Journal of Convergence for Information Technology
Vol. 11. No. 8, pp. 207-211, 2021
e-ISSN 2586-4440
DOI : https://doi.org/10.22156/CS4SMB.2021.11.08.207

Minimizing the total completion time in a two-stage flexible flow shop

Suk-Hun Yoon
Professor, Department of Industrial and Information Systems Engineering Soongsil University

2 단계 유연 흐름 생산에서 평균 완료 시간 최소화 문제

윤석훈
숭실대학교 산업정보시스템공학과 교수

Abstract This paper addresses a two-stage flexible flow shop scheduling problem in which there is one machine in stage 1 and two identical machines in stage 2. The objective is the minimization of the total completion time. The problem is formulated by a mixed integer quadratic programming (MIQP) and a hybrid simulated annealing (HSA) is proposed to solve the MIQP. The HSA adopts the exploration capabilities of a genetic algorithm and incorporates a simulated annealing to reduce the premature convergence. Extensive computational tests on randomly generated problems are carried out to evaluate the performance of the HSA.

Key Words : Scheduling, Flexible Flow Shop, Total Completion Time, Mixed Integer Quadratic Programming, Simulated Annealing

요 약 이 논문은 단계 1에 기계 한 대, 단계 2에 2대의 병렬 기계가 있는 유연 흐름 생산 스케줄링 문제를 다룬다. 목적 함수는 평균 완료 시간을 최소화하는 것이다. 이 문제를 혼합 정수 2차 문제로 정식화하여 혼합 시뮬레이티드 어닐링을 이용하여 풀었다. 혼합 시뮬레이티드 어닐링은 유전자 알고리즘의 탐색 능력을 이용하고 시뮬레이티드 어닐링을 적용하여 너무 이른 수렴 현상을 줄이는 방법이다. 실험을 통하여 혼합 시뮬레이티드 어닐링의 성능을 평가하였다.

주제어 : 스케줄링, 유연 흐름 생산, 평균 완료 시간, 혼합 정수 2차 문제, 시뮬레이티드 어닐링

1. Introduction

A flexible flow shop consists of multiple stages in series with numbers of machines in parallel at each stage. Jobs must be processed only on one machine at each stage[1,2]. Ding et al.[3] considered a flexible flow shop scheduling system to minimize total tardiness and electric power costs simultaneously. They developed a hybrid particle swarm optimization algorithm for the problem. Gong et al.[4] presented a hybrid evolutionary algorithm to solve an energy-efficient flexible flow shop scheduling problem with worker flexibility, where the flexibility of machines and workers as well as the processing time, energy consumption and worker cost related factors were considered simultaneously. Ernst et al.[5] presented two

*Corresponding Author : Suk-Hun Yoon(yoon@ssu.ac.kr)
Received July 10, 2021
Revised July 31, 2021
Accepted August 20, 2021
Published August 28, 2021

integer linear programs for a two-stage flexible flow shop, where first- and second-stage machines formed disjoint pairs with a buffer. Ahonen and de Alvarenga[6] considered a reentrant and flexible flow shop problem where the processing times of the jobs at some stage might depend on the machine sequence the jobs took in the processing flow. They presented solution procedures using a mixed integer programming solver and simulated annealing and tabu search.

This paper presents a simulated annealing-ased methodology for a two-stage flexible flow shop scheduling problem, in which one machine is at stage 1 and two identical machines in parallel at stage 2. The objective is the minimization of the total completion time. In the next Section, the notations and assumptions are defined and a mixed integer quadratic programming (MIQP) model for the problem is provided. By assigning binary values to integer variables, the MIQP reduces to a linear programming that can be solved to obtain an optimum in a reasonable time. In Section 3, a hybrid simulated annealing (HSA) is developed to find the best solution for large size problems. In Section 4, the computational results of the extensive experiments are provided. The performance of the HSA is compared with that of GA. In Section 5, summary and conclusions are provided.

