Computers & Operations Research 36 (2009) 2638–2646

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/cor

An estimation of distribution algorithm for minimizing the total flowtime in permutation flowshop scheduling problems

Bassem Jarboui[a], Mansour Eddaly[a], Patrick Siarry[b,*]

[a]FSEGS, route de l'aéroport km 4, Sfax 3018, Tunisia
[b]LiSSi, Université de Paris 12, 61 avenue du Général de Gaulle, 94010 Créteil, France

ARTICLE INFO

Available online 21 November 2008

Keywords:
Estimation of distribution algorithm
Variable neighbourhood search
Scheduling
Permutation flowshop
Flowtime

ABSTRACT

In this work we propose an estimation of distribution algorithm (EDA) as a new tool aiming at minimizing the total flowtime in permutation flowshop scheduling problems. A variable neighbourhood search is added to the algorithm as an improvement procedure after creating a new offspring. The experiments show that our approach outperforms all existing techniques employed for the problem and can provide new upper bounds.

1. Introduction

In a permutation flowshop scheduling problem (PFSP), there is a set of n jobs that must be processed on a set of m machines in the same order, such that each job is processed on machine 1 in first place, machine 2 in second place,…, and machine m in the last place. Gupta and Stafford [1] have presented a review of developments in flowshop scheduling over the last fifty years. They have listed the assumptions regarding the flowshop scheduling problem in general and the permutation flowshop scheduling problem in particular. The processing times are fixed, known in advance and have non-negative values. All jobs are available for processing at time 0 and all machines are available over the scheduling period. In addition, each machine can process at most one job and each job can be processed on at most one machine. Besides, no pre-emption is allowed, i.e., once the processing of a job on a machine has started, it must be completed without interruption. The most common criteria are the makespan and the total flowtime. This paper addresses the total flowtime (*TFT*) performance measure. This criterion measures the time a job stays in the system, minimizing that time leads to maximizing the utilization of resources. Concerning its complexity, the PFSP has been proved to be NP-complete in the strong sense for more than two machines (Garey et al. [2]).

Diverse approaches were proposed to solve this problem with respect to *TFT* criterion, including exact algorithms such as branch and bound algorithm (Chung et al. [3], Velde [4], Bansal [5], Ignal and Schrage [6]), constructive heuristics (Woo and Yim [7], Framinan and Leisten [8], Liu and Reeves [9], Ho and Gupta [10], Framinan et al. [11], Rajendran [12], Allahverdi and Aldowaisan [13], Li et al. [14]) and metaheuristics like genetic algorithm (GA) (Vempati et al. [15], Zhang et al. [16]), ant colony optimization (ACO) (Gajpal and Rajendran [17], Rajendran and Ziegler [18]) and particle swarm optimization (PSO) (Jarboui et al. [19], Tasgetiren et al. [20]).

Estimation of distribution algorithm (EDA) was introduced by Mühlenbein and Paaß [21]. It constitutes a new tool of evolutionary algorithms (Larranaga and Lozano [22]) based on the probabilistic model learned from a population of individuals. Starting with a population of individuals (candidate solutions), generally generated randomly, this algorithm selects good individuals with respect to their fitness. Then a new distribution of probability is estimated from the selected candidates. Next, new offspring is generated from the estimated distribution. The process is repeated until the termination criterion is met.

In order to improve the quality of EDA solution, it is recommended to use a local search algorithm (Lozano et al. [23]). We propose to use the variable neighbourhood search (VNS) (Mladenović and Hansen [24], Hansen and Mladenović [25]) to improve the performance of our EDA.

This paper is organized as follows: Section 2 presents a formulation of the permutation flowshop scheduling problem. Section 3 describes the basic estimation of distribution algorithm. The EDA for the PFSP is presented in Section 4. Section 5 presents the hybrid EDA. The computational results are presented in Section 6 and a conclusion is given in Section 7.

2. The permutation flowshop scheduling problem (PFSP)

In a PFSP, there is a set of n jobs to be processed through a set of m machines, where each job $j (j = 1, 2, \ldots, n)$ passes through

* Corresponding author.
E-mail addresses: bassem_jarboui@yahoo.fr (B. Jarboui), eddaly.mansour@gmail.com (M. Eddaly), siarry@univ-paris12.fr (P. Siarry).

0305-0548/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2008.11.004

the machines $1, 2, \ldots, m$ in that order, without interruption of the processing (no pre-emption). Referring to the notation of Graham et al. [26] ($\alpha/\beta/\gamma$), the PFSP for *TFT* criterion is denoted by *F/permu/TFT*. Let p_{ij} be the processing time for the job j on the machine $i (i = 1, 2, \ldots, m)$, and C_{ij} denote the completion time of job j on machine i. Then $C_{[j]i}$ is the completion time of the job scheduled in the jth position in the sequence on machine i. $C_{[j]i}$ is computed as: $C_{[j]i} = p_{[j]i} + \max\{C_{[j]i-1}, C_{[j-1]i}\}$. So, we can obtain the *TFT* as follows: $TFT = \sum_{j=1}^{n} C_{[j]m}$.

