View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Elsevier - Publisher Connector

Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Procedia Computer Science 20 (2013) 102 – 107
Procedia
Computer Science

Complex Adaptive Systems, Publication 3
Cihan H. Dagli, Editor in Chief
Conference Organized by Missouri University of Science and Technology
2013- Baltimore, MD

Effective Estimation of Distribution Algorithm for Stochastic Job Shop Scheduling Problem

Xinchang Hao[a], Lin Lin[b,c], Mitsuo Gen[c,d]*, Katsuhisa Ohno[e]

[a]*Waseda University, Kita-Kyushu, Japan,* [b]*Dalian University of Technology Dalian, China*
[c] *Fuzzy Logic Systems Institute Kitakyushu, Japan,* [d]*National Tsing Hua University, Taiwan,* [e]*Aichi Institute of Technology Nagoya, Japan*

Abstract

This paper propose an effective estimation of distribution algorithm (EDA), which solves the stochastic job-shop scheduling problem (S-JSP) with the uncertainty of processing time, to minimize the expected average makespan within a reasonable amount of calculation time. With the framework of proposed EDA, the probability model of operation sequence is estimated firstly. For sampling the processing time of each operation with the Monte Carlo methods, we use allocation method to decide the operation sequence then the expected makespan of each sampling is evaluated. Subsequently, updating mechanism of the probability models is proposed with the best solutions to obtain. Finally, for comparing with some existing algorithms by numerical experiments on the benchmark problems, we demonstrate the proposed effective estimation of distribution algorithm can obtain acceptable solution in the aspects of schedule quality and computational efficiency.

© 2013 The Authors. Published by Elsevier B.V. Open access under CC BY-NC-ND license.
Selection and peer-review under responsibility of Missouri University of Science and Technology
Keyword: Estimation of Distribution Algorithm; Manufacturing Management; Stochastic Job Shop Scheduling Problem

1. Introduction

Over the past sixty years, a great number of researches have been conducted on Job-shop Scheduling Problem (JSP), which is one branch of the scheduling problem and highly popular in the manufacturing industry. JSP is one of the famous combinatorial optimization problems as NP-hard under the precedence and resource constraints [1,2]. For conventional job shop scheduling problem, there is often making the assumptions in traditional machine scheduling theory is that all time parameters are known exactly and in deterministic values. However, there are often uncertainties in manufacturing systems. These uncertainties may stem from a number of possible sources [3]: operation may take more or less time than originally estimated, moreover the resources may become unavailable, due dates may have to be changed, or new orders may have to be incorporated, etc.

At present, the common mathematic methods for modeling scheduling problem with uncertainties are stochastic programming, fuzzy programming, rough sets, grey programming and interval theory [4]. In stochastic programming, the parameters are initially described in terms of probability distributions, and the problem is named as the stochastic scheduling [5].

In real-world problem, most of the job shop scheduling problems is the stochastic scheduling problems. As one of the newest issues, more and more attention is spent on the problem with random processing time. As a result, in the last several decades, a significant amount of results have been achieved on Stochastic Job Shop Scheduling Problem

* Corresponding author. Tel.: +81-948-24-2771; Fax: +81-948-24-3002.
E-mail address: gen@flsi.or.jp

1877-0509 © 2013 The Authors. Published by Elsevier B.V. Open access under CC BY-NC-ND license.
Selection and peer-review under responsibility of Missouri University of Science and Technology
doi:10.1016/j.procs.2013.09.246

(S-JSP).

Meanwhile, there are some novel intelligent evolutionary computation methods are carried out. R. Tavakkoli-Moghaddam et al. [6] proposed a hybrid method using a neural network approach and a simulated annealing algorithm in 2 stages, in order to produce the optimal/near-optimal solution. B. Liu, L. Wang and Y. Jin [7] proposed an approach named PSOSAHT, which is hybrid with simulated annealing (SA) and hypothesis test (HT), for stochastic flow shop scheduling with uncertain processing time. R. Zhou, A.Y.C. Nee and H.P. Lee [8] proposed an ant colony optimization algorithm (ACO) with different levels of machine utilizations, processing time distributions, and performance measures. J. Gu, X. Gu and M. Gu [3] proposed a Novel Parallel Quantum Genetic Algorithm (NPQGA) for the stochastic Job Shop Scheduling Problem with the objective of minimizing the expected value of makespan. M. Gholami and M. Zandieh [9] integrated simulation into genetic algorithm to the dynamic scheduling of a flexible job shop with the objectives of minimizing expected makespan and mean tardiness. D. Lei [10] developed an efficient decomposition-integration genetic algorithm (DIGA) to minimize the maximum fuzzy completion time. S. Horng, S. Lin and F. Yang [11] proposed an evolutionary algorithm ESOO as embedding evolutionary strategy (ES) in ordinal optimization (OO), to solve for a good enough schedule with the objective of minimizing the expected sum of storage expenses and tardiness penalties.

