A MODIFIED SIMULATED ANNEALING SEARCH ALGORITHM FOR SCHEDULING OF CHEMICAL BATCH PROCESSES WITH CIS POLICY

°Hyung Joon Kim, Jae Hak Jung†
Dept. of Chem. Eng., Yeungnam University,
Kyongsan, 712-749, KOREA

Abstract As a trend toward multi-product batch processes is increasing in Chemical Process Industry (CPI), multi-product batch scheduling has been actively studied. But the optimal production scheduling problems for multi-product batch processes are known as NP-complete. Recently Ku and Karimi [5] have studied Simulated Annealing(SA) and Jung et al.[6] have developed Modified Simulated Annealing (MSA) method which was composed of two stage search algorithms for scheduling of batch processes with UIS and NIS. Jung et al.[9] also have studied the Common Intermediate Storage(CIS) policy which have accepted as a high efficient intermediate storage policy. It can be also applied to pipeless mobile intermediate storage pacilities.

In spite of these above researches, there have been no contribution of scheduling of CIS policy for chemical batch processes. In this paper, we have developed another MSA for scheduling chemical batch processes with searching the suitable control parameters for CIS policy and have tested the this algorithm with randomly generated various scheduling problems. From these tests, MSA is outperformed to general SA for CIS batch process system.

Keywords Common Intermediate Storage, Metropolis Algorithm, Modified Simulated Annealing

1. INTRODUCTION

In the last ten years, multi-product batch process scheduling has been studied as an important research field in chemical operations because of the increasing of requirement of small quantity high value-added product. In batch process operations, there are various ways of operations so called intermediate storage policies. The different types of intermediate storage policies which frequently have been studied are unlimited intermediate storage (UIS), finite intermediate storage (FIS), no intermediate storage (NIS), zero wait (ZW) and mixed intermediate storage (MIS) policy.

Recently Ku and Karimi [4] proposed a new method of using intermediate storage tanks called shared storage system as a block of MIS policy. And Jung et al. [9] proposed the common inter- mediate storage (CIS) policy, in which the storages are commonly used throughout the whole system to accomplish the complete flexibility.

We have researched a process scheduling with CIS storage policy. There have been developed and published two kinds of methodologies of process scheduling. One is a scheduling method with mathematical optimization technique. The other is heuristic or rule based methods like branch and bound (BAB) and artificial intelligence.

Recently Ku and Karimi [5] have studied the SA as a new method which is a kind of random search with montecarlotic algorithm and it improved the results of scheduling problems.

Jung et al. developed a Modified Simulated Annealing (MSA) which is better than SA and it proved excellent results for scheduling problem with UIS and NIS policies but it has been hardly studied for batch scheduling of CIS policy. In this paper, we developed MSA for CIS policy with a proper control parameter.

2. COMMON INTERMEDIATE STORAGE

In the case of changing the ordered product items, the operations which require the specific intermediate storage policies, i.e., NIS, FIS and ZW mode, can be

This research was supported by the Yeungnam University Research Grants in 1994.

† To whom all correspondence should be addressed

changed in the same multi-batch system. In order to meet this requirement of flexible operation, the two kinds of configurations of CIS system have been suggested. One is a conventional pipe-valve and pump intermediate storage type and the other is a pipeless storage type. An example of the conventional configuration of CIS system with proper on-off valves, pumps and pipelines for flexible intermediate storage policy is shown in Fig.1.

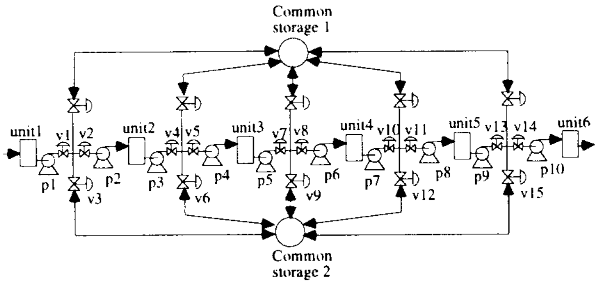

Fig.1. The CIS policy with conventional pipe-valve system.

The conventional CIS system has so many pipelines for storage and it may cause the overcharged capital cost and have high probability of making faults. The latter configuration has pipeless storage vessels which move between storage, set-up site and unit to transfer the materials. It is not a kind of full scale pipeless plant but a partial use of pipeless concept, i.e., only storage facilities are operated as pipeless moving vessels. Materials are still transferred by pipes and pumps from unit to unit. The schematic diagram of pipeless type of intermediate storage and unit for CIS system is shown in Fig.2.

