BULETINUL Universităţii Petrol – Gaze din Ploieşti	Vol. LXVI No. 2/2014	75 – 80	Seria Tehnică

A Genetic Algorithm for Production Scheduling in Flexible Manufacturing Systems

Florentina Alina Toader

Universitatea Petrol-Gaze din Ploiesti, Bd. Bucuresti 39, 100680, Ploieşti
e-mail: toader_florentina_alina@yahoo.com

Abstract

The manufacturing industries respond to the increasing and changing demands of the industrial market; in this context an efficient production scheduling is linked to the resources allocation efficiency by reducing the waste and to a shorter product life cycle and an increased production variety. The main objective of the Flexible Manufacturing Systems (FMS) is the equilibrium between flexibility and productivity. Thus the dedicated literature proposes a series of methods that manages to achieve a proficient production scheduling concerning a list of specific criteria. In this article a simple Genetic Algorithm (GA) ɨ for a FMS is implemented in order to obtain a production scheduling that maximizes the satisfaction of the technological constraints and minimizes the completion time of the jobs. The algorithm is tested in different scenarios and compared with a classical scheduling algorithm (Round Robin) and the results and conclusions are presented.

Key words: *Genetic Algorithms, Production Scheduling, Artificial Intelligence*

Introduction

A Flexible Manufacturing System (FMS) is characterized by Stecke in 1983 as an integrated, complex computer-controlled arrangement of computer numerical controller (CNC) and automated handling devices that can simultaneously process a variety of product types. FMS can adapt to the production sequence of different types of products and have the ability to produce them without special modifications to the original system [10].

The challenge is that the production must be accomplished in the shortest possible time and with optimal usage of system resources. The production scheduling can be defined as the technique that manages each step of the production sequence so it is placed at a specific moment in time, in order to maximize the production efficiency [8].

Job Shop Scheduling problems are considered to be combinatorial optimization problems, included in the NP-complete category [5]. The use of a heuristic method, such as Genetic Algorithm, is justified in this case, according to Nefiz [6].

The present paper is structured as follows: section 1and section 2 contains a brief review over the scientific literature on the proposed subject (production scheduling in flexible manufacturing systems) and the application of genetic algorithm in this area, section 3 presents the mathematical model of the associated problem, the implementation of GA is described in section 4 and the analysis of the obtained results and conclusions in section 5.

Literature review

There are many objectives connected to the problem of production scheduling on Flexible Manufacturing Systems. Due to the large number of constraints and objectives that needs to be simultaneously satisfied, the production scheduling is part of the NP-hard problem category [9]. In 1996, Vaessens [11] suggests the implementation of a local search technique to solve this problem. In 1995 Laureco [5] proposes the simulated annealing technique as a method for the job shop scheduling problem. A complete study of job shop scheduling techniques along with a comparative analysis can be found in a study presented by Jain and Meeran [4]. In 1985 Davis [1] recommends the first implementation of a genetic algorithm in this area. The GA approach has become very popular for its ability to perform an intelligent probabilistic search [12]. A model for solving the problem of planning and scheduling in a production environment with multiproduct capabilities is proposed by Ip et al. [3]. Pongcharoen et al. [7] propose~~s~~ the implementation of a GA in a multiple level product system as a scheduling tool to manage the multiple resource constraints. In 2010 an improved chromosome design is included in a GA to solve FMS [14]. A GA is introduced in [15] to increase the performance of the operation allocation and machine tool selection in FMS. A GA approach for integrated process planning and scheduling in FMS is introduced in [16]. The study of the dedicated literature demonstrates that Genetic Algorithms are involved in multiple researches in different areas, the Flexible Manufacturing System and production scheduling being some of them.

FMS Problem

The mathematical model

The Flexible Manufacturing System can be described as a Job Shop Scheduling problem. The objective is to minimize the completion time of the jobs, represented by the total amount of time needed to complete the requested number of jobs (series for each product).

The Input Data are: P – a set of jobs $(1\ldots n)$, M – a set of machines $(1\ldots m)$; O_i – a set of operations for the job $i\in P$; over the set of operations of the same job, precedence constraints relations are imposed: for example, $j_1 \rightarrow j_2$ between the operations j_1 and j_2, PT_{ij} – the processing time of the operation $j\in O_i$ of the job $i\in P$; W_{ik} – the waiting time of the job $i\in P$ to access the machine $k\in M$ after finishing the previous operation corresponding to the machine $k\in M$ (if it exists).

