants

Current trends in building and automotive industries, as well as in the electric and electronic sectors, reflect a strong increase of flame retardant consumption, world wide.

LANXESS is likewise responding with an attractive range of flame retardants for these industries.

Water Treatment Chemicals

Phosphonates based on a C-P bond show different properties for different application areas.

Bayhibit® - products based on the active substance 2-Phosphonobutane-1,2,4-tricarbonicacid (PBTC), which was developed by LANXESS, are used as scale- and corrosion inhibitors.

Based on the outstanding stability to bleach, lye, chlorine and bromine and their optimal environmental profile, Bayhibit ® products can be used as additives in cleaning agents, for cooling and processing water. This success is based on the high quality which we can guarantee with our longstanding production track record.

Specialties

LANXESS is the leading producer of phosphorous compounds, characterized by their high purity. This applies especially to dialkyl phosphates. Based on their variety of properties, they can be used as defoaming agents like Tributyl phosphate, or as extraction agents for the liquid-liquid extraction like.

Our products can be also used as processing aids, plasticizers, viscosity depressors and catalysts in chemical synthesis.

Lubricant and oil additives

Lubricants are high performance products. They are tailored blends of synthetic base stocks and performance additives.

LANXESS offers a range of products based on phosphorous, which are added to synthetic base stock to confer properties that are necessary for lubrication which do not exist in the base stock uthor names are to be centered beneath the title and printed in Times 12-point, boldface type. In the case of multiple authors, affiliations are in two- or three- or more column format and italicized, not bold. Include e-mail addresses if possible. Follow the author information by two blank lines before main text.

5. The lower bound

The idea of lower bound determination Notations: i ¼ 1,y,m — machine identifier, and

m is the number of machines;
j = 1…n — Chemical batch identifier,
and n is the number of batches;
k =1….n — sequence position index;
and
Pij —the processing time of the jth chemical batch on ith machine.

Assume that there is a flowshop problem including n chemical batches and m machines with setup times.Suppose that the branch of the pth level is one of the sub problems corresponding to the (p+1)th level. For any node N_{p+1} in the (p+1)th layer (p = 0,1,….n-2), the sequence of (p+1) chemical batches is determined. J_{p+1} represents this partial sequence and U_{p+1} the set of the remaining (n-p-1) chemical batches whose sequence has not been determined.

If LB1(Jp+1) is defined as a lower bound to the objective function for the node Np+1,

$$LB1\ (J_{p+1}) = C_m\ (J_{p+1}) + \sum_{j \in U_{p+1}} P_{jm} + Z_{(p+1)\,m.}$$

It consists of the following three parts:

(1) Cm (Jp+1) — the maximal completion time of the mth machine for the partial sequence Jp+1.

(2) P j2Upþ1 Pm;j — the sum of the processing times for the remaining (n p1) rolling batches whose sequence has not been determined on a machine m.

(3) TZ(p+1), m — it is associated with the setup time defined as a lower bound to the sum of the setup times which combine the setup time for the last rolling batch in the partial sequence Jp+1 and the sum of the setup times for the remaining rolling batches whose sequence has not been determined. In the above formula, it is easy to obtain the first two items. The key to the problem is to determine the third one. In the following, we introduce the transformation method of lower bound to obtain item (3).

6. The implementation of the branch-and-bound algorithm

The implementations of the branch-and-bound algorithm the related notations concerning the algorithm are defined before the steps of the

algorithm are described. P ¼ the number of branching eve

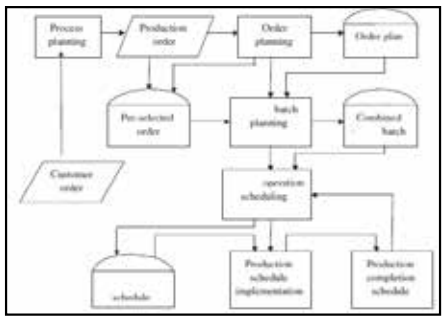

Fig.1.Scheduling Process

ls; C* ¼ the current optimal objective function value;

P* ¼ the rolling batch sequence corresponding to C*; and P1 ¼ the partial sequence corresponding to the current node (the node needs branching now).