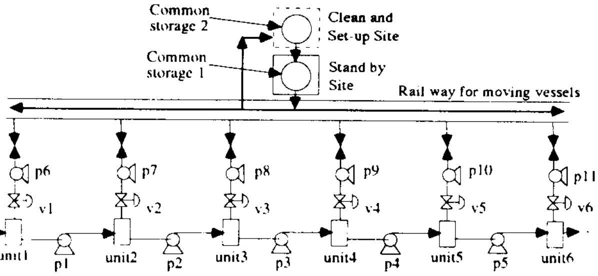

Fig.2. The CIS policy with pipeless movable storage system

For this system, a new transfer time between storage and unit should be defined. The transfer time of pipeless type of intermediate storage between storage and unit is the sum of times of moving storage, total transfer time may become shorter than that of conventional pipe-valve and pump storage system because the net material transfer time is overwhelmingly reduced. Moreover, the pipeless system can greatly reduce the waste material from cleaning pipeline and valve-pumps. However the completion time algorithm for the two kinds of CIS policies is as follows [9].

$$C_{ij}(CIS) = \max[E_{(i-1)j}, C_{i(j-1)}, \min\{\max(SE_1*(l,k)+as(l,k)), \max(SE_2*(l,k)+as(l,k)), \cdots, \max(SE_p*(l,k)+as(l,k)), E_{(i-1)(j+1)}\} - t_{ij} - a_{i(j-1)}] + t_{ij} + a_{i(j-1)}$$

$$(\text{for } j=1,2,\cdots,Q-1 \text{ and } R+1, R+2, \cdots M)$$

$$C_{iQ}(CIS) = \max[C'_{iR} - \sum_{j=Q}^{R} t_{ij} - \sum_{j=Q}^{R} a_{ij}, C_{i(Q-1)}] + t_{iQ} + a_{i(Q-1)} \quad (\text{for } j=Q)$$

$$C_{ij}(CIS) = C_{i(j-1)} + t_{ij} + a_{i(j-1)} \quad (\text{for } j=Q+1, \cdots, R-1)$$

$$C_{ij}(CIS) = \max[C_{i(R-1)}, \min\{\max(SE_1^**(l,k)+as(l,k)), \max(SE_2^**(l,k)+as(l,k)), \cdots, \max(SE_p*(l,k)+as(l,k)), E_{(i-1)(j+1)}\} - t_{ij} - a_{i(j-1)}] + t_{ij} + a_{i(j-1)}$$

$$C_{ij}^*(CIS) = C_{ij}(CIS) + a_{ij}$$

$$\textit{where } C'_{iR} = \max[C_{(i-1)Q} + s_{(i-1)iQ} + \sum_{j=Q}^{R} t_{ij} + \sum_{j=Q-1}^{R-1} a_{ij}, C_{(i-1)(Q+1)} + s_{(j-1)i(Q-1)} + \sum_{j=Q+1}^{R} t_{ij} + \sum_{j=Q}^{R-1} a_{ij}, \cdots, C_{(i-1)R} + s_{(i-1)iR} + \sum_{j=R}^{R} t_{ij} + \sum_{j=R-1}^{R-1} a_{ij}]$$

And following storage situation, the check algorithm is needed for checking the storage availability.

For given i and j :

If (C_{ij}(UIS) < $E_{(i-1)(j+1)}$ (for operation(a)), C_{ij}(UIS)+a_{ij} < $E_{(i-1)(j+1)}$ (for operation (b)) then

Do (for every introduced intermediate storage p (p=1,2,⋯P)

If(there is no [$SS_p(l,k)$, $SE_p(l,k)$] to satisfy both the condition 1) and condition 2) then

$SS_p(i,j)$ = C_{ij}(UIS)

$SE_p(i,j)$ = $E_{(i-1)(j+1)}$ + a_{ij}

Else

If (there are one or more ($SE_p(l,k)$+$as(l,k)$)s to satisfy

$SE_p(l,k)$ + $as(l,k)$ < $E_{(i-1)(j+1)}$−a_{ij}) then

$SS_p(i,j)$=max($SE_p(l,k)$+$as(l,k)$) (l=1,⋯,N,k=1,⋯,M)

=max($SE_p^*(l,k)$+$as(l,k)$

Else

$SS_p(i,j)$ = 0, $SE_p(i,j)$ = 0 and unit j should be holding the product i until the next unit j+1 become ready to process the product i