Considering the above notations, the objective is to find for each operation $j\in O_i$, $i\in P$, the minimum starting time t_{jik} on the machine $k\in M$ as follows [13]:

$$\forall j \in O, i\in P, k\in M, \ t_{jik} \geq 0 \tag{1}$$

$$\forall \ j_1, j_2 \ \in O, i\in P, k\in M, j_1 \rightarrow j_2 : t_{j2ik} \geq t_{j1ik} + PT_{ij} \tag{2}$$

$$\forall \ i\in P, k\in M, j_1, j_2 \ \in O \ , j_1 \neq j_2 \ , \ M_{j1} = M_{j2} : (\ t_{j2ik} \geq t_{j1ik} + PT_{ij1}\ ) \vee (\ t_{j2ik} \geq t_{j1ik} + PT_{ij2}\ ) \tag{3}$$

In order to determine the total completion time $C_{\max}$ of the jobs the following condition must be satisfied [13]:

$$\forall j \in O, i\in P, k\in M, C_{\max} \geq t_{jik} + PT_{ij} \tag{4}$$

In this context the objective of the problem is to minimize $C_{\max}$.

Problem description

A flexible manufacturing system is described by a given number of multifunctional computer numerical controlled machines equipped with different tools and a potential number of products that needs to be processed. Each product has a specific processing sequence of the operations and has specific requirements. The FMS considered is formed by maximum 10 multi-operational machines and supports the fabrication of maximum 10 products in the same shift.

The problem was considered for different FMS sizes and different number of products series as Table 2 presents; also, for each product, a different processing order is considered, as presented in Table 1.

A set of assumptions was made before solving the problem in order to minimize the complexity: at the first moment of time when the production scheduling starts, all the raw materials and machines are available; a machine can process a single product in a moment of time; the product processing once started on a machine cannot be interrupted; transportation time between machines is neglected.

Table 1. Processing order for each type of product

Product Name	Machines order	Processing time on machines
Type1	1, 3, 2	10, 20, 10
Type2	2, 5, 1	10, 30, 20
Type3	3, 2, 1	10, 10, 20,
Type4	7, 5, 3, 6, 4	30, 20, 10, 10
Type5	2, 5, 3, 4, 8, 7	10, 20, 30, 10, 10, 20

Table 2. Different FMS Scenarios

FMS Name	Number of Machines in FMS	Product Name	Number of Product Series
FMS1	3	Type1, Type3	10, 10
FMS2	5	Type1, Type2	10, 10
FMS3	7	Type1, Type3, Type4	8, 10, 8
FMS4	7	Type1, Type2, Type3, Type4	10, 12, 7, 10, 5
FMS5	10	Type1, Type2, Type3, Type4, Type5	10, 10, 10, 10, 10

The Genetic Algorithm Implementation

The Genetic Algorithm (GA) is inspired by the natural evolution principle and represents a population-based optimization algorithm [2]. In the natural process of evolution, the biological selection is made by taking into account that the selected individuals should be the ones with the best genes, which will guarantee the progress of the entire population. So GA starts with a randomly generated population and through the iteration process tries to evolve to the solution with the best quality (by minimizing the fitness function associated to each individual). GA is capable of dealing with various types of problems, including optimization problems in the manufacturing area, but its efficiency depends mostly by the way of handling the constraints of the problem [12].

The chromosome representation is a very important part because it characterizes the way that the problem is mapped on GA individuals (candidate-solutions of the problem). A sample of chromosome sequence corresponding to the problem presented in this article is introduced in Figure 1. Each part of the sequence is formed by two positive numbers that represent the product type and respectively the number of the series (job) it belongs to. For the example presented in the figure 1, the first product that enters to the production line is the Type1, with the series number 8, after that enters the product Type 5, series 3 and so one.

