STUDIA UNIVERSITATIS BABES-BOLYAI, CHEMIA, L, 2, 2005

SCHEDULING BATCH PROCESSING: GENETIC ALGORITHMS VERSUS MATHEMATICAL PROGRAMMING

JERKER BJÖRKQVIST

Åbo Akademi University, Turku, Finland

ABSTRACT. Genetic algorithms have during the recent years gained popularity also in the domain of chemical engineering, among other applications for scheduling purposes. A genetic algorithm builds new sequences by combining and mutating previous sequences of genes, i.e. chromosomes, into a new set of chromosomes. In this new set, only the fittest survive, and the procedure is repeated. As a schedule in chemical batch plant can be seen as a sequence of starting points for the batches, the methodology of genetic algorithms can be applied also to batch scheduling.

In this work, the genetic algorithm approach is used combined with a Discrete Event Simulation (DES) approach. Here the genetic algorithm determines the order of the batches, whereas the DES-approaches insert the batches in the actual schedule. Using this two-stage optimization and some tuning of the DES procedures, schedules with similar objective function values as with mathematical optimization can be achieved, but usually much faster, which is essential for industrial scheduling systems.

INTRODUCTION

Genetic algorithms are a straightforward way of generating sequences that can be used for different purposes. Even if the implementation is rather trivial, applying genetic algorithms to actual problems are seldom as easy. This is also the case with batch scheduling. First of all, the question is how to represent the batches using the genes in the chromosome. One solution is to let the chromosome, i.e. the sequence of genes, represent the feed-order in which the batches are "entered" into the plant. As we now know the order of the batches, we have to somehow determine the real starting times of the batches. Only then we can give a goodness of the schedule, which is needed for the genetic algorithm. One method used for determining starting times for batches is Discrete Event Simulation (DES), used by e.g. Azzarello-Pantel et. al. (1998) [1]. This they call a two-stage methodology, the genetic algorithm determine the relative order of the batches, whereas DES determine the real starting times. Löhl et. al. (1998) [4] use a somewhat similar approach but use an analogue term "schedule builder" for the operations of DES.

One typical issue is that the DES operation has to use some particular heuristics for building the schedule. This heuristics might for instance be to insert the following run as early as possible on any available machine. This heuristics may however yield schedules that are not optimal in some other context. Schedules also often render infeasible, if batches are inserted one by one and internal batch relationships at some point are impossible to fulfill.

The two-stage methodology can also be adopted so that the gene represents for instance the start of a series of batches that are tightly coupled. Here, *tightly coupled* means that intermediates between the batches have limited storage time. This decreases the search space and hence erases candidates for optimal solutions. However, an open question is if these erases candidates truly contain optimal solution to the schedule problem.

In this work, the basic two-stage methodology of a genetic part and a DES part is used. In this approach, three different ways for connecting the genetic algorithm and the DES part is used. The first version is individual insertion of batches; the second is where complete chains of batches are inserted. A chain of batches here corresponds to that part of the production chain where intermediate products have limited storage time. The third version is where batches are inserted one by one, but now the gene represents from which chain the batch should be inserted into the schedule.

This resembles the work simultaneously performed by Berning et. al. (2004) [2], in which the GA schedule-builder approach is augmented by chain building procedures.

In this work, the methodology is tested using both some more theoretical job-shop-type scheduling and with some real-life scheduling problem taken from a pharmaceutical plant. In the plant in question typical conditions for the scheduling arises, such as different type of storage of the intermediates, sequence dependent cleaning and other issues that limits the sequencing possibilities of equipment used.

Similar schedules have earlier been optimized by Roslöf et. al. (2001) [5] and Björkqvist et. al. (2002) [3] using mathematical programming methods. In this work, the schedules are built using the GA/DES-methodology, and the tests shown that GA/DES perform very well. Here this mean that an objective functions of similar quality will usually be reached much faster using GA/DES methodology compared to the mathematical programming approach used in earlier works.

PROBLEM FORMULATION

In a batch processing plant, final products are produced using a set of operations in the plant. A recipe denotes the operations needed for producing some amount of a final product. The recipe also denotes the internal relation between the operations, which typically is precedence constrains, i.e. which operations produces respectively consumes the intermediates. We here call operations that are instantiated "runs". A run can be performed on a set of suitable equipment *Ui*. Now, the scheduling problem is to find starting times *ti* and selected unit *ui* for all runs, according the selected objective function. The objective function can vary, often minimizing the total make-span is used, but other might be important, e.g. minimizing the setup-time needed between operations or minimizing the number of late orders. In this paper, we use the total make-span as the objective function.