2. Notations and problem definition

There are n jobs that must be processed on a machine at stage 1 and any machine at stage 2 in series. Job j is available at time 0 and requires the processing time p_{ij} at stage i. Let C_{ij} be the completion time of job j at stage i and M a big number. Let x_j = 1 if job j is allocated to machine 1 at stage 2 and otherwise -1. Let y_{jl} = 1 if job j precedes job l at stage 1 and otherwise 0.

Then, the problem can be formulated by a mixed integer quadratic programming as follows:

$$\min \sum_{j=1}^{n} C_{2j}$$

s.t.

$$C_{2j} - C_{1j} \geq p_{2j}, \qquad j = 1,\ldots,n \qquad (1)$$

$$C_{1j} + My_{jl} - C_{1l} \geq p_{1j}, \qquad j,l = 1,\ldots,n \qquad (2)$$

$$C_{2j} + M(1 + y_{jl} - x_j x_l) - C_{2l} \geq p_{2j}, \qquad j,l = 1,\ldots,n \qquad (3)$$

$$y_{jl} + y_{lj} = 1, \qquad j,l = 1,\ldots,n \qquad (4)$$

$$C_{1j} \geq p_{1j}, \qquad j = 1,\ldots,n \qquad (5)$$

$$x_j = -1 \text{ or } 1, \qquad j = 1,\ldots,n \qquad (6)$$

$$y_{jl} = 0 \text{ or } 1, \qquad j,l = 1,\ldots,n \qquad (7)$$

Constraint set (1) ensures that a job cannot transfer to the next stage without completion. Constraint set (2) states that two jobs at stage 1 cannot be processed simultaneously. Constraint set (3) assures that two jobs cannot be processed at a time on the same machine at stage 2. Constraint set (4) implies that only permutation schedules can be considered for an optimal schedule. Constraint set (5) states that all jobs are available at time 0. Constraint sets (6) and (7) insure binary values of variables x_j and y_{jl}.

3. Hybrid Simulated Annealing

Simulated annealing (SA) is a widely used metaheuristic to create the process that is capable of escaping from local optima and carrying out a robust search for a feasible region[7,8]. In early iterations, SA often accepts solutions with higher objective values in order to explore large search space. The search process gradually converges to best solution by rejecting non-decreasing solutions with increasing probability[9,10]. Let z_c be an objective value for the current solution, z_n be an objective value for a candidate of the next solution, and T a parameter (called temperature) that affects acceptance of non-decreasing solutions[11-13]. Among the neighborhoods of the current solution, better solution is accepted for the next solution and a solution with higher objective value can be selected with the following probability [14]:

$$\Pr(acceptance) = e^{\frac{z_c - z_n}{T}} \quad (8)$$

The proposed HSA adopts three basic operators of GA (selection, crossover and mutation) and incorporates SA to enhance the exploration capabilities and restrain from the premature convergence of the GA. Solutions (individuals, sequences) are represented as chromosomes using binary coding for x_j and y_{jl}. For example, consider a three-job 2-stage flexible flow shop where the processing times of job 1, 2 and 3 at stage 1 are all 2 time units and 7, 6 and 4 time units at stage 2, respectively. Suppose that the job sequence at stage 1 is 1-2-3. If jobs 1 and 3 are transferred to machine 1 and job 2 to machine 2 at stage 2, then the solution can be represented by (x_1, x_2, x_3, y_{12}, y_{13}, y_{23}) = (1, -1, 1, 0, 1, 0). The Gantt chart of this schedule is shown as in Fig. 1 and the total completion time can be calculated as 32 (= 9 + 10 + 13).

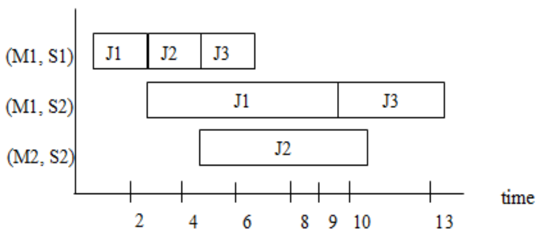


Fig. 1. A schedule for a three-job flexible flow shop

By assigning binary values to integer variables (x_j and y_{jl}), the MIQP reduces to the LP and thus, the total completion times can be obtained in polynomial time. The total completion times of individuals are sorted in descending order. Let l and $f(s)$ be the rank and the fitness of individual s, respectively. Then $f(s)$ = $2l/w(w + 1)$. The ratio of the fitness value of each chromosome to the total fitness value is used as the probability that the chromosome is selected in the selection process. The stochastic remainder selection procedure without replacement [15] is used to construct a mating pool. The individuals in the mating pool undergo crossover and mutation processes to produce offspring for the next generation. Whenever the best individual is not updated, SA is applied to the worst individual of the next generation to get the better one. This process of HSA continues until the predetermined number of generations.