3. Estimation of distribution algorithm (EDA)

In 1996, a new tool of evolutionary algorithms, called "Estimation of Distribution Algorithm" (Mühlenbein and Paaß [21], Larranaga and Lozano [22]), was proposed. Unlike other approaches of evolutionary algorithms, EDA uses neither crossover nor mutation. Therefore, it generates new offspring according to a probabilistic model learned from a population of parents. The main steps of the canonical EDA are described as follows (Lozano et al. [23]): first, a random initial population is commonly used. Second, a subpopulation of Q parent individuals is selected with a selection method based on the fitness function. Third, the probability of distribution of the selected parents is estimated by a probabilistic model. Fourth, new offspring are generated according to the estimated probability. Finally, some individuals in the current population are replaced with new generated offspring. These steps are repeated until one stopping criterion is met. In the combinatorial context, several EDA applications were developed, such as knapsack problem, travelling salesman problem, clustering and jobshop scheduling problems (Larranaga and Lozano [22]).

4. Our proposed EDA for PFSP

For solving the PFSP with respect to the *TFT* minimization, our proposed solution algorithm is based on the EDA. A new way for the design of the probabilistic model is proposed and the framework of the algorithm is introduced hereafter for discussion.

4.1. Solution representation and initial population

The well-known job based encoding scheme is frequently used in the literature of the PFSP, so it is also used in our algorithm. In this representation, the jth number in the permutation denotes the job located in position j. In order to guarantee the diversification in the population, we use an initial random population of P individuals, uniformly distributed.

4.2. Selection

In our algorithm we adopted the selection procedure of Reeves [27] for solving the flowshop scheduling problem. We describe this procedure as follows:

(i) first, for each individual p, calculate the fitness value $f(p) = 1/TFT(p)$; (ii) second, the individuals of the initial population are sorted in ascending order according to their fitness, i.e. the individual with a higher *TFT* value will be at the top of the list. Finally, the selection of parents is made relatively to the probability:

$$prob(r) = \frac{2r}{P(P+1)}$$

where r is the rank of the rth individual in the sorted list.

4.3. Probabilistic model and creation of new individuals

The probabilistic model constitutes the main issue for EDA and the performance of the algorithm is closely related to it (Lozano et al. [23]), the best choice of the model is crucial. This step consists in building an estimation of distribution for the subset of Q selected individuals. In our algorithm, we determine the estimation of distribution model while taking into account both the order of the job in the sequence and the similar blocks of jobs presented in the selected parents.

Let:

- η_{jk} be the number of times of appearance of job j before or in the position k in the subset of the selected sequences augmented by a given constant δ_1. The value of η_{jk} refers to the importance of the order of the jobs in the sequence.
- $\mu_{j[k-1]}$ be the number of times of appearance of job j immediately after the job in the position $k-1$ in the subset of the selected sequences augmented by a given δ_2. $\mu_{j[k-1]}$ indicates the importance of the similar blocks of jobs in the sequences. In such way, we prefer to conserve the similar blocks as much as possible.

We note that δ_1 and δ_2 are two parameters used for the diversification of the solutions. Indeed, we employed these parameters in order to slow down the convergence of the algorithm.

- Ω_k: the set of jobs not already scheduled until position k.

We define π_{jk} the probability of selection of the job j in the kth position by the following formula: $\pi_{jk} = \eta_{jk} \times \mu_{j[k-1]}/\sum_{l \in \Omega_k} (\eta_{lk} \times \mu_{l[k-1]})$. According to this probability, for each position k, we select a job j from the set of not already scheduled jobs in the sequence of a new individual.

4.4. Replacement

Replacement is the last phase in the EDA, it consists in updating the population. Therefore, at each iteration, O offspring are generated from the subset of the selected parents. There are many techniques for deciding if the new individuals will be added to the population.

In our algorithm, we compare the new individual with the worst individual in the current population. If the offspring is best than this individual and the sequence of the offspring is unique, then the worst individual is removed from the population and will be replaced with the new individual.

4.5. Stopping criterion

The stopping condition indicates when the search will be terminated. Various stopping criteria may be listed, such as maximum number of generations, bound of time, maximum number of iterations without improvement, etc. We set a maximum number of iterations and a maximal computational time in our algorithm.

5. Hybrid EDA for PFSP (EDA–VNS)

Aiming at improving the performance of EDA and preventing it from being stuck into a local optimum, a successful way is to hybridize it with local search methods (Lozano et al. [23]). We propose to apply a VNS algorithm (Mladenović and Hansen [24], Hansen and Mladenović [25]) as an improvement procedure after the creation of a new individual.