MATEC Web of Conferences **44**, 02026 (2016)
DOI: 10.1051/matecconf/20164402026

Study on disruption management scheduling problem of flow shop under supply chain environment

Hong Guang Bo [1], Long Long Li[1], Yu Liu[2,3], Jia Heng Wang[1]
[1]*Institute of Production Operation and Logistics Management, Dalian University of Technology, Dalian 116023, China*
[2]*CSR Qingdao Sifang Co., Ltd. Qingdao 266111, China*
[3]*Crrc Central Research Institute Qingdao 266111, China*

Abstract. This paper presents a disruption scheduling model for an environment of proportional two-machine no-wait flow shop. To achieve the objects of minimization of weighted sum of makespan and minimization of weighted sum of tardiness, we introduce a revised PSO algorithm which is designed with a neighborhood search structure. According to the experiment, the effectivity of the method proposed is proven.

1 Introduction

In reality, unexpected events are often inevitable to cause disruption to flow shop system. They lead to failure of system control. Therefore, we need to carry out a scheduling recovery plan to minimize the lost.

Scheduling recovery problem is practically useful in production and processing. Many of them are proven NP-hard problems [1] and the problem in this research is one of them. There are abundant researches about the global static scheduling and periodic rolling scheduling [2]. However, random disruption is more common in production and processing practical environment. In disruption management problem, we should pay more attention to the combination of the initial scheduling objective and the new scheduling plan instead of global optimization. Lee [3] proposes two methods for unfinished jobs: one is to arrange them to other machines with extra cost and another is to wait till the recovery. Liou[4] have studied the disruption management problem in a single machine. Bo [5] establishes the scheduling model based on SPT rule.

This paper studies the disruption management recovery problem in the environment of proportional two-machine no-wait flow shop. We propose a disruption management model pred-mgt, and solve it with the HDPSO algorithm. A case study has been done to testify our research.

2 Problem description

In this research, we assume that that in the supply chain, there is a supplier known as M1 and a manufacturer known as M2. A job is processed only once at either M1 or M2, and it must be processed continuously in the supply chain. M1 or M2 can process only one job at a time, and a job's process time at M1 is proportional to that at M2. The process time of any job at M2 is longer than that at M1. The job being disrupted has to be reprocessed from the start of the supply chain. The processing environment can be named as proportional two-machine no-wait flow shop environment. The following notations are used : $J=\{1,2,\ldots,j,\ldots n\}$ （$n>1$ ）means job set to be processed, $W=\{\omega_1,\omega_2,\ldots,\omega_j,\ldots\omega_n\}$ （$n>1$ ）means job weight of priority $M=\{M_1,M_2\}$ supply chain member set, v_i means the speed of the jobs processed at M_i, p_{ij} means process time of job j at M_i, s_{ij} means start time of job j at M_i, C_{ij} means finish time of job j at M_i, C_j means finish time of job j at the whole supply chain(i.e. C_{2j}), π means feasible process schedule, $\prod$ means the set of feasible process schedule.

2.1 The initial scheduling scheme

The objective of the initial scheduling scheme is the minimization of weighted sum of makespan, which is shown as $\sum_{j=1}^{n}\omega_j C_j$. It has been proven that with the *WSPT* rule, the objective can be accomplished. Therefore, the initial scheduling can be described as $F|nwt|F(\pi)$, in which the best process schedule is $\bar{\pi}$, and the objective value is $F(\bar{\pi})=\sum_{j=1}^{n}\omega_j C_j$.

2.2 Disruption

In reality, unexpected events may cause disruption in supply chain. Some of them like power outages can impact all the members of the supply chain. Under this

