An Emergent Synthesis Approach to Simultaneous Process Planning and Scheduling

K. Ueda[1](1), N. Fujii[1], R. Inoue[2]

[1]Research into Artifacts, Center for Engineering (RACE), The University of Tokyo, Japan

[2]School of Engineering, The University of Tokyo, Japan

Abstract

Optimality of a process plan and a production schedule frequently conflict. It is difficult to determine a proper plan that meets both objectives simultaneously. This paper proposes a new simultaneous process planning and scheduling method to solve dilemmas posed by such situations using evolutionary artificial neural networks based on emergent synthesis. The effectiveness of the proposed method is confirmed by solving a benchmark problem, thereby demonstrating high productivity resulting from role-sharing among machines. Results also show that the proposed method is applicable to more realistic problems including larger volumes of products and production demand fluctuations.

Keywords:

Manufacturing system, Emergent synthesis, Evolutionary artificial neural network

1 INTRODUCTION

Process plans and production schedules play important roles in achieving high-productivity and low-cost manufacturing. Obtaining a proper plan that simultaneously accomplishes process planning and the scheduling objectives is not easy because their optimality can frequently conflict. The quality of the process plan can decrease when the quality of the production scheduling is preferred; on the other hand, the quality of the production schedule can be corrupted as the process plan is prioritized. It is difficult to obtain a proper process plan and a production schedule under such dilemma-like conditions using traditional planning methods, in which the process-plan optimization and production scheduling are determined sequentially.

New planning methods to integrate process planning and production scheduling have been proposed to resolve such dilemmas posed by process planning and scheduling needs. For example, Chryssolouris and Chan [1] and Zijm [2] introduced process plan alternatives and changed the process plan when the scheduling results are not feasible. Palmer [3] used simulated annealing techniques to integrate process planning and scheduling. Sadeh [4] attempted to implement communication functions between a process planning module and a production scheduling module to integrate them. However, because most of these methods repeat separated optimizations, they sometimes encountered problems by which the solutions do not converge but are instead periodic. A method that simultaneously forms process plans and production schedules using a genetic algorithm (GA) based method was proposed by Morad and Zalzala [5]. The proposed method can reportedly obtain a good solution to fulfill both objective functions of process planning and scheduling. Nevertheless, it remains difficult for the method to adapt to environmental changes such as altered production requirements. A method that is adaptable to environmental changes must be developed because market demands often cause turbulent fluctuations.

This paper describes a new method based on emergent synthesis [6]. It realizes simultaneous process planning and scheduling that resolves the dilemma between process planning and scheduling. In the proposed method, each machine has a learning unit developing evolutionary artificial neural networks (EANN) [7] proposed in the area of evolutionary robotics. Thereby, it can accomplish simultaneous process planning and scheduling as a result of local interactions among machines. It is desired that the proposed method be adaptable to environmental changes through re-learning performed by each machine after environmental changes.

2 SIMULTANEOUS PROCESS PLANNING AND SCHEDULING USING EANN

2.1 Emergent synthesis based approach

Emergent synthesis has been proposed as a system synthesis methodology by which system behavior emerges as a global structure that is built up through bottom-up and top-down dynamic processes. The emergent synthesis approach includes three synthesis problem classes: (Class I) the environmental description and specification are complete - the problem is described completely; (Class II) the environmental description is incomplete and the specification is complete; (Class III) both the environmental description and the specification are incomplete.

Optimization of process planning or scheduling is a typical example of a Class I problem. It is usually solved as a combinatorial optimization problem. However, obtaining optimal solutions is not easy because the problem can be NP-hard and the search space can easily explode. Furthermore, it is more difficult to optimize process planning and scheduling simultaneously under conditions in which production environments include fluctuations. Under such conditions, the problems can be regarded as Class II problems in which the system must adapt to environmental changes. The process plan and the schedule must be changed adaptively because of production demand fluctuations. Reportedly, emergent synthesis approaches are effective to resolve Class I problems and are essential to resolve Class II problems [8]. A new method, based on an emergent synthesis approach to simultaneous process planning and scheduling, is proposed by developing EANN in this study.

2.2 EANN based planning method

In the proposed method, process plans and production schedules are generated through interaction among local decisions of machine agents. A machine agent decision includes: (1) scheduling – selection of a product to process from waiting products in the machine's buffer; and (2) process planning – selection of a machine to be used for subsequent processing of the selected product. Figure 1 depicts a machine's decision about process planning and scheduling. These selections are performed simultaneously when a machine is about to start a process. This simultaneity enables the system to produce a process plan and schedule simultaneously. Each machine learns to make appropriate decisions that suit the state of the production floor using EANN, which is introduced into each machine. In the EANN, the ANN structure is determined through an evolutionary process: weight and threshold values of an ANN are encoded into a gene. These values are updated using GA. It is expected that an effective and adaptable learning system is realized by introducing EANN.

