Multi-stage Parallel Machines and Lot-Streaming Scheduling Problems – A Case Study for Solar Cell Industry

Hi-Shih Wang, Li-Chih Wang, Tzu-Li Chen, Yin-Yann Chen, Chen-Yang Cheng

Multi-stage parallel machines and lot-streaming scheduling problems – A case study for solar cell industry

Hi-Shih Wang[1], Li-Chih Wang[2], Tzu-Li Chen[3], and Yin-Yann Chen[4],

Chen-Yang Cheng[5]

[1,2, 5]Department of Industrial Engineering and Enterprise information, Tunghai University
Taichung 40704, Taiwan, ROC.
.[3]Department of Information Management, Fu Jen Catholic University
New Taipei City 24205, Taiwan, ROC
[4]Department of Industrial Management, National Formosa University
Yunlin County 632, Taiwan, ROC
Corresponding E-mail: chengcy@thu.edu.tw

Abstract. This research focuses on a parallel machines scheduling problem considering lot streaming which is similar to the traditional hybrid flow shop scheduling (HFS). In a typical HFS with parallel machines problem, the allocation of machine resources for each order should be determined in advance. In addition, the size of each sublot is splited by parallel machines configuration. However, allocation of machine resources, sublot size and lot sequence are highly mutual influence. If allocation of machine resources has been determined, adjustment on production sequence is unable to reduce production makespan. Without splitting a given job into sublots, the production scheduling cannot have overlapping of successive operations in multi-stage parallel machines environment thereby contributing to the best production scheduling. Therefore, this research motivated from a solar cell industry is going to explore these issues. The multi-stage and parallel-machines scheduling problem in the solar cell industry simultaneously considers the optimal sublot size, sublot sequence, parallel machines sublot scheduling and machine configurations through dynamically allocating all sublot to parallel machines. We formulate this problem as a mixed integer linear programming (MILP) model considering the practical characteristics including parallel machines, dedicated machines, sequence-independent setup time, and sequence-dependent setup time. A hybrid-coded particle swarm optimization (HCPSO) is developed to find a near-optimal solution. At the end of this study, the result of this research will compare with the optimization method of mixed integer linear programming and case study.

Keywords: Hybrid flow shop scheduling, particle swarm optimization, solar cell industry, lot streaming

1 Introduction

Lately, many countries have focused their research and development efforts on sustainable energies such as wind, tidal, and solar energies. Among these, solar energy has attracted the greatest attention. Solar cell manufacturing follows the hybrid flow shop (HFS) mode, which is a flow shop mode incorporated with multiple processes and parallel machines. [1,2] proposed a flow shop mixed mode that consists of flow shop scheduling (FSS) and parallel machine scheduling (PMS)[3].

Initially, HFS could only match a single machine for one order. Later on, Chen and Lee (1999)[4] proposed a production environment wherein one order could plan mutiple machines (multiprocessor task) (Fig. 1.), and wherein a series of studies follows the same assumption that configure only when the machine resources for each order is known. In this same enviroment, [5] planned the allocation of machines for each order and worked out an optimal production sequence configured by various scheduling algorithms.

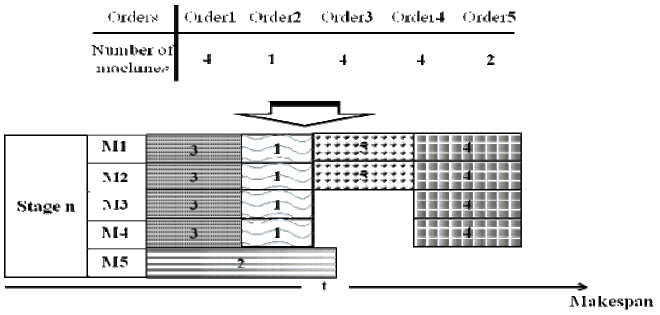

Fig. 1. Schematic diagram showing a single order being assigned to multiple machines (machine configuration is known)

The production environment nowadays is constantly changing, thereby generating more complicated environments. Therefore, the production environment has no way of finding an optimal scheduling combination through sequence adjustment in an existing machine configuration. Recently, the heated solar cell industry has been classified as an architecture of parallel machine resource in an HFS environment in academic research, but they are not identical. As a result, the configuration pattern of parallel machine for this order is not clear among line managers. Thus, the production sequence has to be planned using traditional dispatching rules such as the earliest due date, which cannot provide an optimal schedule plan. Hence, [6] created a schedule plan with an unknown machine configuration and order production to extend the HFS environment. Each order can plan a maximum number of machines in the process, and the number of machines is subject to dynamic adjustment. This plan proposes a scheduling solution for this problem and gives proper assessment to solve such kinds of production problems.