Li Longlong: Lee_dlut@163.com

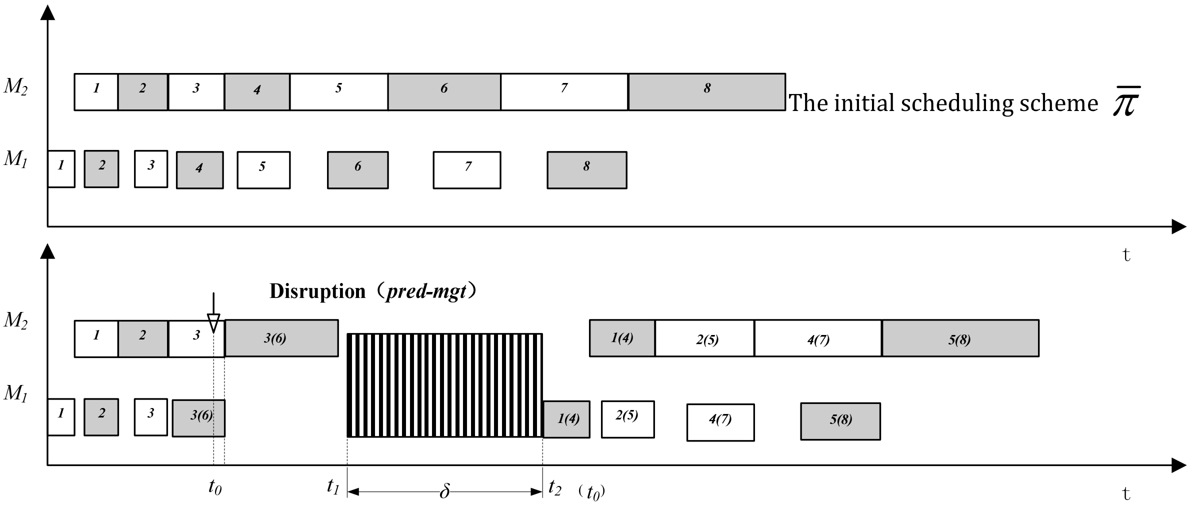

Figure.1. Disruption schematic diagram in supply chain

circumstance, in this research, the best process schedule $\overline{\pi}$ for the initial scheduling scheme will be not the best even infeasible. Hence, we need to reschedule the remaining jobs $J'=\{1,2,\cdots,j,\cdots,n'\}$ （$1\le n'\le n$）.

In figure 1, the first half shows the initial scheduling scheme $\overline{\pi}$, while the second half shows the disruption management strategy with the interference of disruption. The disruption $\Delta M\,|\,[t_1,t_2]$ (briefly notated as ΔM) occurs during $t_1 \sim t_2$, and the duration is notated as $\delta = t_2 - t_1$. The start time after the disruption is set as t_0 . We can see that job 4 and the jobs after are divided into two parts, one is processed before the disruption while the other after. Obviously, the condition $t_1 > min\{t_0+p_{1j}+p_{2j}|j\in J'\}$ should be met, so that certain job or jobs can be scheduled in the window between job 3 and the disruption. The remaining jobs defined as a job set J' are renumbered from 1 to n' according to the $WSPT$ rule.

In this situation, two objectives are taken consideration. One is the initial objective, the minimization of weighted sum of makespan $F_I(\pi')=\sum_{j=1}^{n}\omega_j C_j$, and the other is the recovery objective, the minimization of weighted sum of tardiness $F_{II}(\pi')=\sum_{j=1}^{n'}\omega_j T_j$.

In summary, the scheduling problem with disruption can be shown as following with three-parametric method:

$$F_2\left|nwt-sc, uni-dif, \Delta M, prep-mgt\right|F_I(\pi'), F_{II}(\pi')$$

3 Modeling and solution

3.1. Problem Model

The remaining jobs influenced by the disruption should be divided into two parts. Therefore, we define J^{t_1} and J^{t_2} as job set of jobs which are processed before t_1 and after t_2 respectively. Obviously we know $J^{t_1}\in J', J^{t_2}\in J'$. We define $C_j^{t_1}$ and $C_j^{t_2}$ as completion time of job j if it's scheduled before t_1 or after t_2 . As a result, the problem is translated into a problem aiming to allocate job j before t_1 or after t_2 . The problem's disruption management model $pred\text{-}mgt$ is presented below:

$$pred-mgt: \min_{i\in\{1,2\}, j\in J'}\{F_I(\pi')=\sum_{j=1}^{n}\omega_j C_j, F_{II}(\pi')=\sum_{j=1}^{n'}\omega_j T_j\}$$