Step 1: Initialization. Let p ¼ 0, P1 ¼ f (empty set), C* ¼ N. Assume that the current node always corresponds to P1. In this way the root node acts as the current node.

Step 2: Calculate the lower bound at each node branching from the current node and sequence the subnodes according to the values of the lower bounds from minimum to maximum. Let p ¼ p+1.

Step 3: If the current node has been fully searched, then let p ¼ p 1; go to Step 6. Otherwise, suppose that node Q is the active node of the current node with the minimum lower bound, and then the rolling batch in the pth position corresponding to the node Q is added at the end of P1. Next, the node Q becomes the current node; go to Step 4.

Step 4: Because the current node is the node with the minimum lower bound, which is chosen from the nodes of the same layer with the same parent node, it is unnecessary to search other nodes of the same layer with the same parent node if the lower bound at the current node is greater than or equal to C*. Thus, the upper layer node (parent node) of the current node has been detected completely. Let p ¼ p 1, eliminate the last rolling batch in P1 and go to Step 6; otherwise, go to Step 5.

Step 5: If p ¼ n, then a relatively good sequence is obtained. Let P* ¼ P1, C* acts as a lower bound at the current node. Let p ¼ p 1, remove the last rolling batch in P1 and go to Step 6; otherwise, go to Step 2.

Step 6: If p6¼0, eliminate the last rolling batch in P1 and go to Step 3; otherwise, the algorithm stops. C* is the optimal value of the objective function and C* corresponds to the optimal sequence. ,

7. Particle Swarm optimization

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling.

PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles.

Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. This location is called lbest. When a particle takes all the population as its topological neighbors, the best value is a global best and is called gbest.

The particle swarm optimization concept consists of, at each time step, changing the velocity of (accelerating) each particle toward its pbest and lbest locations (local version of PSO). Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward pbest and lbest locations. In past several years, PSO has been successfully applied in many research and application areas. It is demonstrated that PSO gets better results in a faster, cheaper way compared with other methods.

Another reason that PSO is attractive is that there are few parameters to adjust. One version, with slight variations, works well in a wide variety of applications. Particle swarm optimization has been used for approaches that can be used across a wide range of applications, as well as for specific applications focused on a specific requirement indicate full details of any application software used in the submission.

8. Application in chemical process scheduling of PSO

ICPSOm	=	1297
CPSOm	=	1297
ICPSOminm	=	1297
ICPSOmaxm	=	1297
ICPSOaverage	=	1297
ICPSOstd	=	0
CPSOminm	=	1297
CPSOmaxm	=	1297
CPSOaverage	=	1297
CPSOstd	=	0

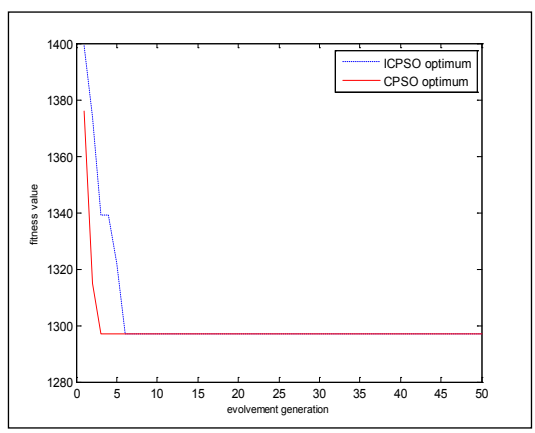

Fig.2. Out put for 80 No of Machines

ICPSO	=	1278
CPSOm	=	1297
ICPSOminm	=	1278
ICPSOmaxm	=	1297
ICPSOaverage	=	1297
ICPSOstd	=	10.9697
CPSOminm	=	1297
CPSOmaxm	=	1297
CPSOaverage	=	1297
CPSOstd	=	0

