

Comparative Study of Different Approaches to Solve Batch Process Scheduling and Optimisation Problems

Yaqing TAN^{1,2}, Wei HUANG¹, Yanming SUN², Yong YUE¹,

¹Department of Computer Science and Technology, University of Bedfordshire, Park Square, Luton, LU1 3JU, UK

²School of Business Administration, South China University of Technology, Guangzhou, 510640, China

yaqing.tan@beds.ac.uk, wei.huang@beds.ac.uk, meymsun@scut.edu.cn, yong.yue@beds.ac.uk

Abstract—Effective approaches are important to batch process scheduling problems, especially those with complex constraints. However, most research focus on improving optimisation techniques, and those concentrate on comparing their difference are inadequate. This study develops an optimisation model of batch process scheduling problems with complex constraints and investigates the performance of different optimisation techniques, such as Genetic Algorithm (GA) and Constraint Programming (CP). It finds that CP has a better capacity to handle batch process problems with complex constraints but it costs longer time.

Keywords—batch process scheduling; comparative study; Genetic Algorithm (GA); Constraint Programming (CP)

I. INTRODUCTION

Batch processes are widely used in process industry, such as chemical industry, food industry and polymer industry, to meet fast-changing consumers' need and market. Unlike continuous or discrete process, batch processes produce products in batches, which displays its inherent adaption for small-volume, high-value added products.

However, the adaption involves many unique features of batch process, such as Storage capacity, waiting policies and parallel machine allocation, posing huge challenges for batch process scheduling and optimization problems. Several optimization techniques are attracted to the problems, for instance, classic mathematical programming, especially Mixed Integer Linear Programming (MILP), meta-heuristics algorithms including Genetic Algorithm (GA), Tabu Search (TS) and Simulated Annealing (SA), and Constraint Programming (CP), etc.

Each technique specialises in handling different kinds of scheduling and optimisation problems. However, research on comparing the niches of different techniques is inadequate. Moreover, most research focuses on simpler job-shop scheduling problems rather than batch process scheduling problems. To obtain a deep knowledge for the best use of the techniques, the paper investigates and compares the performance of GA—one of meta-heuristics algorithms, and CP in batch process scheduling and optimisation problems.

The remainder of this paper is organised as follows. Section 2 is a brief discussion of related literature and

techniques in batch process scheduling. Section 3 presents an optimisation model of batch process. Section 4 details GA and CP approaches to solve batch process scheduling problems. In Section 5, a case study of batch process is presented. Section 6 discusses the results from Section 5 and reveals the differences in GA and CP. Finally, Section 7 concludes in paper.

II. LITERATURE REVIEW

A. Batch process scheduling problems

The classic job shop scheduling problems, one of the N-P hard problems, represent the general scheduling problems that are made up with a number of different machines and a number of different operations. However, with the development of process industries, batch process scheduling problems with more constraints emerge, which made them distinguish for job shop scheduling problems.

Storage capacity, waiting policies and parallel machine allocation are main concern in the batch processes. Specialised storage device are used to hold intermediate materials until machines for the next operations are available. Huang expatiated on storage capacity and waiting policies of batch process [1]:

“In chemical batch plants, it is usual to have limited storage capability between stages and hence FIS (finite Intermediate storage) policy is applied... FW (Finite wait) mode is not only necessary to limit the wait time of intermediate materials in a storage unit...”

Apart from FIS and FW, parallel machines account for the third distinctive character in batch process scheduling. Parallel machines are available for the same tasks and effective utilization of them increases productivities.

B. Batch process scheduling techniques

Many optimisation algorithms have been applied to solve batch process scheduling problems to improve productivity in the process industry, mixed integer linear programming (MILP), meta-heuristics and constraint programming, to name a few.

1) MILP

MILP is one of the widely used methods in the batch process scheduling problems owing to its rigorousness

and simplicity [2], and its applications are still in a fast growth.