$$s.t.\quad C_j = C_j^{t_1}\cdot x_j + C_j^{t_2}\cdot(1-x_j), \quad C_j^{t_1}\le t_1$$

$$s_{2j}\ge C_{1(j\text{-}1)}+p_{1j}, \quad \forall j\ge 2$$

$$C_{ij}=s_{ij}+p_{ij}$$

$$\left(s_{ij_1}\ge C_{ij_2}\right)\vee\left(s_{ij_2}\ge C_{ij_1}\right)$$

$$p_{1(j\text{-}1)}\le p_{1j}, \quad \forall j\ge 2, \quad j\in J^{t_1}\ or\ j\in J^{t_2}$$

$$p_{1x}/p_{2x}=p_{1y}/p_{2y}, p_{1x}\le p_{2y}, \forall x,y\in N'$$

$$x_j=\begin{cases}1 & j\in J^{t_1}\\ 0 & j\in J^{t_2}\end{cases}, \quad j=1,2,\cdots,n'$$

3.2 A solution based on HDPSO algorithm

In this research, we propose the Hybrid Discrete Particle Swarm Optimization (HDPSO) algorithm to solve the $pred\text{-}mgt$ model. The basic thought of the HDPSO algorithm is described as follow:

Initialization: The position of every particle in the population is represented by a n' -dimensional vector $X_i^{(R)}=[x'_{i,1},x'_{i,2},\cdots,x'_{i,n'}]$. $x'_{i,k}$ （$k\in\{1,2,\cdots,n'\}$）is randomly initialized with an even-distributed random number in (0,1).

Particle evolution: The update formula of particle position consists of $\bar{P}_i(t)$ (pbest), $\bar{P}_g(t)$ (gbest) and f_N (neighborhood search strategy). The particle evolution strategy is designed as follow:

$$X_i(t+1)=f_N\left(\bar{M}_{best}\oplus\left(f_C\left(c_{p3}\otimes\left(f_M\left(c_{p1}\otimes\bar{P}_i(t)\right),f_M\left(c_{p2}\otimes\bar{P}_g(t)\right)\right)\right)\right)\right)$$

In the evolution strategy:

① $f_M\left(c_{p1}\otimes\bar{P}_i(t)\right)$ means that $\bar{P}_i(t)$ mutates with a probability of $0\le c_{p1}\le 0.5$: A random number *rand* is generated at first, and if $rand<c_{p1}$, we choose two random elements of $\bar{P}_i(t)$ and swap them with each other, otherwise, the position of particle remains unchanged $f_M\left(c_{p1}\otimes\bar{P}_i(t)\right)=\bar{P}_i(t)$.

② $f_M\left(c_{p2}\otimes\bar{P}_g(t)\right)$ means that $\bar{P}_g(t)$ mutates with a probability of $0\le c_{p2}\le 0.5$: A random number *rand* is generated at first, and if $rand<c_{p2}$, we choose two random elements of $\bar{P}_g(t)$ and swap them with each other, otherwise, the position of particle remains unchanged $f_M\left(c_{p2}\otimes\bar{P}_g(t)\right)=\bar{P}_g(t)$.

③ $f_C\left(c_{p3}\otimes\left(f_M\left(c_{p1}\otimes\bar{P}_i(t)\right),f_M\left(c_{p2}\otimes\bar{P}_g(t)\right)\right)\right)$ means that $f_M\left(c_{p1}\otimes\bar{P}_i(t)\right)$ and $f_M\left(c_{p2}\otimes\bar{P}_g(t)\right)$ crossovers with a probability of $0.5\le c_{p3}\le 1.0$: A random number *rand* is generated at first, and if $rand<c_{p3}$, we choose an random interval $[\sigma,\delta]$ （$0<\sigma,\delta\le n$） as the part that they cross with each other, otherwise, the position of particle remains unchanged $f_C\left(c_{p3}\otimes\left(f_M\left(c_{p1}\otimes\bar{P}_i(t)\right),f_M\left(c_{p2}\otimes\bar{P}_g(t)\right)\right)\right)=f_M\left(c_{p1}\otimes\bar{P}_i(t)\right)$

④ f_N means local search to $f_C\left(c_{p3}\otimes\left(f_M\left(c_{p1}\otimes\bar{P}_i(t)\right),f_M\left(c_{p2}\otimes\bar{P}_g(t)\right)\right)\right)$ with a step length of $\bar{M}_{best}$, in which $\bar{M}_{best}=\beta\cdot(1/M)\cdot\sum_{k=1}^{M}d_H\left(X_i(t),\bar{P}_k(t)\right)\cdot ln(1/u)$.

