Engineering Applications of Artificial Intelligence 62 (2017) 80–95

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

An effective multi-start multi-level evolutionary local search for the flexible job-shop problem☆

S. Kemmoé-Tchomté[a], D. Lamy[b], N. Tchernev[b,*]

[a] CRCGM EA 3849: Centre de Recherche Clermontois en Gestion et Management Clermont Auvergne University, Clermont-Ferrand, France
[b] LIMOS UMR 6158: Laboratoire d'Informatique de Modélisation et d'Optimisation des Système Clermont Auvergne University, Clermont-Ferrand, France

ARTICLE INFO

Keywords:
Flexible Job-shop
Metaheuristics
GRASP
Multi-level ELS
Neighbourhood structures

ABSTRACT

In this paper, an improved greedy randomized adaptive search procedure (GRASP) with a multi-level evolutionary local search (mELS) paradigm is proposed to solve the Flexible Job-shop Problem (FJSP). The FJSP is a generalisation of the well-known Job-Shop Problem with the specificity of allowing an operation to be processed by any machine from a given set. The GRASP metaheuristic is used for diversification and the mELS is used for intensification. Four different neighbourhood structures are formalised. A procedure for fast estimation of the neighbourhood quality is also proposed to accelerate local search phases. The metaheuristic has been tested on several datasets from the literature. The experimental results demonstrate that the proposed GRASP-mELS has achieved significant improvements for solving FJSP from the viewpoint of both quality of solutions and computation time. A comparison among the proposed GRASP-mELS and other state-of-the-art algorithms is also provided in order to show the effectiveness and efficiency of the proposed metaheuristic.

1. Introduction

Flexible job shop scheduling problem (FJSP) is an extension of the classical Job-shop scheduling problem (JSP) (see for instance some recent papers (Wang and Duan, 2014; Peng et al., 2015; Ku and Beck, 2016)). The FJSP, which takes into account routing flexibility, can be found in many applications in the modern manufacturing system. The specificity of FJSP is that operations related to a given job should be processed by a machine selected in a set of available ones. Considering the assignment problem resulting from this selection and the common scheduling problem relevant to the JSP, the FJSP is at least as complex as the JSP, which is known to be strongly NP-hard (Garey et al., 1976). Hence, solving such a problem advocates the use of metaheuristics in order to find valuable results, even though they are not optimal in a rather short computation time.

In this paper, an improved Greedy Randomized Adaptive Search Procedure (GRASP) is presented. Introduced by Feo et al. (1994), GRASP is a multi-start local search metaheuristic where each initial solution is constructed using a greedy randomised heuristic. The specificity of the proposed metaheuristic is to hybridise the GRASP with a multi-level Evolutionary Local Search (mELS) metaheuristic which extends the ELS procedure of Wolf and Merz (2007). Therefore, in this paper, a metaheuristic algorithm called GRASP-mELS is proposed. This metaheuristic has very effective searching ability and can balance the intensification and diversification very well by using four different neighbourhood structures during the search process. Experiments show that the proposed algorithm is able to return good solutions on a large number of instances found in the literature. The metaheuristic approach proposed in this paper differs from the FJSP literature according to the following points:

- An efficient construction heuristic, which generates good starting solutions.
- A local search based on critical path analysis, which encompasses both machine changes and operation permutations, is designed. This local search makes use of a procedure for fast estimation of the neighbours' quality.
- Definition of new neighbourhoods integrated in a highly randomised neighbourhood structure, where several permutations or machine changes are made to generate a neighbour of a solution.
- An improved intensification scheme, the multi-level ELS, embedded in a GRASP metaheuristic.

The paper is organised as follows: in the second section, the

☆ This article is an extended and improved version of a preliminary GRASP based metaheuristic provided in Kemmoé et al. (2016) which did not include procedure for fast estimation of the neighbourhood quality and neighbourhood structure formalisation. Experiments are updated and completed with Hurink benchmark.