SCHEDULING USING MATHEMATICAL PROGRAMMING

A mathematical programming approach can be used for scheduling. Here, we use a Mixed Integer Linear Programming (MILP) formulation for specifying the optimization problem. The basic approach is to compare relative orders of the runs and build inequalities accordingly:

$$t_j + t^p{}_j < t_j \ \vee \ t_j + t^p{}_j < t_j$$

which using a "Big-M" reformulation is transferred into a MILP expression

$$t_i + t^p{}_i < t_j + y_{i,j} \ M$$

$$t_j + t^p{}_j < t_i + (1 - y_{i,j}) \ M$$

where $y_{i,j}$ is a binary variable specifying if run *i* will precede run *j*. In the case of more general batch scheduling, this notation is not enough, but these formulation are not given here, but can be found in Björkqvist et. al. (2002) [3].

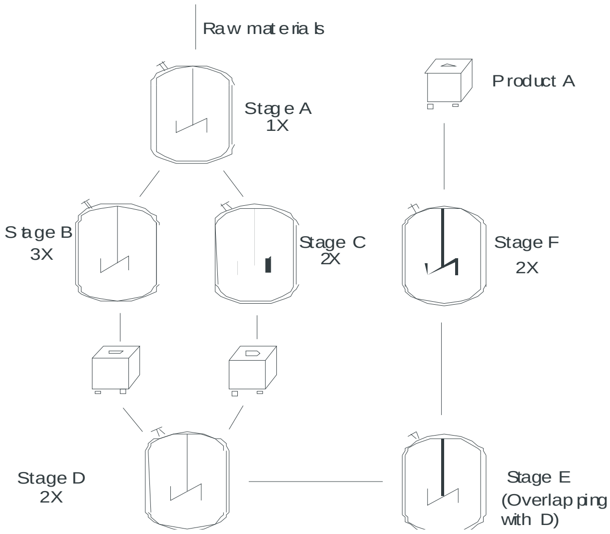

Figure 1. Example of a recipe

The basic MILP formulation presented above works fine for small problems, and optimal solutions to the scheduling problem will be provided. Practical problem can however seldom be solved using direct MILP formulations, the methodology used here is instead a iterative procedure, which is basically contains the following steps:

1. Solve a subset of the original scheduling problem to optimality
2. Fix the relative order of the runs in the previously solved problem, enlarge the subset of the problem and resolve until completely solved

This procedure is moreover called Sequential Updating Procedure (SUP). If we start reinserting runs in the schedule, we call this procedure Post Processing (PP).

SCHEDULING USING GENETIC ALGORITHMS

The proposed Genetic Algorithm approach is actually a combined Genetic Algorithm and Discrete Event Simulation (DES) approach. The Discrete Event Simulation is used for simulating systems where discrete events change the otherwise continuous simulations. Here, the discrete events are when operations are started and stopped in the plant. Now, we let the Genetic Algorithm provide the order in which the events are taking place, whereas the DES-simulation provides the timing information. The DES here use the logic:

* For the given run, selected the machine where it first can be started, and derive the starting time for this run

Now, when the procedure is repeated for each run, a schedule is generated. From this schedule, the metrics of the schedule can be calculated. Some selected metrics (e.g. the total make-span) of the schedule is now providing the fitness-value.

In a genetic algorithm, a chromosome represents a solution candidate to an optimization problem. In our scheduling case, the chromosome represents in which order DES will insert runs to the schedule. We let the genes in the chromosome specify which run the DES should insert next.

However, many times this lead to infeasible situations, as the production process also requires some particular run precedence. If the chromosome specifies a run order that that contradicts to this precedence, the schedule cannot be build. Hence, we let the gene specify which groups of runs to be scheduled next, inside this group the runs are ordered according to internal precedence. This is the approach used here.

The GA is implemented the standard way using a crossover, mutation and elitism.