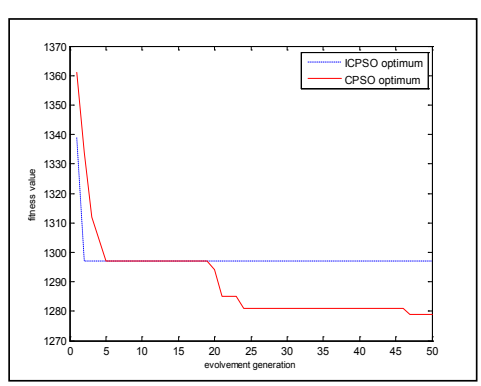

Fig.3. Out put for 90 No of Machines

Fig.4. Out put for 100 No of Machines

ICPSOm	=	1297
CPSOm	=	1297
CPSOminm	=	1297
CPSOmaxm	=	1297
ICPSOaverage	=	1297
ICPSOstd	=	0
CPSOminm	=	1279
CPSOmaxm	=	1297
CPSOaverage	=	1291
CPSOstd	=	10.3923

9. Conclusion

The batch chemical process in Laxness chemical Industry has been studied. Particle swam optimization is applied for the 80,90,100 machine are scheduled. The standard deviation found zero that means the convergence taken place in the optimization on chemical process. The time is taking as main factor with out varying quantity and quality of the chemical. This can extended with practical data as well

REFERENCES

[1] M. Dawande, J. Kalagnanam, H. S. Lee, C. Reddy, S. Siegel, and M. Trumbo, "The slab design problem in the steel industry," Interfaces, vol. 34, no. 3, pp. 215–225, 2004. [2] B. Bona, P. Brandimarte, C. Greco, and G. Menga, "Hybrid hierarchical scheduling and control systems in manufacturing," IEEE Trans. Robot. Autom, vol. 6, no. 6, pp. 673–686, Dec. 1990. [3] J. C. Ammons, T. Govindaraj, and C. M. Mitchell, "Decision models for aiding FMS scheduling and control," IEEE Trans. Syst., Man, Cybern., vol. 18, no. 5, pp. 744–756, Sep./Oct. 1988. [4] L. X. Tang and X. P. Wang, "A predictive reactive scheduling method for color-coating production in steel industry," Int. J. Adv. Manuf. Technol., vol. 35, no. 7-8, pp. 633–645, 2008. [5] N. G. Hall and C. Sriskandarajah, "A survey of machine scheduling problems with blocking and no-wait in process," Operation Research, vol. 44, no. 3, pp. 510–525, 1996. [6] E. D. Kosiba, J. R. Wright, and A. E. Cobbs, "Discrete event sequencing as a traveling salesman problem," Computer Industry, vol. 19, no. 3, pp. 317–327, 1992. [7] L. Lopez, M. W. Carter, and M. Gendreau, "The hot strip mill production scheduling problem: A tabu search approach," European Journal of Operation Research, vol. 106, no. 2-3, pp. 317–335, 1998. [8] L. X. Tang, J. Y. Liu, A. Y. Rong, and Z. H. Yang, "Multiple traveling salesman problem model for hot scheduling in Shanghai Baoshan Iron & Steel Complex," European Journal of Operation Research, vol. 124, No. 2, pp. 267–282, 2000. [9] P. Cowling, "A flexible decision support system for steel hot rolling mill scheduling," Computer Industrial Engineering, vol. 45, no. 2, pp. 307–321, 2003. [10] X. P. Wang and L. X. Tang, "Integration of batching and scheduling for hot rolling production in the steel industry," International Journal of Advanced Manufacturing Technology, vol. 36, no. 5-6, pp. 431–441, 2008.