3.3 Neighborhood search design

To make up the weakness of the local search of HDPSO algorithm, we propose the neighborhood search operator.

There are two neighborhood structures in this research, insert and swap.

Insert: Choose a job processed before t_1 and a job processed after t_2 randomly. Insert the latter one into the position of the former one, and the jobs between them all move afterwards for one position.

Swap: Choose a job processed before t_1 and a job processed after t_2 randomly. Swap them with each other.

After the implementation of the two neighbourhood structures, if in the new order, the completion time of jobs before t_1 is less than t_1 and the objective evaluation becomes better, the operation of insert and swap is allowed, otherwise it's prohibited.

4 Algorithm experiment

4.1. Experiment design

In this research, processing speed of M_1 and M_2 is set as $v_2=v_1/5$. The number of initial jobs ready to be processed is 50. Jobs are sorted with the WSPT rule. p_{1j} is set as a random production time array, so p_{2j} is easy to get. The weight of jobs is w set as a random weight array. During the case, six group experiments are carried out. They are differentiated with the different disruption window. For each group, 10 times of experiments are carried out. To demonstrate the usefulness of neighborhood search, we compare the HDPSO algorithm above with the HDPSO1 algorithm (a special HDPSO algorithm without neighborhood search).

The parameters of the algorithm: the number of particles in the population is 40. The probability of particle mutation is $c_{p1}=c_{p2}=0.2$, while the probability of particle crossover is $c_{p3}=0.8$. The algorithm convergence scaling factor is $\beta=1$. The number of iterations in one experiment is 60.

Algorithms are coded and performed with $C\#$ language in $VisualStudio2012$. Running environment is $Inter\ Core\ \ i3-2120@3.30GHz\ /4GDDR3/Windows7$ 64 bit SP1.

4.2. Experiments analysis

In this study, we use six classic indicators to evaluate the performance of the algorithm. They are ONVG, CM, D_{av} and $D_{\max}$, TS, MS and AQ. The experimental data is shown in Table.1.

For ONVG, HDPSO algorithm has more pareto solutions than HDPSO1 algorithm. For CM, the pareto solutions of HDPSO algorithm can dominate that of HDPSO1 algorithm. For D_{av} and $D_{\max}$, HDPSO algorithm is better than HDPSO1 algorithm in terms of average distance and minimum distance. For MS, the coverage of the optimal pareto frontier of HDPSO algorithm is better than that of HDPSO1 algorithm.