TEST USING JOB-SHOP-PROBLEMS

In order to validate the general quality of the schedules achieved with the two approaches, some standard job-shop scheduling problems were solved. In a job-shop problem, a job is a collection of operations (runs), each of which shall be performed once on each machine. In the job-shop problem, the order in which the runs shall be performed on the machines are given. Here, we test with two different sizes of job-shop-problems, which are built according to the OR Library of test problems. The job-shop instances where 6x6, 10x10, 20x5, 15x15. The objective function used here is to minimize the make-span.

Table 1

Objective function value (minimizing make-span) and processing times for some job-shop problems

JSP size	SUP	SUP+PP	GA
6x6	58 (0 s)	55 (0,1 s)	55 (14 s)
10x10	1401 (0,9 s)	994 (450 s)	1018 (11 s)
20x5	1259 (90 s)	1206 (260 s)	1207 (16 s)
15x15	1503 (6 s)	1191 (75 s)	1243 (86 s)

Additionally, we can compare how the objective function value is developed over time for the methods SUP+PP versus GA. In figure 2 the typical trends are shown. GA drops rapidly to an asymptotical level, where as the mathematical programming approach slowly moves towards the same level, and by time if will be better than the GA approach. This has been observed as the general trend.

SOLVING A PRODUCTION PLANNING PROBLEM

Next, the two methodologies are applied to the actual problem, a scheduling problem for batch operations. What here is different from the pure job-shop problem is that the schedulers also have to handle some new sequencing constrains, such as limited shelf life of intermediate products. Other new issues are for instance that one run may wait on several runs before it can be started. Intermediate storage is also an important issue, as the production equipment itself often is used as storage for intermediates. Now the question is if we can draw similar conclusions for the more general scheduling problem in the chemical industry.

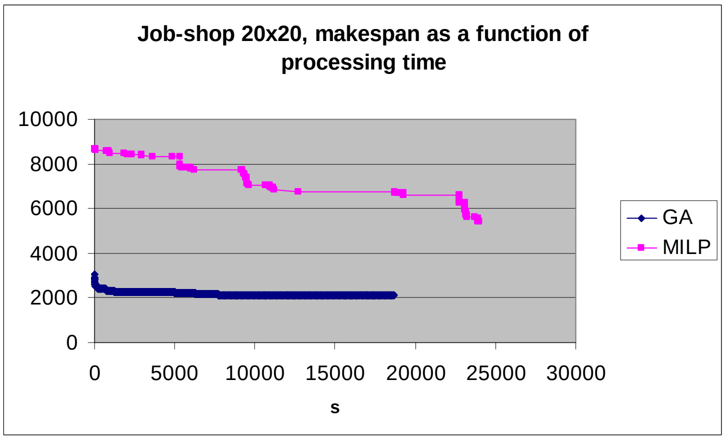

Figure 2. Objective function value for SUP+PP / GA

As an example, a fictive product P is produced at a plant. This product P is produced according to the recipe given in figure 3.

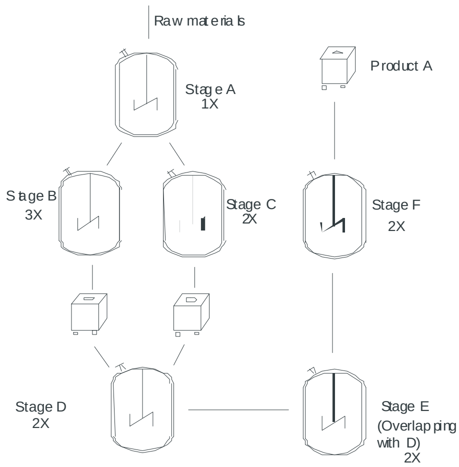

Figure 3. Production recipe for A

In the figure we note that the production of one batch of final product A is started by stage A, where after intermediates products are used by stage B and stage C. After this we store the intermediates temporally and feed them to stage D. So it continues until we finally get product A. Each stage of production has a production time, cleaning time and available units according to table 2.

The basic batch is for production of 150 kg final product A. Now the scheduling problem is solved by both the mathematical approach and the GA approach. The results are given in table 3, here has also the total make-span been used as the objective function.