Part 1		Part 2		Part 3		Part 4		Part 5		Part 6		Part 7		Part 8		Part 9		Part 10	
1	8	5	3	1	2	3	5	4	2	5	1	3	9	1	4	4	1	2	3

Fig. 1. Chromosome sequence sample

The fitness function is designed to evaluate the performance of the individuals (candidate-solutions) at every step. Its value shows in what measure the corresponding schedule respects the objectives. To determine the fitness function value, a set of relevant information is found in the chromosome structure: the total processing time, the idle time of the machines, the idle time of the products and the total number of finished products. The mathematical formula to determine the fitness function is, where the operation order is considerate the one in the chromosome:

$$f(P,O,M) = \sum_{i\in P}\sum_{j\in O_i}\sum_{k\in M}\frac{t_{xjk}}{w_{ik}} \tag{4}$$

Genetic Algorithm Pseudocode

Begin

Initialization

Repeat

Evaluate Fitness

Selection

Crossover

Mutation

Until stopping conditions are met

End

The selection of the best individuals is based on the value of the fitness function and the selected individuals are chosen to be parents. In order to do that, the crossover method chosen is tow-point crossover which allows changing the chromosome chains between two positions and generates two new children chromosome chains. In order to obtain two children that respect the production constraint, a function is designed with the purpose of solving the discrepancies occurred before including the children into the current population.

Also to respect the nature model which GA is based on, a mutation operator is implemented. This operator performs small changes in the chromosome chain, consisting in changing the order of two products in the production schedule to be mutated, taking into account the constraints and the objectives.

Results and Discussion

The algorithm is implemented in C++ language, using QT Creator development environment. The product types that should be included in the current plan can be selected due to the application interface. The rest of the input data such as FMS scenario, production sequences and completion time for each step of the proposed scenario, are included in an input text file.

Once the product types are selected, the series number for each one can be set in the corresponding text box. After running the algorithm, the application output represents a short preview over the solution composed by ordered couples of numbers representing the product type and the production series.

The application is tested for a number of 5 different scenarios, as presented in table 1 and 2. Each data set has a different number of operations, a different production sequence and different machines who can be assigned to a product. The parameters that are specific to GA, such as initial population size, maximum number of generations, maximum number of individuals, crossover and mutation probability are included in an input text file.

The input test parameters values were: crossover probability CP = {0.4, 0.6, 0.8}, mutation probability MP = {0.02, 0.15, 0.08, 0.20}, initial population dimension IPD = {10, 20, 30, 40, 50}, final population dimension FPD = {60, 80, 100, 150, 200}. For different input values (as presented in table 1 and 2) and different parameters values (presented above), the final fitness values are evaluated and compared with the results obtained by applying Round Robin (RR), an usual algorithm for scheduling, as presented in Table 3.

Table 3. Final results

FMS Scenario	CP/MP/IPD/FPD	Best GA fitness	Best RR fitness
FMS1	0.6/0.20/30/200	0.301	0.298
FMS2	0.8/0.20/20/150	0.412	0.415
FMS3	0.6/0.20/20/200	0.621	0.650
FMS4	0.6/0.20/10/200	0.705	0.700
FMS5	0.8/0.20/20/200	0.908	0.980

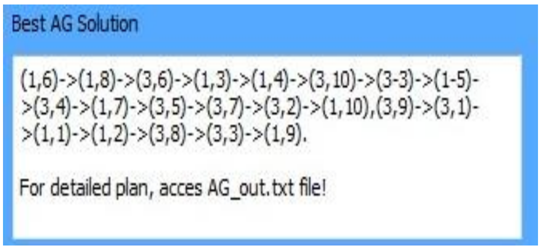

Fig. 2. Best AG Solution for FMS1

For example the best solution for FMS1 returned in the application interface is presented in figure 2. The detailed schedule including the accessing and release times on each machine is included in an attached text file. The final results presented in Table 3 indicate the fact that more accurate solutions can be obtained by increasing the mutation frequency so that the algorithm avoid the decoy of a local optimal solution and turns towards the global optimal solution.

Conclusions

The current paper presents a production scheduling problem approach using Genetic Algorithms. The challenge is to identify a schedule that satisfies all the constraints of the problems and respects all the systems' characteristics. The algorithm is tested using various sets of input data and parameters values in the context of different FMS scenarios. The solution quality is measured through a fitness function and the proposed implementation manages to obtain good results, by comparison to a classical scheduling algorithm (Round Robin). Furthermore, the quality of the obtained solutions seems to increase if the mutation rate in higher, because in this context it manages to avoid being trapped in a local optimum area.