3.1 Summary of Hybrid Simulated Annealing

Step 1 (Initialization)

Produce an initial population using a random number generator.

Step 2 (Evaluation and selection)

(a) Calculate the fitness values of individuals in the population.

(b) Use the stochastic remainder sampling without replacement to select individuals of the current population to form a mating pool.

Step 3 (Reproduction)

(a) Mate two individuals in the mating pool randomly to construct a couple until $w/2$ couples are made

(b) Apply the one-point crossover with a constant crossover rate to the couples.

(c) Apply the mutation with a constant mutation rate to the individuals that go through the crossover to w offspring.

(d) When the individual of lowest objective value is obtained, record this as best one and go to Step 5. Otherwise, go to Step 4.

Step 4 (Simulated Annealing)
Apply SA to the least fit individual to find the better one and go to Step 5.

Step 5 (Termination test)
If HSA reaches the predeternined number of generations, stop. Otherwise, go to Step 2.

4. Computational Experiments

The MIQP and HSA were coded in Visual C++ and ran on an Intel Core i7 CPU @3.4 GHz PC with the software IBM ILOG CPLEX.Test problems were generated randomly using the pseudo-random numbers. Processing times were generated according to the integer uniform distributions provided in [1, 100]. The size of test problems is defined by the number of jobs (5, 10, 15 and 20). Experiments were composed of two parts: the preliminary test and main test. In the preliminary test, five problems of different job sizes were solved to find the best parameter set of the HSA. The parameter set includes population size (N_p), number of generations (N_g) and mutation rate (p_m). The best result was obtained with population size of 100, 50 generations and mutation rate of 0.01. The geometric cooling schedule of ten temperatures (10 iterations in each temperature) and decreasing ratio of 0.5 were used for the main test.

Ten test problems of different job sizes were solved by the HSA with the best parameter set found in the preliminary test. For small size problems (5 and 10 jobs), the results of HSA were compared with the optimal solutions obtained by the MIQP. HSA achieved optimal solutions for all 10 small size problems. The GA solved the large size problems (15 and 20 jobs). The results of the GA and HSA are shown in Table 1. The HSA provides 8.05% better solutions than the GA on the average, which implies that the HSA helps to avoid the premature convergence.

Table 1. Results for medium and large size flexible flow shop problems

No. of	GA	HSA	%Dev
Jobs	z_g	z_h	$(z_g-z_h/z_g)\times100$
15	757	708	6.93
20	1,007	923	9.17

5. Conclusions

This paper addressed the scheduling problem to minimize the total completion time in a 2-stage flexible flow shop with one machine at stage 1 and two identical machines in parallel at stage 2. The problem was formulated as the mixed integer quadratic programming. Since the problem is NP-hard, an exact algorithm to obtain an optimal solution requires enormous efforts as the problem size grows. For this reason, the HSA has been developed to obtain the near optimal solution, which combines the diverse explore of the GA and the intensive searcg of the SA.

The MIQP can be applied to an extended version of the flexible flow shop scheduling problem with numbers of stages that consist of two identical parallel machines. When the number of machines at stages are more than three, the number of integer variables grows fast and thus, permutation representations of the problem might be an alternative option for the metaheuristics. The various implementation of the hybrid metaheuristic algorithm such as Tabu search and ant colony optimization is an interesting future research for the more complex flexible flow shop environment.

The performance of the HSA depends on the parameter sets for the SA and the GA module. However, finding best combinations of the parameters costs huge efforts and the use of the experimental designs might be considered to save times. Especially, how much high an initial temperature of the SA has to be set for a global optimum is an important research topic.