Table.1 Algorithms performance indicator comparison

Indicators		(1)ONVG		(2)CM		(3)-1 D_{av}		(3)-2 D_{max}		(4)TS		(5)MS		(6)AQ	
Algorithms		HDPSO	HDPSO1	HDPSO	HDPSO1	HDPSO	HDPSO1	HDPSO	HDPSO1	HDPSO	HDPSO1	HDPSO	HDPSO1	HDPSO	HDPSO1
Experiment1 [300,350]	mean	43.5	5.7	0.9690	0	0.0125	0.0197	0.1026	0.2929	13.7558	13.6945	0.8636	0.2472	17763.4	20104.75
	best	52	7	1	0	0.0042	0.0082	0.051	0.1983	8.0766	3.2353	1.05325	0.5561	17734.15	19811.07
	worst	35	3	0.8333	0	0.0256	0.0267	0.1366	0.5064	19.4817	25.6991	0.6760	0.0619	17797.91	20424.45
Experiment 2 [300,320]	mean	45.7	6.3	0.9357	0	0.0124	0.0226	0.1003	0.2854	12.23895	15.5095	0.9507	0.2917	17466.08	19583.53
	best	50	10	1	0	0.0029	0.0096	0.0471	0.2079	9.9048	5.8462	1.1087	0.4621	17412.37	19352.88
	worst	37	4	0.5	0	0.0308	0.0329	0.1549	0.3501	15.1915	25.2460	0.7798	0.1476	17611.54	20167.61
Experiment 3 [496,530]	mean	45.6	5.7	0.8980	0.0026	0.0096	0.0114	0.0725	0.3296	13.2403	14.4857	0.7319	0.2387	16209.42	19368.17
	best	72	7	1	0.0263	0.0020	0.0039	0.0251	0.1766	7.4479	4.2795	0.9025	0.4313	15986.1	19040.85
	worst	37	3	0.6666	0	0.0440	0.0155	0.1102	0.4925	16.6139	27.3881	0.5910	0.0208	16418.6	19624.56
Experiment 4 [496,510]	mean	44.8	5.7	0.8748	0.0123	0.0124	0.0144	0.0982	0.3237	17.3647	12.4398	0.7159	0.2657	15979.77	18940.11
	best	77	9	1	0.0526	0.0520	0.0263	0.1517	0.4757	29.5629	30.1311	1	0.5659	16179.57	19496.28
	worst	37	3	0.5714	0	0.0037	0.0067	0.0492	0.1595	10.2750	1.8966	0.4803	0.0729	15753.22	17998.38
Experiment 5 [800,820]	mean	46.5	6.3	0.9421	0	0.0107	0.0195	0.0926	0.4061	18.1912	15.6393	0.7187	0.3569	15073.64	18383.1
	best	53	9	1	0	0.0066	0.0105	0.0444	0.2827	13.2861	9.4082	0.9543	0.6413	14792.15	17951.21
	worst	31	4	0.8	0	0.0150	0.0317	0.1453	0.4766	29.0093	40.5226	0.4157	0.1155	15333.57	18889.59
Experiment 6 [760,820]	mean	46.5	6.3	0.9421	0	0.0107	0.0195	0.0926	0.4061	18.1912	15.6393	0.7187	0.3569	15073.64	18383.1
	best	53	9	1	0	0.0066	0.0105	0.0444	0.2827	13.2861	9.4082	0.9543	0.6413	14792.15	17951.21
	worst	31	4	0.8	0	0.0150	0.0317	0.1453	0.4766	29.0093	40.5226	0.4157	0.1155	15333.57	18889.59

For AQ, HDPSO algorithm gives better consideration of approximation and dispersion than HDPSO1 algorithm. To sum up, HDPSO algorithm has a better comprehensive performance than HDPSO1 algorithm. In other words, neighborhood search structure has a great improvement of the algorithm to solve this problem.

5. Conclusion

In this study, a recovery model for disruption in proportional two-machine no-wait flow shop environment has been developed, in order to minimize the weighted sum of makespan and the weighted sum of tardiness. It’s solved using the HDPSO algorithm which is revised from PSO algorithm. Moreover, a neighborhood search structure with insert and swap is developed to improve the performance of the algorithm.

The environment and disruption mentioned in our study is less practical than that in real life like that disruptions do occur synchronously and so on. These conditions need our further research in the future.

References

1. Pinedo M. Scheduling theory, algorithm and systems. [M]. Fourth edition, New York : Springer Science + Business Media,LLC, 2012.
2. Stefan Voß,Andreas Witt. Hybrid flow shop scheduling as a multi-mode multi-project scheduling problem with batching requirements: A real-world application[J]. International Journal of Production Economics. 2005 (2)
3. Lee C Y,Joseph Y T,Leung,Yu G. Two Machine Scheduling under Disruptions with Transportation Considerations[J].Journal of Scheduling, 2006,9(1) :35-48.
4. Liou Tian Shy,Wang Mao Jun. Ranking fuzzy numbers with integral value[J]. Computer and Mathematics with Application, 2005,49: 1731- 1753.
5. BO Hong-guang ;PAN Yu-tao ;MA Xiao-yan. Disruption Management for Production Rescheduling in Two-machine No-wait Flow Shop [J].Operation Research and Management Science.2013,04:111-119+125.

Acknowledgements

This work was supported by the National Science and Technology Support Program (2015BAF08B02),the National Natural Science Foundation of China (71172137), and the Fundamental Research Funds for the Central Universities (DUT14RW101)