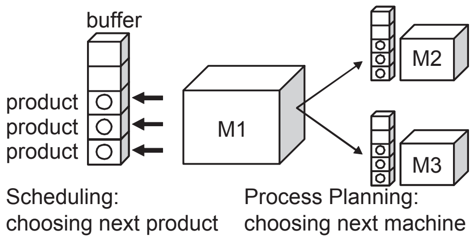

Figure 1: A machine agent producing a process plan and scheduling simultaneously

2.3 Problem descriptions

The proposed method is applied to a benchmark problem proposed by Sundaram and Fu [9], which is a job-shop type problem that involves five machines and five job types. Each product has four processes. Table 1 shows the process times (min) achieved by the machines. Each bracketed number is an identification number of the machine which performs the process. The benchmark problem objective is achieving the shortest make-span. In that problem, the make-span can reach 33 min when one job of each type is produced. In the next section, the proposed method is applied to a problem that produces one product of each type. Later, the proposed method is applied to more complex problems in which 20 products of each type are produced and also a problem including production demand changes.

Table 1: Production time used by each machine

Prod.	Process 1	Process 2	Process 3	Process 4
1	5(1), 3(2)	7(2)	6(3)	3(4), 4(5)
2	7 (1)	4(2), 6(3)	7(3), 7(4)	10 (5)
3	4(1),5(2)8 (3)	5(4)	6(4), 5(5)	4(5)
4	2(2), 6(3)	8(3)	3(3), 8(4)	7(4), 4(5)
5	3(1), 5(3)	7(3)	9(4), 6(5)	3 (5)

2.4 Setting of EANN

Figure 2 illustrates the structure of the artificial neural networks (ANN) used in this study. A three-layered feed-forward ANN is employed. Input information to the ANN includes the product types in each buffer, the occupation rate of other machines' buffers, and whether other machines are processing products or not. The input layer has 16 neurons. Outputs from the ANN are selection of a product from the buffer and selection of a machine to be used in the subsequent process: each machine forms the process plan and the schedule simultaneously. The output layer has five neurons; the hidden layer has eight.

The weight and threshold values of the ANN are encoded into a genetic string using real values. The number of bits in the gene is set as 318. A multiple-population parallel genetic algorithm [10] is used to maintain high variety among genes. The sub-population size is set to five. Migration occurs among sub-populations at intervals of 100 generations. These parameters are chosen based on the results of preliminary experiments. An elite strategy is also introduced.

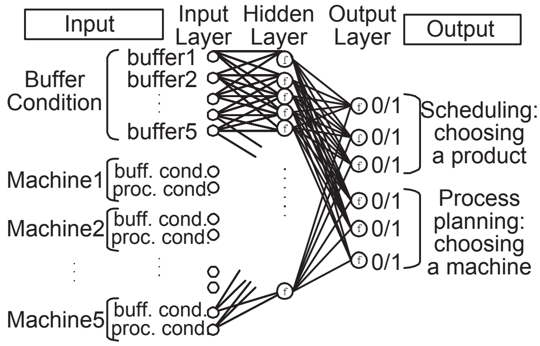

Figure 2: ANN configuration in a machine

The production result decoded from a gene is evaluated from the viewpoint of productivity. The production objective is set to minimizing the make-span. The fitness value of each gene is calculated as

$$f(t) = \begin{cases} \exp(\alpha * \dfrac{T_{threshold} - t}{T_{threshold} - T_{target}}) - P, & (t \le T_{threshold}), \\ 1, & (t > T_{threshold}) \end{cases} \quad (1)$$

where t denotes the make-span, T_{target} is the target value of the make-span, and α is a constant number. Also, P represents a punishment calculated by $P = \beta * erro\_no$, where *erro_no* is the frequency of selecting empty buffers by the machine. Finally, β is a constant. A positive evaluation value in the case of a make-span larger than $T_{threshold}$ is introduced to retain variety in the population so that genes having a lower evaluation value can remain.

3 SIMULATION RESULTS AND DISCUSSION OF A BENCHMARK PROBLEM

3.1 Settings of the simulations

The proposed simultaneous process planning and scheduling method developing EANN is applied to the benchmark problem as a preliminary experiment to confirm the feasibility of the proposed method. Only one job exists in each product type to be produced.

The evaluation parameters are set as follows: α = log10, β = 0.01, T_{target} = 33, and $T_{threshold}$ = 50. The parameters for GA are also set as follows: the maximal generation is set to 1000, the population size in a sub-population is 10, the elite size is one in a sub-population, and the mutation rate is set to 0.8.

3.2 Simulation results

Results of the computer simulations show that the make-span remains at the large value in the early stage of the simulation because all weight values and the threshold are set to random values in the initial condition. The make-span decreases concomitant with the simulation execution. Finally, the optimal value, 33, is achieved using the proposed methods shown in Table 2.