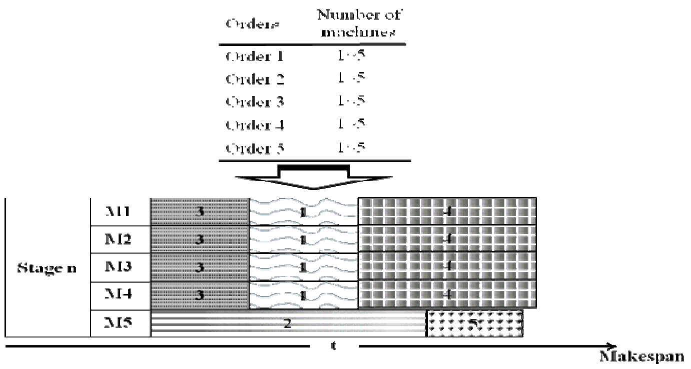

Fig. 2. Schematic diagram showing a single order being assigned to multiple machines (machine configuration is unknown)

Methods such as lot splitting have been developed and introduced into the manufacturing industry to shorten time of completion. Literature suggests that lot splitting can provide better performance ([7],[8]). With respect to the production architecture of parallel machine resource in an HFS environment, studies have paid little attention to variable lot splitting. Therefore, the current study proposes an actual instance-crystal silicon solar cell industry based on the architecture of HFS production to consider process properties such as parallel machine, special machine, setup time, etc., and to solve scheduling issues through the particle swarm optimization (PSO) with the batch, batch size, and sequence of order as unknown variables.

The aim of this research is to determine the job production sequence, number of sublots, and which machines these sublots should be assigned to. This study employs the production scheduling of crystalline silicon solar cells as a case study that focuses on the four characteristics to establish a suitable product planning and reduce the makespan time. The characteristics include parallel processing, dedicated machines, sequence-independent setup time, and sequence-dependent setup time. Due the computational complexity of the model, A hybrid-coded particle swarm optimization algorithm (HCPSO) was used to design to obtain the near-optimal scheduling configuration. Our preliminary computational study shows that the developed HCPSO not only provides good quality solutions within a reasonable amount of time but also outperforms the classic branch and bound method and the current heuristic practiced by the case company. The rest of the paper is organized as follows; Section 2 defines the multi-stage and parallel-machine scheduling problem in the solar cell industry; Section 3 develops a hybrid coded PSO algorithm to obtain the near-optimal solution; Section 4 addresses the excellent performance of the HCPSO algorithm through the computational study and Section 5 finally presents the concluding remarks.

2 Hybrid flow shop scheduling for solar cell manufacturing

The manufacturing of crystal silicon solar cells comprises six processes, and each process has its practical characteristics that influence the production schedule. The production characteristics are detailed below:

1. Dedicated machines: Crystalline silicon solar cells are basically of two types: single-crystal silicon solar cells and polysilicon solar cells. Manufacturing machines of both types of solar cells are the same. However, the difference lies in the texturing stage. The dedicated polysilicon machines use an acid texturing method for the polysilicon solar cells, whereas the single silicon dedicated machines use an alkaline texturing method for the single silicon solar cells. Therefore, the need for different numbers of dedicated machines for each of these processes is obvious due to the variance in capacity. This difference impacts the subsequent scheduling method in the job production process.
2. Parallel machine processing: Identical parallel machines are used in the manufacturing process of crystalline silicon solar cells. When the job demand is high, a job must be allocated to more than one machine, increasing the capacity and reducing the makespan required to complete the job.
3. Sequence independent setup time: In the printing stage, the electrodes are printed on both sides of the silicon that is used to collect and conduct the current flow. Depending on customers' requirements, different densities of printing designs are available. Because a few order can have the similarity design, the print setup are almost necessary for all kinds of orders, which is referred to as sequence independent setup time.
4. Sequence dependent setup time: Due to the number of electrodes on the surface, the crystalline silicon solar cell can be categorized as: 2 busbars and 3 busbars. In the testing stage, the measurement probe must be adjusted according to the number of electrodes in both busbar types. Therefore, the probe adjustment time will be affected by the job sequence, which influences the setup time. This is referred to as sequence dependent setup time. Setting the optimal production sequence to reduce the number of setups and shorten the overall completion time is the key focus of this restriction.