A continuous time MILP model dealing with parallel equipment and task is proposed by Pinto and Grossmann[3] to reduce computational time by preordering constraints strategy and a decomposition scheme separately. Chen et al. [4] introduced some heuristic rules to cut down the size of the MILP model without affecting the optimality of the scheduling problem.

However, the computational time increases dramatically with the linearly increasing size of the problems. He and Hui [5] made a comparison between MILP and GA. The result showed that GA obtains a better capacity to solve large-size problems with different objectives in the batch plants.

2) Meta-heuristics

Meta-heuristics, such as Genetic Algorithms (GA), Tabu Search (TS) and Simulated Annealing (SA), all belonged to local search methods. Because of their excellent ability obtaining solutions within a reasonable time, meta-heuristics are preferred to resolve larger scheduling problems[6].

As local search methods, meta-heuristics do not formulate problems as the MILP does, which cannot guarantee to obtain the optimal solution. Furthermore, since certain kinds of constraints cannot be embedded directly into the search procedures, some approaches are developed to solve the dilemma. Penalty functions are the most popular and simplest method handling constraints in the meta-heuristics [7]. However, the problem of choosing suitable parameters for penalty functions has not been resolved yet [8].

In the paper, penalty functions are used to deal with complex constraints regarding FIS, FW and machine allocation, the three typical constraints in batch process scheduling problems. Contrasting to the other use of penalty function, parameters do not act as fatal point influencing objective functions but feedback relating to the number of constraints.

3) Constraint programming

Constraint programming (CP) is a relatively novel technique to batch process scheduling problems. Its high quality efficiency in dealing with constraints can be seen from its name, since there is no restriction to the types of constraints [9]. Usually, a CP approach consists of two phrases [10]: first, reducing the size of search space; and then, search. Zeballos et al. [11] presented a CP approach with a CP model and a search strategy, easily handling various waiting and storage policies. They admitted that optimal solutions cannot be obtained for several problems within 900s, which revealed the drawback of CP in terms of computational time.

However, most work of scheduling optimisation uses MILP and meta-heuristics. Little research adopted CP as the main technique to solve the problems [12-14]. In

addition, most work concentrates on the field of job-shop scheduling rather than batch process scheduling problems which normally involve in a mass of complex constraints. Even though handling constrains is considered to be a weakness of meta-heuristics, researchers [7, 8, 15] have been exploring new ways on GA, which acts as an example of meta-heuristics, to solve the problem and came out with considerable results. Therefore, the paper aims at addressing the performance of GA and CP to solve batch process scheduling problems with constraints.

III. BATCH MANUFACTURING OPTIMISATION MODEL

A. Batch process problem description

In batch production processes, parallel machines are used for the same operation, such as blending, storage and packing. Different final products are produced through different operations according to their specification, Fig.1.

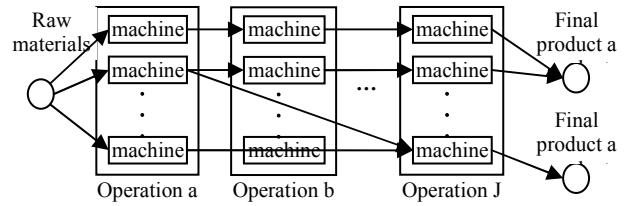


Fig.1 Batch production processes

Usually, raw materials are processed by batches, and each batch passes through a series of pre-ordered operations, for example, mixing, blending and separating, etc. Waiting time in the some operations is finite for safety concern.

B. General batch process scheduling model

1) Basic notation

- I : the total number of batches;
- J : the total number of operations;
- M : the total number of machines;
- i : one of the batches in the production;
- j : one of the operations in the process;
- m : one of the machines in the process;
- $T_{i,j}$: processing time of operation j for batch i ;
- M_j : total number of parallel machines for operation j ;
- $O_{i,j}$: operation j for batch i ;
- $ST(O_{i,j})$: start time of operation j for batch i ;
- $ET(O_{i,j})$: end time of operation j for batch i ;
- $W_{j,m}$: binary parameter for parallel machine m for operation j . When machine m is used for operation j , $W_{j,m}=1$; when machine m is not used for operation j , $W_{j,m}=0$;
- WT_j : the maximum waiting time in the operation j ;
- $Q_{m,t}$: quantity of materials in machine m at the time t ;
- MC_m : fixed capacity of machine m ;
- $MakeSpan$: the end time of the last operation.