REFERENCES

[1] M. L. Pinedo. (2016). *Scheduling: Theory, Algorithms, and Systems* (5th Ed.). New York : Springer.

[2] K. R. Baker & D. Trietsch. (2019). *Principles of Sequencing and Scheduling* (2^{nd} Ed.). New York : Wiley.

[3] J. Ding, S. Schulz, L. Shen, U. Buscher & Z. Lü. (2021). Energy aware scheduling in flexible flow shops with hybrid particle swarm optimization. *Computers and Operations Research, 125.* 105088
DOI : 10.1016/j.cor.2020.105088

[4] G. Gong, R. Chiong, Q. Deng, W. Han, L. Zhang, W. Lin & K. Li. (2020). Energy-efficient flexible flow shop scheduling with worker flexibility. *Expert Systems with Applications*, 141. 112902
DOI : 10.1016/j.eswa.2019.112902

[5] A. Ernst, J. Fung, G. Singh & Y. Zinder. (2019). Flexible flow shop with dedicated buffers. *Discrete Applied Mathematics, 261,* 148-163.

[6] H. Ahonen & A. G. de Alvarenga. (2017). Scheduling flexible flow shop with recirculation and machine sequence-dependent processing times: formulation and solution procedures. *International Journal of Advanced Manufacturing Technology, 89,* 765-777.

[7] M. Gendreau & J.-Y. Potvin (eds) (2018). *Handbook of Metaheuristics* (3^{rd} Ed.). New York : Springer.

[8] M. Almaraashi, R. John, A. Hopgood & S. Ahmadi (2016). Learning of interval and general type-2 fuzzy logic systems using simulated annealing: Theory and practice. *Information Sciences*, *360,* 21-42.

[9] M. T. Assadi & M. Bagher. (2016). Differential evolution and Population-based simulated annealing for truck scheduling problem in multiple door cross-docking systems. *Computers & Industrial Engineering, 96,* 149-161.

[10] R. Bellio, S. Ceschia, L.D. Gaspero, A. Schaerf & T. Urli. (2016). Feature-based tuning of simulated annealing applied to the curriculum-based course timetabling problem. *Computers & Operations Research, 65,* 83-92.

[11] V. Courchelle, M. Soler, D. González-Arribas & D. Delahaye. (2019). A simulated annealing approach to 3D strategic aircraft deconfliction based on en-route speed changes under wind and temperature uncertainties. *Transportation Research Part C: Emerging Technologies, 103,* 194-210.

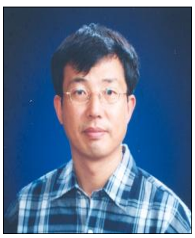
[12] M. L. D. Dias & A. R. R. Neto. (2017). Training soft margin support vector machines by simulated annealing: A dual approach. *Expert Systems with Applications, 87,* 157-169.

[13] A. M. Fathollahi-Fard, K. Govindan, M. Hajiaghaei-Keshteli & A. Ahmadi. (2019). A green home health care supply chain: New modified simulated annealing algorithms. *Journal of Cleaner Production, 240.* 118200.
DOI : 10.1016/j.jclepro.2019.118200

[14] A. A. Kida, A. E. L. Rivas & L. A. Gallego. (2020). An improved simulated annealing-linear programming hybrid algorithm applied to the optimal coordination of directional overcurrent relays. *Electric Power Systems Research, 181.* 106197.
DOI : /10.1016/j.epsr.2020.106197

[15] D. E. Goldberg. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Reading : Addison Wesley.

윤 석 훈(Suk-Hun Yoon) [정회원]



· 1988년 2월 : 서울대학교 산업공학과(공학사)
· 1995년 12월 : NC A&T State University, 산업공학과(공학석사)
· 2002년 5월 : Pennsylvania State University, 산업공학과(공학박사)
· 2002년 9월 ~ 현재 : 숭실대학교 산업정보시스템공학과 교수
· 관심분야 : 최적화, 메타휴리스틱, 스케줄링
· E-Mail : yoon@ssu.ac.kr