Intelligent manufacturing scheduling based on meta-heuristics, such as GAs, SA, ACO and Particle Swarm Optimization (PSO), have become some of the common tools for finding satisfactory solutions. Recently, there are growing interests in stochastic optimization methods called Estimation of Distribution Algorithms (EDAs) that build and sample explicit probabilistic model for the distribution of promising candidate solutions found so far and use the constructed model to guide further search behavior.

In this paper, we propose an effective EDA which solves the stochastic job shop scheduling problem (S-JSP) with the uncertainty of processing time, to minimize the expected average makespan within a reasonable amount of calculation time. With the framework of EDA, the probability model of operation sequence is formulated. By sampling the processing time of each operation with the Monte Carlo methods, we use allocation method to decide the operation sequence and then the expected makespan of each sampling is calculated. The remainder of this paper is organized as follows: Section 2 provides a review of the S-JSP; Section 3 presents the proposed EDA approach in the detail; Section 4 provides experimental comparisons that apply the EDA approach for analyzing and solving several S-JSP; and finally, Section 5 offers a conclusion.

2. Stochastic Job Shop Scheduling Problem

In order to solve a job shop scheduling problem in stochastic and static environments. It assumes that the probability distribution of the processing time is known in advance. The realized outcome of a random processing time of operation only gets to be known at the completion of the processing. In this paper, we use a pure integer programming model to transmute the processing times in term of stochastic variable. It assumes that the probability distribution of the processing time is known in advance. The stochastic job shop scheduling problem (S-JSP) can be formulated as an extended version of JSP. The stochastic expected value model of S-JSP may be formulated as follows: The makespan is the maximum completion time of jobs and objective is to find a schedule that minimizes the expected value of makespan $C_{\max}$

Indices:

i, k: the index of jobs; $i, k = 1 \ldots J$

j, h: the index of operations; $j, h = 1 \ldots N$

m: the index of machines; $m = 1 \ldots M$

Parameters:

J: the number of jobs

N: the number of operations

M: the number of machines

o_{ij}: the operation j of the job i

ξp_{ijm}: the random processing time of operation o_{ij} on machine m, a stochastic variable, subjected to independent ormal distribution

Ξ: a positive number big enough as a penalty factor.

***Decision Variables*:**

$x_{ikm} = 1$ if job i precedes job k on machine m; 0 otherwise.
$y_{ijm} = 1$ if it is available to process operation o_{ij} on machine m; 0 otherwise.
ξs_{ijm}: the starting time operation o_{ij} on machine m, a stochastic variable
ξc_{ijm}: the completion time operation o_{ij} on machine m, a stochastic variable
.

$$\min \quad C_{\max} = \max_{i=1,\dots,J} \left\{ \max_{j=1,\dots,N} \left\{ \max_{m=1,\dots,M} \xi c_{ijm} \right\} \right\} \tag{1}$$

s. t.

$$\sum_{m=1}^{M} y_{ijm} \le 1, \qquad \forall i, j \tag{2}$$

$$\sum_{m=1}^{M} x_{ikm}\, y_{ijm} = 1, \qquad \forall i, j, k \tag{3}$$

$$\sum_{m=1}^{M} \xi s_{ijm} + \xi p_{ijm}\, y_{ijm} \le \sum_{m=1}^{M} \xi s_{i(j+1)m}, \qquad \forall i, j \tag{4}$$

$$\sum_{i=1}^{J}\sum_{j=1}^{N} (\xi s_{ijm} + \xi p_{ijm}\, y_{ijm}) \le \sum_{i=1}^{J}\sum_{k=1}^{J}\sum_{h=1}^{N} (\xi s_{khm}\, x_{ikm} + \Xi(1\text{-}x_{ikm})), \qquad \forall m \tag{5}$$

$$\xi c_{ijm} \ge 0,\ \xi s_{ijm} \ge 0,\ x_{ikm}, y_{ijm} \in [0,1], \quad \forall i, k, j, m \tag{6}$$

where, the equation (2) shows that only one operation can be in each sequence on a machine. The equation (3) guarantees that each operation for each job must be allocated to just one machine in a sequence. The equation (4) guarantees the operation precedence sequences for each job. The equation (5) shows that the processing time of each of operation does not have any overlap with any other. Equation (6) represents the nonnegative restrictions.