2) Model expression

a) Objective function

In the paper, the objective function is to minimise the makespan.

$$\text{Min MakeSpan} \quad (1)$$

b) Temporal constraints

Temporal constraints are a group of general constraints, serving for all kinds of scheduling problems. Precedence constraints indicate that no idle time exists between two neighbouring operations owing to the nature of flow in the process industry. That is, in the same batch, the end time of the previous operation is the start time of the next operation.

$$ET(O_{i,j-1}) = ST(O_{i,j}) \quad (2)$$

c) Specialised constraints

There are three specialised constraints for batch process scheduling problems: machine allocation constraints, FW constraints and FIS constraints, which demonstrate the disparities between job shop scheduling problems and batch process scheduling problems.

Machine allocation constraints emerge with the appearance of parallel machines- some batches can be processed by different machines for the same operation. But one machine can only work on one batch at a time, and the number of machines in use cannot exceed the total number of machines for the operation.

$$\sum_{m=1}^M W_{j,m} \leq M_j \quad (3)$$

FW is inevitable for some operations for batch process, not only because of the scarcity of machines for the next operation, but unstable intermediate materials as well:

$$ET(O_{i,j}) - ST(O_{i,j}) \leq WT_j \quad (4)$$

As for FIS constraints, the amount of materials in one machine should be lower than its capacity to ensure steady performance.

$$Q_{m,t} \leq MC_m \quad (5)$$

IV. TWO DIFFERENT ALGORITHMS FOR SOLVING BATCH PROCESS SCHEDULING PROBLEMS

A. Genetic Algorithms

1) Chromosomes, with their crossover and mutation

Chromosomes, the key element of GA, and their encoding affect the performance of GA remarkably. In the paper, permutation encoding uses a set of integers to display the order of operations. Each integer represents a specific batch, and an operation of the batch is carried out when the integer appears in the chromosome.

For crossover, one or more integers in the chromosome are chosen randomly first, but at least two integers are left for the change. Then, the selected numbers are kept in the same positions in both chromosome and the un-chosen numbers are replaced by the integers in the other chromosome following the same sequence as in the original chromosome. The method for crossover ensures the feasibility of solutions for retaining the same number of each integer in the chromosome. An illustration is presented below (Fig.2):

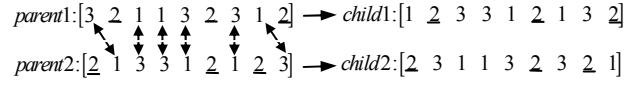


Fig.2 Chromosomes mutation

Mutation appears in a rather simple way. First, two randomly selected points divide the chromosome of a chosen parent into three parts. Then the middle part is reversed to obtain the next generation.

2) Penalty functions

From the structure of chromosomes, it is easy to notice that chromosomes alone cannot implement constraints of the batch process scheduling problems. Therefore, penalty functions are used to transform the constrained optimisation problems to unconstrained problems.

In the paper, exterior penalty functions are used, which means that the quest for optimal solutions starts from infeasible solutions to feasible solutions [16]. Generally, parameters in the functions play remarkable roles in searching process. However, since all constraints in the problems are “death constraints”, which cannot be violated at all, the parameters are set to be a figure with a greater order of magnitude.