End if

End if

Continue

Else

for every introduced intermediate storage p(p=1,2,⋯,P)

$SS_p(i,j)$ = 0

$SE_p(i,j)$ = 0 this block can be eliminated by the initial definition of every $SS_p(i,j)$ and $SE_p(i,j)$ are zero

End if

where, a_{ij} = transfer time which is required for transferring product i out of unit j to $(j+1)$

$s_{i(i+1)j}$ = set-up time which is required for preparing the $(i+1)$th product after the ith product

$as(l,k)$ = the storage clean-up and preparing time after using of ith product after k unit

C_{ij} =completion time of processing ith product in j unit

t_{ij} = processing time of product i at unit j

$C_{ij}^* = C_{ij} + a_{ij}$

$E_{ij} = C_{ij} + a_{ij} + s_{i(i+1)j}$

$SS(l,k)$=storage starting times for product l after unit k

$SE(l,k)+as(l,k)$=storage ending times for product l after unit k

$SE_p^*(l,k)$ = a set of common storage ending (emptying) times $(SE(l,k)s)$ of pth (p=1,2,⋯,P) common storage which satisfy both the condition and the condition 2 for product l after unit k

$$SS(l,k) < E_{(i-1)(j+1)} + a_{ij} + as(i,j) \quad \text{----- (condition 1)}$$

$$SE(l,k) + as(l,k) > C_{ij}(UIS) \quad \text{----- (condition 2)}$$

3. MODIFIED SIMULATED ANNEALING

In recent years, SA has been successfully used to solve several combinatorial optimization problems. The basic idea behind a SA is to consider even poor solution as a intermediate acceptable one with suitable probability for avoiding local optimum. It used a concept of different configurations of energies of physical analogous state. Usually the physical systems, i.e., annealing steel, have many local minimum analogous states. The algorithm starts with a randomly chosen initial state. At every step of the algorithm, new candidate solutions are generated from the current solution by means of random perturbations. Let E_1 be the objective function value of the current solution and E_2 be that of the new solution. For the case of minimization problems, if $E_2 \leq E_1$, the new solution becomes E_2 necessarily. But for the case of $E_2 > E_1$, both of E_1 and E_2 can be a new solution with proper probability. The probability of accepting E_2 as the new solution is given by $P(\Delta E) = \exp[-(E_2 - E_1)/kT]$. A random number which is uniformly distributed in the interval [0, 1] is generated and compare with $P(\Delta E)$. If a generated random number is less than $P(\Delta E)$, the new solution is changed to old one, otherwise, it is discarded and another solution is generated from the current one. The algorithm continues when a certain termination criterion, such as a prespecified number of rearrangements, is satisfied. This scheme has been known as the Metropolis algorithm.

The efficiency of SA are strongly affected by the following parameters, the initial value of the control parameter T, the number of solutions generated at each T, the decreasing rate of T, and the efficiency in generating new solutions from old solutions. SA is terminated after $3N^3$(N is the number of products) sequences have generated. The basic issue is how the algorithm should move from one sequenence to another. Karimi used the simple strategy of reducing 5% of kT for each iteration which is composed with 20 discrete iterations steps, i.e., kT remains constant for each step but decreases by 5% from one group to next rearrangements with $0.15N^3$ iterations. This means that the final value of kT is 0.95^{20} or approximately 0.36 times the initial kT value.

We developed a new method that SA based on Metropolis Algorithm called Modified Simulated Annealing (MSA).

MSA is composed of two stage search algorithm. At the first stage, Rapid Access Extensive Search (RAES) algorithm was used for better location of initial state. RAES has been reported a simple and predominant method for flowshop scheduling. RAES was proposed by Dannenbring to solve the multiunit UIS scheduling problem [3]. It is two-phase heuristic with a recursive improvement strategy. In the first phase, a pseudo-two-unit UIS problem is generated from the original M-unit problem as follows:

$$t^*_{i1} = \sum_{j=1}^{M} (M-j+1) t_{ij} \qquad t^*_{i2} = \sum_{j=1}^{M} j t_{ij} \qquad i = 1, 2, \cdots, N$$

where t^*_{ij} is the processing time of product i on unit j for the pseudo-two-unit problem and applying it to Johnson's algorithm for generating initial sequence. In the second phase, the initial sequence is improved by generating $(N-1)$ sequences via pairwise interchanges of adjacent products in the sequence and selecting the best of those as the next solution. The procedure is repeated until no improvement is found. MSA used the result of RAES as an initial sequence.