* Corresponding author.
E-mail addresses: Sylverin.KEMMOE_TCHOMTE@u-clermont1.fr (S. Kemmoé-Tchomté), lamy@isima.fr (D. Lamy), tchernev@isima.fr (N. Tchernev).

http://dx.doi.org/10.1016/j.engappai.2017.04.002
Received 9 July 2016; Received in revised form 28 January 2017; Accepted 2 April 2017

Nomenclature

n	Number of jobs to schedule
m	Number of machines
J	Set of jobs to schedule, $J=\{J_1, ..., J_r\}$
M	The set of all machines, $M=\{M_1, ..., M_m\}$
i,j,k,l	Indices for jobs
a,b,c,d	Indices for job operations
u,v,v',w,w'	Indices for machines
$M_{i,a}$	The set of available machines for operation $O_{i,a}$
n_i	Number of operations of job J_i
$O_{i,a}$	a^{th} operation of job J_i
$p_{O_{i,a},v}$	Processing time of operation $O_{i,a}$ on machine v
$s_{O_{i,a}}$	Starting date of operation $O_{i,a}$
$c_{O_{i,a}}$	Ending date of operation $O_{i,a}$
C_{max}	Total completion time of all operations
$C_{max}{}^e$	Estimated makespan
α	Assignment of all operations
π	Schedule of all operations
ρ	Critical path of $G(\alpha,\pi)$
$S(\alpha,\pi)$	Solution associated to α and π
$G(\alpha,\pi)$	Graph associated to a solution
$E(\alpha,\pi)$	Set of disjunctive arcs representing the processing orders on machines
Nn^{π}, $N_1^{\pi-\rho}$	Neighbourhood concerning schedules. n represents the number of times the neighbourhood is applied. ρ means the neighbourhood is applied on the critical path.
$Nn^{\alpha}N_1^{\alpha-\rho}$	Neighbourhood concerning assignments
B, B', B''	Critical blocks (consecutive operations in a critical path, assigned to the same machine)
$SM_{O_{i,a}}$	Operation succeeding $O_{i,a}$ in the assigned machine
$PM_{O_{i,a}}$	Operation preceding $O_{i,a}$ in the assigned machine

formulation of the Flexible Job-shop problem is introduced. Some related works, which motivated this study, are given in the third section. The fourth section refers to the metaheuristic and to the integrated procedures such as neighbourhood structures and the local search. Experiment results on four well known sets of problems taken from the literature are then presented and compared to the state of the art approaches. Finally, a conclusion in the sixth section of this paper summarises key points and future research.

2. Problem definition

2.1. Problem settings

The FJSP problem has been first introduced by Brucker and Schlie (1990). It is formally formulated as follows: a set J of n jobs, $J=\{J_1, ..., J_n\}$ must be scheduled on a set M of m machines, $M=\{M_1, ..., M_m\}$. Each job J_i consists in a number of n_i operations $O_{i,l}, ..., O_{i,\ n_i}$, which have to be processed in this order. An operation $O_{i,a}$ is allowed to be executed in any machine of a given set $M_{i,a} \subseteq M$. The processing time of an operation $O_{i,a}$ on a machine $M_v \in M_{i,a}$ is noted $p_{O_{i,a},v}$. Furthermore, each operation can be assigned to only one machine. Note that several operations of a same job could be assigned to the same machine. Considering that there is an assignment and a schedule for all the operations, the starting date of an operation is noted $s_{O_{i,a}}$ and its ending date is noted $c_{O_{i,a}}$. The objective is to minimise the total completion time, also called makespan and noted C_{max}, of all the operations. This criterion is given as follows: $C_{max} = \max_{1\le i\le r, 1\le a\le n_i}(c_{O_{i,a}})$. Several assumptions are made for the classical problem, involving availability of all machines and release dates of all jobs at time 0.