FIS and FW constraints are carried out with the help of penalty functions. Once a constraint is violated, penalty value will be added to the objective functions. The more constraints are violated, the larger the outcome will be. The whole procedure in pseudo code is shown below:

```

Read the sequence of all operations.
For each operation
Allocate to the available machines.
Calculate its start time and end time.
If end time-start time > predetermined waiting time,
then
Add penalty value to the objective function.
End if
Calculate the quantities in the storage machine.
If the quantities > capacity, then
Add penalty value to the objective function.
End if
End loop

```

Two of the chromosomes with the smallest outcomes, which violate the least constraints, are chosen to be parents, producing better offsprings to obtain the minimum in objective functions. After iterations, the number of violations will decrease and the feasible solution will be found. In some extreme cases, no feasible solution is found, but it is possible to analyse deep into the problem by using the information of constraint violation from the results.

3) Machine allocation

Because of the existence of parallel machines for one operation, machine allocation is dispensable in the step. All machines for the same operation are numbered, and products are allotted to the machines with a small number first. When the products come, they go to machine 1 first. If machine 1 is in-process, they go to machine 2 instead.

B. Constraint Programming

1) Variables

Comparing to chromosomes of GA, variables consist of the foundation of CP methods. Variables are employed to provide all information to operations, such as when to start or end, where to process, etc.

Two kinds of variable, interval variables and sequence variables, are essential for scheduling problems. Interval variables express the time performing operations, sequence variables display the order of the operations on each machine.

2) Constraints

a) Waiting time constraints

Waiting time constraints are set easily by utilising interval variables, which represent a period of time directly. Waiting time is restricted with the exact values provided by interval variables.

b) Capacity constraints

Capacity constraints are carried out with the help of interval variables. At the start time of an operation, the quantities of the product that will be processed are added. At the end time, the quantities are deducted. Then, the quantities of product in a machine are known. To restrict the quantity, a function that confines the quantities less than the capacity during the processes is employed.

c) Machine allocation

Machine allocation is implemented by a data structure suggesting the available parallel machines in the specific operation. In addition, two more functions regarding the overlaps of machines and batches are developed to exclude the situations that one machine works on different batches simultaneously or one batch is processed on the different machines simultaneously.

V. EXPERIMENT

A. A case study

To provide a convincing comparison between the performance of GA and CP in batch process scheduling problems, a simple example from a published paper [1] is introduced here as a case study, see Fig.3.

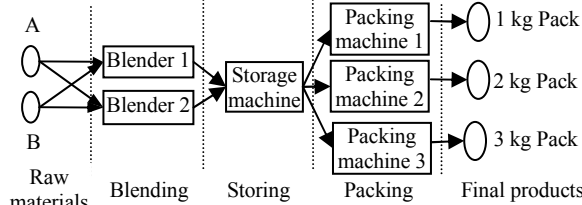


Fig.3 Production process of the example

60 tonne raw material A and 60 tonne raw material B are blended with equal proportion in the two identical blenders with a fixed capacity of 5 tonne. After blending for 2 hours, all products pass on to the storage machine with a capacity of 15 tonne, waiting to be packed. However, the products can only wait in the storage machine for 6 hours at most. For packing tasks, the packing machine is able to pack 2500 packs per hour. The example is designed to obtain 20 tonne of 1-kg-pack-

size products, 20 tonne of 2-kg-pack-size products and 20 tonne of 3-kg-pack-size products as final products.

In the example, to meet the restriction of capacity and waiting time, the products are divided into 12 batches, operating in a cyclical mode to achieve the product demand. The example is solved on a PC with AMD Athlon (tm) 2.91 GHz Processor and 1.75 GB of RAM by GA and CP, respectively. The scheduling models are built on an hourly basis, aiming at minimum makespan, which is the total time completing all the operations. For the purpose of comparing GA and CP, some criteria are put forwards to evaluate the algorithms:

- The optimal solutions from each algorithm.
- The computational time for the optimal solutions.

B. Parameters for GA and search strategy for CP

As for GA, the parameters are set as follow:

- Selection function: The parents are selected by tournament. The winners of the competition are chosen to be parents.
- Crossover rate: 0.2
- Mutation rate: 0.01
- Population size: 300
- Termination condition: The algorithm stops if the average change in the objective function value over 500 generations is less than $1e-6$.