3 Hybrid-coded particle swarm optimization algorithm

In this section, a novel hybrid-coded particle swarm optimization algorithm (HCPSO) is designed to find the near-optimal solution through the evolutionary process because of the computational complexity of the proposed MILP model. The details of the elements are described as follows.

1. Particle representation

The decision of HFS problems must simultaneously determine the batch numbers and size of each manufacturing order and the batch sequence for all manufacturing orders in each stage. There are two parts in the HCPSO which is different from the traditional PSO. The first part called the master (the numbers and size of batch) particle, indicates the batch numbers of the each order and the each batch size. For each particle within first part, there exists a second part called the slave (batch sequence) particle, indicates the batch sequence decision using batch-based encoding.

2. Generate an initial population

In our HCPSO implementation, the initial population of master and slave particle is randomly generated.

3. Initialize the value of *Cr*

In PSO, the parameters w, r_1, r_2 are critical factors influencing the convergence level of the algorithm (Naka et al., 2003).The research in this paper displaces the random numbers of r 1 and r 2 with the *Cr*, which makes its convergence level better (Chuang et al., 2008b; Sun et al., 2011b). The equation to calculate *Cr* is as follows:

$$Cr(n+1) = k \times Cr(n) \times (1 - Cr(n)) \qquad (1)$$

In Eq. (1), *Cr*(*n*) represents the *Cr* of the *n* time; *k* represents the driving parameter, controlling the oscillation of *Cr*. When initializing *Cr*(0), the *Cr*(0) generated by random numbers can not equal {0, 0.25, 0.5, 0.75, 1} and *k* must equal to 4.

4. Update the inertia weight

Appropriate inertia weights enable a particle to have exploration capability in the initial period and better exploitation capability in the final period. A higher inertia weight implies larger incremental changes in velocity per iteration, and thus the exploration of new search areas for better solution. However, a smaller inertia weight signifies less variation in velocity, providing slower change in terms of fine tuning a local search. Therefore, it would be better that the searching process should start with a high inertia weight for global exploration, with the inertia weight decreasing to facilitate finer local explorations in later iterations. The equation to update the inertia weight adopted in this paper is proposed by Fan and Chiu (2007), nonlinearly decreasing weight method. In this equation, *t* is the iteration number; *w*(*t*) is the inertia weight of the *t* iteration.

$$w(t) = \left(\frac{2}{t}\right)^{0.3} \qquad (2)$$

5. Calculate the fitness value

Based on the known the numbers, size and sequence of batch, this step precedes the forward capacity allocation to calculate the fitness values (makespan) of all the particles.

6. Update particle best (*pBset*)

The *pBest* is the best position of each particle in its own searching process. During the iterations, the particle's fitness evaluation is compared with *pBest.* If the current value is better than *pBest,* then set *pBest* value equal to the current value.

7. Update global best (*gBset*)

Compare fitness evaluation with the population's overall previous best, *gBest*. If the current value is better than *gBest*, then update the current particle's value to *gBest*.

8. Update *Cr*, velocity and position of the particle

Assuming that the search space is D-dimensional, the *i*-th particle of the swarm is represented by a D-dimensional vector $X_i = (X_{i1}, X_{i2}, \dots, X_{iD})$ and the position change (velocity) of the *i*-th particle is $V_i = (V_{i1}, V_{i2}, \dots, V_{iD})$. The best particle of the swarm, that is, the particle with the best objective function value, is denoted by *gBest*. The best previous position of the *i*-th particle in its own searching trajectory is recorded and represented as *pBest*. This paper adopts the Eq.(3) to update the *Cr* and the velocities and positions of the particles are manipulated according to the following equations (the superscript *t* denotes the iteration):

$$V_{id}(t+1) = w \times V_{id}(t) + c_1 \times Cr(n) \times \left(pbest_{id}(t) - X_{id}(t)\right) \\ +c_2 \times (1 - Cr(n)) \times \left(gbest_d(t) - X_{id}(t)\right) \tag{3}$$

$$X_{id}(t+1) = X_{id}(\mathrm{t}) + V_{id}(t+1) \tag{4}$$

where i = 1, 2, y, N, and N is the size of the population; w is the inertia weight which was developed to better control exploration and exploitation; c_1 and c_2 are two positive constants, called the cognitive and social parameters respectively; and *Cr*(*n*) represents the *Cr* of the *n* time, as stated in 3. Eq. (3) is used to determine the *i*-th particle's new velocity, at each iteration, while Eq. (4) provides the new position of the *i*-th particle, adding its new velocity to its current position.