References

1. Davis, L. – Job Shop Scheduling with Genetic Algorithms, *Proceedings of the Second International Confference on Genetic Algorithms*, Lawrence Eribaum Associates, Mahwah, NJ, 1985.
2. Golbers, D.E. – *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing, Co. Inc., 1989.
3. Ip, W.H., Li, Y., Man, K.F., Tang, K.S. – Multi-product planning and scheduling using genetic algorithm approach, *Comput. Ing. Engl.*, Vol. 38, 2004, pp. 283-296.
4. Jain, A., Meeran, S. – A state-of-art Review of Job Shop Scheduling Techniques, *European Journal of Operational Research*, 1999.

5. L a u r e c o , H . R . – Local Optimization ant the Job Shop Scheduling Problem, *European Journal of Operational Research*, 1995.
6. N a f i z , H . , L e e , L . S . – Algorithm on Single Machine Scheduling Problem to minimize Total Weight Completion Time, *European Journal of Scientific Research*, Vol. 35, No. 5, 2009.
7. P o n g c h a r o e n , P . , H i c k , C . , B r a i d e n , P . M . – The development of genetic algorithms for the finite capacity scheduling of complex products, with multiple levels of product structure, *European Journal of Operational Research*, Vol. 152, no. 1, 2004, pp. 215-225.
8. S a w i k , T . – *Production Planning and Scheduling in Flexible Assembly Systems,* Springer, 1999.
9. S r i n i v a s , M . K . T i w a r i , A l l a d a , V . – Solving the machine-loading problem in a flexible manufacturing system using a combinatorial auction-based approach, *International Journal of Production Research*, Vol. 42, no.9, 2004, pp 1879-1893.
10. S t e c k e , K . E . – Formulation and Solution on a Non-linear Integer Production Planning Problem for Flexible Manufacturing Systems, *International Journal of Management Science*, No. 29(3), 1983, pp. 273-288.
11. V a s s e s e n s , R . J . M . A s r t s , E . H . L . L e n s t r a , J . K . L J o b – Shop Scheduling by Local Search, *Informs Journal on Computing,* Vol. 8, No. 3, 1996.
12. Y u s o f , U . K . , B u d i a r t o , R . , D e r i s , S . – Contraints-Chromosome Genetic-Algorithm for Flexible Manufacturing Systems Machine-Loading Problem, *International Journal of Innovative Computing, Information and Control*, Vol. 8, No. 3(A), 2012, pp. 1591-1609.
13. T o a d e r , F . A . – Production Scheduling by Using ACO and PSO Techniques, *Proceedingc of The International Conference on Development and Application System*, 2014, pp.170-175.
14. Z h a n g , G , L i a n g , G , S h i , Y . – An effective genetic algorithm for the flexible job-shop scheduling problem, *Expert Systems with Applications*, Vol.38, Issue 4, 2011, pp.3563-3573.
15. J a h r o m i , M . H . M . A , N a j a f i , E . – Grouping genetic algorithm for multi-objective machine tool selection and operation allocation in a FMS, *TJEAS Journal*, 2012, pp.96-100.
16. L i h o n g , Q . , S h e n g p i n g , L . – An improved genetic algorithm for integrated process planning and scheduling, *The International Journal of Advanced Manufacturing Technology*, Vol. 58, Issue 5-8, 2011, pp.727-740.

Algoritm Genetic pentru Planificarea Producției în Sisteme Flexibile de Fabricație

Rezumat

Industria de fabricație trebuie să răspundă unei cereri a pieței in continuă creștere și schimbare și în acest context o planificare eficientă a producției este legată de eficiența utilizării resurselor prin reducerea pierderilor, prin scurtarea ciclului de producție și prin creșterea varietății de producție. Sistemele Flexibile de Fabricație (SFF) au ca obiectiv principal menținerea echilibrului dintre flexibilitate si productivitate, iar literatura de specialitate propune o serie de metode pentru a obține o programare a producției eficientă, ținând cont de o serie de criterii. În acest articol se propune un Algoritm Genetic (AG) implementat pentru un SFF cu scopul de a obține o planificare a producției care maximizează satisfacerea constrângerilor tehnologice si minimizează timpul de finalizare. Algoritmul propus este testat in condiții diferite în comparație cu un algoritm consacrat de planificare, iar rezultatele obținute și concluziile sunt prezentate în lucrare.