A constructive search and constraint propagation are used in CP to obtain the results[17]. The constraint propagation is used in the first to reduce the search space by removing unfitted values from domains which contain all possible values for the results. After the constraint propagation, a constructive search is carried out. It tried the remaining values in the domains to see whether a feasible solution can be found.

C. Experiments and results

1) Optimal solutions of the example

Both of GA and CP solved the problem in the case study and obtained the same result: 19 hours as the smallest makespan, which also is the same as the result by Huang and Chen [1], using different approaches. However, the schedules provided by GA and CP are different, even with the same result (Fig.4 and Fig.5).

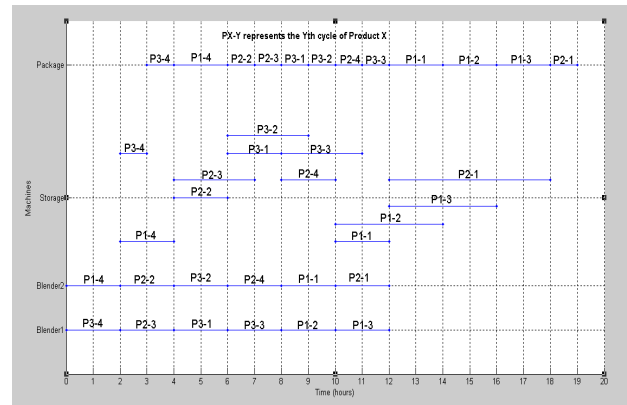


Fig.4 The optimal solution from GA

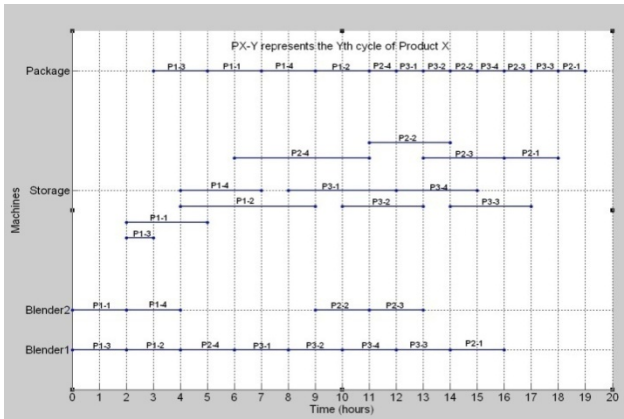


Fig.5 The optimal solution from CP

In the blending operation, the GA approach assigned tasks to the two blenders evenly. In contrast, CP tended to use Blender1 so much that Blender2 were idle for some time. Additionally, the schedules obtained by GA varied from different runs while those from CP remained the same. Regardless of above differences, CP cost 9.0s to obtain the results, which was triple of GA. The small case above only provides a superficial look at the performance of GA and CP, and a further discussion will be made in the next section.

VI. DISCUSSION

Generally speaking, the search process of GA mainly involves the continuous changes in chromosomes. The uncontrollable changes lead to the difficulty to obtain feasible solutions in some complex problems and different schedules for each run. On the other hand, it is difficult to embed constraints information, such as FIS and FW, into chromosomes. To solve the problem, penalty functions are used in the paper. However, because of the huge influence of parameters in the penalty functions on the results [16], GA cannot guarantee to find the feasible solutions for some problems without proper parameters, sometimes.

Unlike mutations in chromosomes of GA, CP searches every possible combination of values in the search space, just like to search in a tree, from branches to branches. CP searches the tree with a fixed pattern, which accounts for the fact that the solution for the same problem is the same. Since CP searches every branches to attain the result, it is able to find the feasible solutions for the problems as long as the solutions exist, but it costs a longer time, comparing with GA.

VII. CONCLUSION

Batch process scheduling and optimisation problems have gathered more and more attention from industry and academia. However, there is still little research reported on comparing performance of different optimisation algorithms in the field, particularly between meta-heuristic and CP. The purpose of this study was therefore to explore their capabilities in solving batch process scheduling problems with complex constraints and evaluate their performance to find the proper one to study

on the problems further. From the experiments and discussion, it can be concluded that CP obtains a better capability for problems with complex constraints but costs longer time. Further research is needed for making better use of optimisation algorithms.