In small size problems (N≤9), average percentage deviations of RAES which are based on the optimal solutions are reported within 10 percentage for batch flowshop with UIS. In large size problems ($N \geq 10$),, they are shown within 20 percentage away from optimum. For this reason, initial probability of MSA has not to be used $P_1(\Delta E) \sim 0.5$ like SA. That's because this value results in giving up a lot of block of good solution. At the second stage, we used lower probability of up-hill movement. Thus if it is not a predominant solution, it is hardly accepted as a new one.

Therefore we controlled initial probability of up-hill movement as follows :

$$P_1(\Delta E) = \exp(-(E_1 - E_0)/k*t_1)$$
$$= \exp(-(E_1 - E_0)/0.75*(\sum_i^{3000} |\Delta E_i| /3000)$$
$$\sim 0.264$$

Initial probability of up-hill movement is 0.329 and the last probability is as follow :

$$P_{20}(\Delta E) = 0.36*\exp(-(E_i - E_{i-1})/k*t_i)$$
$$= 0.36*\exp(-(E_i - E_{i-1})/0.75*(\sum_i^{3000} |\Delta E_i| /3000)$$
$$\sim 0.095$$

The other procedures of MSA is same as SA.

4. EVALUATION AND CONCLUSION

We have tested various scale of problems which were randomly generated for evauate performance of this study. The size of problems that have used in this study are 6×4, 6×8, ···,10×8, and 11×4 as notation of N×M. For each size of problems we have tested 20 problems which were randomly generated. The specific results of them are as shown in Table 1.

TABLE 1. The results of this study

Size of prob.	Best % of RAES	Best % of SA	Best % of MSA
6×4	65	55	80
6×8	40	90	95
7×4	45	90	95
7×8	65	30	85
8×4	20	55	100
8×8	75	10	95
9×4	5	45	75
9×8	15	50	75
10×4	10	10	95
10×8	10	20	85
11×4	0	30	75

Table 1 shows that MSA is superior to RAES and SA. In the case of smaller size problems, the results of MSA is slightly better than those of SA but as the size of problems are bigger, the results are much better than that of the other methods. At MSA, a lot of trial and error research should be made to get the suitable probability of up-hill movement with adjusting control parameter T. As the storage policy is changed, this parameter needs to be changed for better result. This study shows that the MSA should be better than conventional scheduling method with suitable control parameter of tempeture T for probability of up-hill movement.

REFERENCES

[1] G. V. Reklaitis, "Review of scheduling of process operations", *AIChE Symp. Ser.*, vol. 78, pp. 119-131, 1982.

[2] G. V. Reklaitis, "Perspectives on scheduling and planning of process operations", *Proceedings of 4th International Symp. on PSE*, Montebello, Canada 1991.

[3] H. M. Ku, D. Rajagopalan and I. Karimi, "Scheduling in batch processes", *Chem. Engng. Prog.*, August, pp. 35-45, 1987.

[4] H. M. Ku and I. Karimi., "Completion time algorithms for serial multiproduct batch processes with shared storage", *Comput. chem. Engng*, vol. 14, pp. 49-69, 1990.

[5] H. M. Ku and I. Karimi, "An Evaluation of Simulate Annealing for Batch Process Scheduling", *Ind. Engng Chem. Res.*, vol. 30 , pp. 163-169, 1991.

[6] J. H. Jung et al. "Scheduling of Multi-product Batch Processes Using Modified Simulated Annealing", *HWAHAK KONGHAK*, vol. 32, pp. 753-764, 1994.

[7] I. J. Lee, H. Jung, D. R. Yang and K. S. Chang, "Completion times algorithm and scheduling strategy of multi-product batch processes for common intermediate storage (CIS) policy : flexible intermediate storage", *FOCAPO '93*, Crested Butte, U.S.A., 1993.

[8] J. H. Jung, H. Lee, D. R. Yang and I. Lee, "Completion times and optimal scheduling for serial multi-product processes with transfer and set-up times in zero wait policy", *Comput. chem. Engng*, vol. 18, pp. 537-544, 1994.

[9] J. H. Jung, H. Lee and I. Lee, "Completion times algorithm of multi-product batch processes for CIS policy with nonzero transfer and set-up times", *Comput. chem. Engng*, accepted and in press, 1995.