3. Proposed Estimation of Distribution Algorithm

EDAs is a class of population-based optimization algorithm that extracts statistical information from the population of solutions, which uses the estimated statistical information to generate new solutions instead of the crossover and mutation operators [12]. The algorithm starts by generating a population solution. A set of solutions (promising data) is selected from the population using a selection method, and the promising data is used to estimate the probability model. Finally, the new candidate solutions are incorporated into a solution pool, which keeps these individuals contribute to the makeup of promising data. The iteration will continue until the predefined termination criteria is met. The pseudo-code for the proposal is presented in Fig. 1. Hao et al reported a cooperative EDA for solving the simultaneous multiple resources scheduling problem by the semiconductor final test scheduling [14].

For traditional EAs, the representation of a chromosome for an individual is generated by mapping the decision space into the search space or directly encoding the decision variables. Each position in a probability vector indicates the distribution of probability regarding each variable. When prior knowledge of distribution is not assumed, the domain of discrete variable X is a set of predefined values (x). The distribution of random variable X has the same equal probability; the initialization is as following:

$$P_{t=0}(X) = \frac{1}{|X|} \tag{7}$$

where $|X|$ denotes the number of values in the set of domain X.

After the initialization of EDA, probability sample new alternative solutions, and the new solutions are evaluated according to a specific system objective. EDA collects all new alternative solutions and replaces the inferior solutions in the promising data. The probability distribution of X can be estimated as follows:

$$B_t(X = x) = \frac{\mathrm{N}(X = x) + 1/|X|}{prSize + 1/|X|} \tag{8}$$

where $\mathrm{N}(X = x)$ denotes the number of instances in promising solutions with variable X = x, and represents the low bound to the probability of X.

The distribution probability of X in the probability vector is learned toward the estimated distribution of promising data, as follows:

```
procedure: EDA-main routine
  input: problem data, parameters
 output: the best solution S_best
begin
    initialzation:
        step1:  t ← 0;
        step2:  initialize population Pop(t) by encoding and probability model P(t);
        step3:  evaluate Pop(t) by decoding and keep the best solution S_best;
    Optimization:
      while( not meeting termination criterion)
          step4:  subPop = select(Pop(t));
          step5:  P(t+1) = estimate(supPop, P(t));
          step6:  newPop = create(subPop,P(t+1));
          step7:  Pop(t) = reproduce(Pop(t), newPop);
          step8:  evaluate Pop(t) by decoding and update the best solution S_best;
         step10:  t ← t+1;
      end;
      output the best solution S_best;
end;
```

Fig. 1 Pseudo-code for EDA

$$P_{t+1}(X=x)=(1-\alpha)P_t(X=x)+\alpha B_t(X=x) \quad (9)$$

where α denotes the learning rate from the current promising solutions; in particular, for α = 1, the probability distribution is completely reconstructed by the current promising solutions.

To maintain the diversity of sampling, the distribution probability of X is updated toward the estimation distribution. The distribution can be tuned with probability pm of the mutation, and the mutation is performed using the following definition:

$$P_{t+1}(X=x)=\frac{P_t(X=x)+\lambda_m}{\sum_{x'\in X\setminus\{x\}}\max(P_t(x)-\lambda_m/(|X|-1),\varepsilon)+(P_t(X=x)+\lambda_m)} \quad (10)$$

where λ_m is the mutation shift that controls the amount for mutation operation, and ε is a small probability value to avoid the negative probability value.

4. Experiments and Discussion

4.1. Application of EDA for S-JSP

The representation of the operation sequence uses job-based encoding [2], and the length of the chromosome equals the total number of operations. The job number denotes the operation of each job, and the l-th occurrence of a job number refers to the l-th operation in the sequence of this job. For a job-based operations sequence vector $v_1=[3,1,3,1,2,2,3,1,2]$ (shown in Fig. 2), the operations sequence can be interrupted as follows: (3,1), (1,1), (3,2), (1,2), (2,1), (2,2), (3,3), (1,3), (2,3).