REFERENCES

- [1]Huang, W. and B. Chen, *Scheduling of batch plants: Constraint-based approach and performance investigation*. International Journal of Production Economics, 2007. **105**(2): p. 425-444.
- [2]Floudas, C. and X. Lin, *Mixed Integer Linear Programming in Process Scheduling: Modeling, Algorithms, and Applications*. Annals of Operations Research, 2005. **139**(1): p. 131-162.
- [3]Pinto, J.M. and I.E. Grossmann, *A Continuous Time Mixed Integer Linear Programming Model for Short Term Scheduling of Multistage Batch Plants*. Industrial & Engineering Chemistry Research, 1995. **34**(9): p. 3037-3051.
- [4]Chen, C., et al., *Optimal Short-Term Scheduling of Multiproduct Single-Stage Batch Plants with Parallel Lines*. Industrial & Engineering Chemistry Research, 2002. **41**(5): p. 1249-1260.
- [5]He, Y. and C.-W. Hui, *Genetic algorithm for large-size multi-stage batch plant scheduling*. Chemical Engineering Science, 2007. **62**(5): p. 1504-1523.
- [6]He, Y. and C.-W. Hui, *A rule-based genetic algorithm for the scheduling of single-stage multi-product batch plants with parallel units*. Computers & Chemical Engineering, 2008. **32**(12): p. 3067-3083.
- [7]Venkatraman, S. and G.G. Yen, *A Generic Framework for Constrained Optimization Using Genetic Algorithms*. Evolutionary Computation, IEEE Transactions on, 2005. **9**(4): p. 424-435.
- [8]Al Jadaan, O., L. Rajamani, and C.R. Rao. *Parameterless penalty function for solving constrained evolutionary optimization*. in *Hybrid Intelligent Models and Applications, 2009. HIMA '09. IEEE Workshop on*. 2009.
- [9]Mendez, C.A., et al., *State-of-the-art review of optimization methods for short-term scheduling of batch processes*. Computers & Chemical Engineering, 2006. **30**(6-7): p. 913-946.
- [10]Banaszak, Z.A., M.B. Zaremba, and W. Muszyński, *Constraint programming for project-driven manufacturing*. International Journal of Production Economics, 2009. **120**(2): p. 463-475.
- [11]Zeballos, L.J., J.M. Novas, and G.P. Henning, *A CP formulation for scheduling multiproduct multistage batch plants*. Computers & Chemical Engineering, 2011. **35**(12): p. 2973-2989.
- [12]He, Y. and C.W. Hui, *Genetic algorithm for large-size multi-stage batch plant scheduling*.

Chemical Engineering Science, 2007. **62**(5): p. 1504-1523.

[13]13.B.P.Das, et al., *Comparative study of time-based and activity-based production scheduling*, in *2nd Conference on Process Integration, Modelling and Optimization for Energy Saving and Pollution Reduction*. 1999: Budapest, Hungary. p. 7-10.

[14]14.Shaw, K.J., et al. *Genetic algorithms for multiobjective scheduling of combined batch/continuous process plants*. in *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*. 2000.

[15]15.Wang, K., et al., *A genetic algorithm for online-scheduling of a multiproduct polymer batch*

plant. Computers & Chemical Engineering, 2000. **24**(2-7): p. 393-400.

[16]16.Coello Coello, C.A., *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, 2002. **191**(11-12): p. 1245-1287.

[17]17.IBM. *IBM ILOG CPLEX CP Optimizer*. 2012 [cited 2012 29th, March]; Available from: http://www-01.ibm.com/software/websphere/products/optimization/cplex-studio-preview-edition/?S_TACT=109HE1MW&S_CMP=web_ibm_ws_ilg-opt_hero_cplexoptimizationstudio-ov.