Table 2

Stages for production of A

Stage	Units	Def. cleaning	Processing	Shelf-life
A	U1, U2	4	7	48
B	U3, U4	2	4	60
C	U3, U4	2	7	60
D	U1, U2	4	10	24
E	U5	4	8	24
F	U6	0	1	48

Table 3

Optimization results for production of A

Case	Amount	SUP+PP	GA1	GA2
1	300	2160 (0s)	2160 (0s)	2460 (0s)
2	600	3360 (3s)	3300 (1s)	3600 (1s)
3	900	4440 (198 s)	4440 (24s)	4740 (4s)
4	1200	5760 (171s)	5580 (1s)	6060 (1s)
5	2400	13740 (407s)	10440 (18s)	10380 (38s)

In the table GA1 and GA2 denotes slightly different DES inserting strategies, where GA1 is a single run inserting strategy and GA2 is a strategy of always inserting runs in the precedence order. Here we see that the GA optimization method is rather competitive, and produce good results very fast. However, the trend is also here like the trend in job-shop problem that in the long run SUP+PP produces better results.

The corresponding comparison of objective value development during the optimization is shown in figure 4. In the beginning, the GA approach rapidly drops to the level that remains almost constant. SUP+PP (MILP in the figure) in this case also seems to be remain on a higher level than the GA approach, but would eventually reach the optimal objective function value.

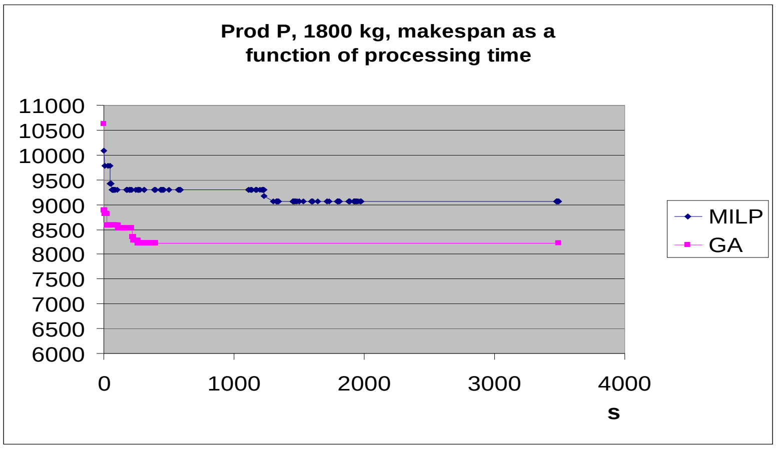

Figure 4. Objective function as a function of optimization time

SUMMARY

In the paper, we have shown two different ways of optimizing batch scheduling. We have shown that both mathematical programming and genetic algorithms can be applied, and it was shown that scheduling results using both approaches are comparable. However, the general trend seems to be that genetic algorithms more rapidly produce good results, hence genetic algorithms should be a good choice for interactive systems, where the need for fast response times of the scheduling subsystem is essential. However, the mathematical programming approach can still be important for getting information on the quality of the solution.

REFERENCES

1. Azzaro-Pantel, C., Bernal-Haro, L., Baudet, P., Domenech, S., Pibouleau, L., 1998. A two-stage methodology for short-term batch plant scheduling: discrete-event simulation and genetic algorithm. Comput. Chem. Eng. 22 (10), 807-825.
2. Berning, G., Brandenburg, M., Gürsoy, K., Kussi, J.S., Mehta, V., Tölle, F-J., 2004. Integrating collaborative planning and supply chain optimization for the chemical process industry (I) methodology. Comput. Chem. Eng. 28 (6-7) 913-927.
3. Björkqvist, J., Roslöf, J., Rönnback, R., Westerlund, T., 2002. Evaluation of an iterative mathematical optimization approach for industrial size batch scheduling. In: Pierucci, S. (Ed.), Selected Papers of The Fifth Italian Conference on Chemical and Process Engineering. Vol. 5, 33-40.
4. Löhl, T., Schulz, C., Engell, S., 1998. Sequencing of Batch Operations for a Highly Coupled Production Process: Genetic Algorithms Versus Mathematical Programming. Comput. Chem. Eng. 22 (Suppl) S579-S585.
5. Roslöf, J., Harjunkoski, I., Björkqvist J., Karlsson S., Westerlund T., 2001. An MILP-based reordering algorithm for complex industrial scheduling and rescheduling, Comput. Chem. Eng 25 (4-6), 821-828.