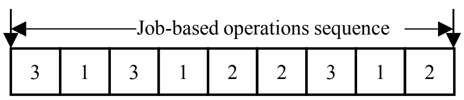

Fig.2 Illustration of the representation of a solution for S-JSP

Let η_{jl} be the number of times that job i appears before or in position l in the promising data D. It denotes the importance of the order of jobs. μ_{il} is the number of times that job i appears immediately after job i' when job i' is in position l-1. μ_{il} indicates the importance of the similar blocks of jobs in the promising data D. Then, the probability for positioning job i in the l-th position of the offspring for generation t is determined by

$$p_t(i,l) = \beta M(\eta_{il}) + (1-\beta)M(\mu_{il}) \tag{11}$$

where $M(\eta_{il}) = 1 - \exp(-\eta_{ij} / \sum_{i' \in D} \eta_{i'j}) / (\sum_{i'' \in D} 1 - \exp(-\eta_{i''j} / \sum_{i' \in D} \eta_{i'j}))$

$$M(\mu_{il}) = 1 - \exp(-\mu_{ij} / \sum_{i' \in D} \mu_{i'j}) / (\sum_{i'' \in D} 1 - \exp(-\mu_{i''j} / \sum_{i' \in D} \mu_{i'j}))$$

$M(\eta_{il})$ and $M(\mu_{il})$ are completeness measures for importance of the order jobs and similar blocks respectively. β is behavior coefficient, which is used to adjust the preference on which covering percentage we want to have a good discrimination.

To examine the practical viability and efficiency of the proposed EDA, we designed a numerical study to compare EDA with efficient algorithms from previous studies. The proposed EDA was compared with CCQGA[13] based on a set of simulation data of testing standard FT benchmark problems—FT06, FT10, and FT20. In order to ensure the fairness of comparison, The mean of normal distribution comes from the processing time of deterministic benchmark problems and the variance is generated from uniform distribution $U[0, 1]$. The experiments were conducted on a personal computer with an Intel Core I5 CPU at 2.8 GHz and 2 GB RAM. The parameters and strategies of related algorithms are categorized in Table 1.

In order to evaluate the performance of a given algorithm for S-JSP, the following two measures, which introduced by Gu et al [13], is adopted: The first metric is called "expected makespan", which is used for evaluating the quality of the solution. This metric reflects the performance of algorithms in a statistical sense. The formulations could be given as

$$\mathrm{E}(C_{\max}) = (1/N_{\mathrm{run}}) \times \sum_{k=1}^{N_{\mathrm{run}}} H_i \tag{12}$$

Where N_{run} is the number of simulations, and H_i is the makespan value at the k-th run of the method ($i = 1, \ldots, Nb$).

The second metric is called "standard deviation", which indicates relative deviation degree of the solution. The formulations could be given as

$$\sigma = \sqrt{(1/N_{\mathrm{run}}) \times \sum_{i=1}^{N\mathrm{run}} (H_i - \mathrm{E}(C_{\max}))^2} \tag{13}$$

4.2. Results and Discussion

Performance: Table 2 shows that EDA algorithms exhibited superior performance to CCQGA n all experiments. In contrast to the evolutionary operator of CCQGA, EDA uses the estimated probability distributions of decision variables relating to operation sequence and machine setup planning. It provides a prediction mechanism on the variant of decision variables. As standard deviation shown in the Table. 2, EDA achieved superior stability to random strategy-based algorithms, although the accuracy of prediction was affected by the promising solutions.

Table. 1 The parameters of CCAGA and EDA for S-JSP

	CCQGA	**EDA**
Iteration	1000	1000
Population	100(50,50)	100
Selection	roulette	tournament(k)
Strategy	co-evolutionary	-
Operators	CycleCrossover(P_c) Mutation(P_m)	Sampling Improvement(P_i)
Parameters	P_m= 0.10 P_c= 0.80 promisingRate = 0.5	samplingRate = 0.4 promisingRate = 0.5 α = 0.02 β = 0.02 P_i = 0.4 k = 2

Table. 2 Expected value and standard deviation of makespan of CCQGA and EDA

	FT06		FT10		FT20	
	CCQGA	EDA	CCQGA	EDA	CCQGA	EDA
Min $C_{\max}$	54.76	54.25	1008.70	998.47	1270.84	1265.05
Max $C_{\max}$	55.65	55.50	1074.90	1063.49	1329.19	1324.70
$\boldsymbol{E}(C_{\max})$	55.15	55.02	1054.52	1043.29	1314.25	1304.35
σ	0.36	0.28	18.08	15.13	20.02	16.77

***Computation Cost*:** The computational costs of evolutionary-based algorithms mainly depend on the number of fitness evaluations. The difference of time complexity between CCQGA and the proposed approach mainly relies on the operators. EDA samples a new candidate solution and improved the current candidate solution according to probability distribution. Moreover, it estimates the univariate margin distribution of decision variables using the promising data. The average CPU times of CCQGA with cooperative scheme on the FT06, FT10 and FT20 problems were 174.28, 683.61 and 789.43 s, respectively. The computational times of EDA conducted on FT06, FT10 and FT20 problems were 165.15, 648.79 and 749.64 s respectively. The results show that CEDA achieved similar computational efficiency to CCQGA.