9. Determine whether the same optimal solution of the population which iterates *n* times exists

If yes, execute 10 and precede the boundary search. If not, skip to 11.

10. Boundary search

The boundary search aims to prevent that the current solution falls into the local optimum and enable it to avoid being a regional solution, and in turn to find the global optimum. The boundary search is to generate new particles for each dimension by random. The amount is 10% of the population, with which displace the worst 10% of the original population.

11. To decide wither the designated times of iteration are reached

The termination criterion of the HCPSO algorithm proposed in this paper is that when the number of iteration exceeds the designated maximum iteration times, terminate the algorithm. If it is not reached, return to 4.

4 Computational study

We test objective value for HCPSO algorithm. The objective values of all problems are shown in Table 1. From this table, we can observe that the value of average solution for HCPSO. This shows that the proposed HCPSO algorithm can find the near-optimal solution. In the large samples (from problem #6 to problem # 9), the developed HCPSO algorithm still generates better solutions for the large samples in the reasonable time. Consequently, from above analysis, our results claim that the proposed HCPSO algorithm not only provides the near-optimal solutions irrespective of the size of the sample data; it also generates the better solutions for any samples in which MILP algorithm cannot found any feasible solutions.

Table 1. The objective value for B&B

Problem Number	Problem Size (order,machine,stage)	Min. Solution	Average Solution	Max. Solution
1	(3, 3, 4)	46512	46767.6	46800
2	(3, 5, 4)	36000	37555.6	38448
3	(5 3, 4)	72000	72064.8	72648
4	(5, 5, 6)	60480	60761	61020
5	(5, 8, 6)	42120	42288.6	42480
6	(10, 8, 6)	78552	81453	82980
7	(10 10, 6)	66960	74760.6	78084
8	(20, 10, 8)	131439	135654.6	143280
9	(20, 15, 10)	106740	117176.4	128016

5 Summary

This paper presents a multi-stage and parallel-machine scheduling problem which is similar to the traditional hybrid flow shop scheduling (HFS) in the solar cell industry. The multi-stage and parallel-machines scheduling problem simultaneously determines order production sequence, multiprocessor task scheduling and optimal machine configuration through dynamically allocating all jobs to multiple machines under the minimization of the maximum makespan. A mixed integer linear programming model has been proposed, in consideration of many practical characteristics including hybrid flow shop, parallel machine system, specified machines, sequence-independent setup time, and sequence-dependent setup time. Because of the computational complexity, a hybrid approach based on the variable neighborhood search and particle swarm optimization (HCPSO) is developed to obtain the near-optimal solution. The computational study shows that the proposed algorithm could be more suitable and efficient for solving large size problems than the conventional B&B algorithm. Moreover, HCPSO also has better improvement than the current heuristic practiced by the case company based on realistic data. For the future research, other heuristic algorithm can be developed to efficiently attack large-scale instances and compare with the proposed algorithm.

6 References

1. Salvador MS (1973) A solution to a special class of flow shop scheduling problems. In: Elmaghraby SE, editor Symposium on the theory of scheduling and its applications Berlin: Springer:83-91
2. Salvador MS (ed) (1972) A solution to a special class of flow shop scheduling problems. Symposium on the theory of scheduling and its applications. Case Western Reserve University,
3. Ruiz R, Vázquez-Rodríguez JA (2010) The hybrid flow shop scheduling problem European Journal of Operational Research 205 (1):1-18
4. Chen J, Lee C-Y (1999) General Multiprocessor Task Scheduling. Naval Research Logistics 64 (1):57-74
5. Engin O, Ceran G, Yilmaz MK (2011) An efficient genetic algorithmnext term for previous termhybrid flow shop scheduling with multiprocessor task problemsnext term. Applied Soft Computing 11 (3):3056-3065
6. Chuang M-C (2011) A genetic algorithm for multi-stage parallel machines scheduling problems – A case study for solar Cell industry. Tunghai University, Taichung
7. Ranga VR HF, Duncan KHF, Jack CH (2000) Lot streaming in multistage production systems. International Journal of Production Economics 66 (2):199-211
8. Zhang WY, C.; Liu,J. & Linn, R. J. (2005) Multi-job lot streaming to minimize the mean completion time in m-1 hybrid flowshops. International Journal of Production Economics 96 (0):189-200