2.2. Graph modelling

The classical representation of job scheduling problems is usually based on the disjunctive graph model presented by Roy and Sussmann (1964). It is possible to extend this representation to FJSP as shown by Dauzère-Pérès and Paulli (1997). Using the disjunctive graph model, any FJSP instance can be represented by a disjunctive graph $G=(V,A,E)$, where V represents the set of nodes, A the set of conjunctive (oriented) arcs and E the set of pairs of disjunctive (non-oriented) arcs. The nodes are associated with operations $O_{i,a}$. There are also two particular nodes to represent the start and the end of the schedule: (*i*) a source node, denoted 0 connected to the first operation of each job and (*ii*) a sink node, denoted *, linked with the last operation of each job. The conjunctive arcs are used to represent the precedence constraints of the different operations of the jobs and connect each pair of consecutive operations of the same job. Each pair of disjunctive arcs connects two operations, belonging to different jobs, which may be processed on the same machine. A solution corresponds to an acyclic subgraph that encompasses all conjunctive arcs and that contains at most one disjunctive arc for each pair of disjunctive operations. An optimal solution corresponds to the feasible subgraph with the minimal makespan C_{max}.

A solution can be noted $S(\alpha,\pi)$. In $S(\alpha,\pi)$, α represents a feasible assignment of each operation $O_{i,a}$ to a machine $M_v \in M(O_{i,a})$. ($v = \alpha(O_{i,a})$). A feasible assignment is a vector where each operation is assigned to one and only one machine. In $S(\alpha,\pi)$, π is a schedule of the operations, respecting job sequences, on all the machines in M. A solution Graph could be noted $G(\alpha,\pi)=(V,A,E(\alpha,\pi))$ with $E(\alpha,\pi)$ being the set of disjunctive arcs representing the processing orders on machines. The makespan of $S(\alpha,\pi)$ is the cost of any critical path in $G(\alpha,\pi)$. In the following, the notation B indicates a machine-block which is a set of consecutive operations needing the same machine in a critical path of $G(\alpha,\pi)$, following the definition of (Grabowski et al., 1986). Fig. 1 represents a solution for a problem with 4 jobs and 5 machines. Operations $O_{2,2}$, $O_{3,2}$ and $O_{4,2}$ form a machine-block since they are proceeded by the same machine M_5. Note that $O_{4,3}$ is also processed on machine M_5 but does not belong to the machine-block because of precedence constraints and therefore it is impossible to permute order of operations $O_{4,2}$ and $O_{4,3}$.

3. Related works

Two main approaches have been used in the research done so far concerning FJSP: *(i)* hierarchical solving, which consists in finding the best schedules while considering that assignments of machines are fixed (Brandimarte, 1993); or *(ii)* considering simultaneously the scheduling and assignment problems by using effective neighbourhoods (Mastrolilli and Gambardella, 2000). Actually, most of works are

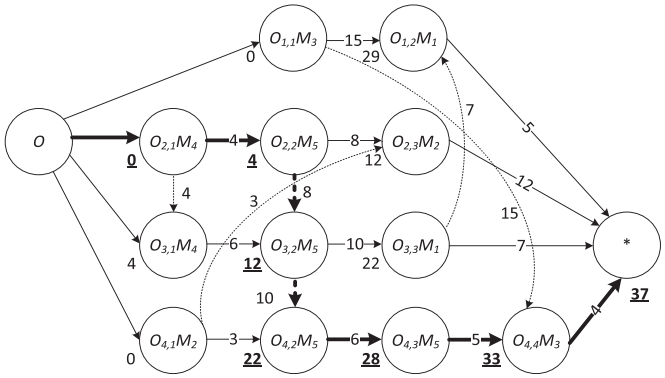

Fig. 1. A schedule of a problem with 4 jobs and 5 machines represented by a solution graph. Bold arcs show a critical path whose length, i.e.: the makespan, is 37.