5. Conclusion

This paper presents an effective estimation of distribution algorithm (EDA), which solves the stochastic job shop scheduling problem (S-JSP) with the uncertainty of processing time. It minimizes the expected average makespan within a reasonable amount of calculation time. With the framework of the proposed EDA, the explicit probability model of the operation sequence is estimated on the distribution of good solutions found so far and use the constructed model to guide further search behavior. The sampling operator based on the probability model achieves better convergence and stability than the conventional operator such crossover and mutation. In our future work, further experiments will be conducted to determine the accuracy of the proposed EDA in response to variations among the parameters. Furthermore, we will extend EDA to adapt to multiple-objective optimizations.

Acknowledgments: This work is partly supported by the Grant-in-Aid for Scientific Research (C) of The Japan Society of Promotion of Science (JSPS) No. 24510219.0001, National Science Council (NSC 101-2811-E-007-004, NSC 102-2811-E-007-005) and the Fundamental Research Funds (Software+X) of Dalian University of Technology (No. DUT12JR05, No. DUT12JR12.

References

1. E.L. Lawler, J.K. Lenstra, A.R. Kan, D.B. Shmoys, Sequencing and scheduling: Algorithms and complexity, *Handbooks of Operations Res. & Management Sci.* 4 (1993) 445–522.
2. M. Gen, R. Cheng, *Genetic algorithms and engineering optimization*, John Wiley & Sons, 2000.
3. W. Herroelen, R. Leus, Project scheduling under uncertainty: Survey and research potentials, *Eur. J. Operational Res.* 165 (2005) 289–306.
4. J. Gu, X. Gu, M. Gu, A novel parallel quantum genetic algorithm for stochastic job shop scheduling, *J. Math. Anal. Appl.* 355 (2009) 63–81.
5. P. Kall, S.W. Wallace, *Stochastic programming*, John Wiley and Sons Ltd, 1994.
6. R. Tavakkoli-Moghaddam, F. Jolai, F. Vaziri, P.K. Ahmed, A. Azaron, A hybrid method for solving stochastic job shop scheduling problems, *Appl. Math. Comput*ation 170 (2005) 185–206.
7. B. Liu, L. Wang, Y. Jin, Hybrid particle swarm optimization for flow shop scheduling with stochastic processing time, in: *Computer Intelligent Security,* Springer, 2005: pp. 630–637.
8. R. Zhou, A.Y.C. Nee, H.P. Lee, Performance of an ant colony optimisation algorithm in dynamic job shop scheduling problems, *Int. J. Prod. Res*. 47 (2009) 2903–2920.
9. M. Gholami, M. Zandieh, Integrating simulation and genetic algorithm to schedule a dynamic flexible job shop, *J. Intelligent Manuf.* 20 (2009) 481–498.
10. D. Lei, A genetic algorithm for flexible job shop scheduling with fuzzy processing time, *Int. J. Prod. Res*. 48 (2010) 2995–3013.
11. S.-C. Horng, S.-S. Lin, F.-Y. Yang, Evolutionary algorithm for stochastic job shop scheduling with random processing time, *Expert Syst. Appl.* 39 (2012) 3603–3610.
12. P. Larrañaga, J.A. Lozano, *Estimation of distribution algorithms*: *a new tool for evolutionary computation*, Springer, 2002.
13. J. Gu, M. Gu, C. Cao, X. Gu, A novel competitive co-evolutionary quantum genetic algorithm for stochastic job shop scheduling problem, *Computers & Oper. Res.* 37 (2010) 927–937.
14. X-C Hao, J-Z Wu, C-F Chien, M. Gen, The cooperative estimation of distribution algorithm: a novel approach for semiconductor final test scheduling problems, *J. Intelligent Manuf.*, DOI 10.1007/s10845-013-0746-x, (2013) 13pp.