Available online at www.sciencedirect.com
SCIENCE DIRECT
Engineering Applications of Artificial Intelligence 19 (2006) 135–143
Engineering Applications of ARTIFICIAL INTELLIGENCE
www.elsevier.com/locate/engappai

Hybrid solution algorithms for task scheduling problem with moving executors

Jerzy Józefczyk*

Systems Research Institute of Polish Academy of Sciences, Laboratory of Knowledge Systems and Artificial Intelligence, Podwale St. 75, 50–449 Wroclaw, Poland

Received 1 December 2004; received in revised form 27 May 2005; accepted 1 August 2005
Available online 12 October 2005

Abstract

Heuristic algorithms for solving the task scheduling problem with moving executors to minimize the sum of completion times are considered. The corresponding combinatorial optimization problem is formulated. Three hybrid solution algorithms are introduced. As a basis an evolutionary algorithm is assumed that is combined with the procedure that uses simulated annealing metaheuristics. The results of simulation experiments are given in which the influence of parameters of the solution algorithms as well as of the number of tasks on the quality of scheduling and on the time of computation is investigated.

Keywords: Scheduling algorithms; Artificial intelligence; Flexible manufacturing systems

1. Introduction

Problems of task scheduling with moving executors are a generalization of traditional task scheduling problems. It is assumed that executors of tasks move between places at a plane or in a space, where the tasks can be performed. These places called workstations are supplemented by a depot being a place, where executors begin and end their work. Modern discrete manufacturing systems are the main area of applications of the complex scheduling problem in the cases when the movement of plants being in production is impossible or too expensive and it is better to move executors. For example, it occurs when plants in production (e.g. big engine casings, railway carriage elements) are too big or too heavy to move among executors (machines). Another example may concern maintenance tasks on stationary machines or transport tasks performed by mobile robots or automated guided vehicles in flexible manufacturing systems. Automated service systems as well as computer systems, where the times of the movement of executors (agents) are not neglected, can be the prospective examples for the applications of the scheduling problem under consideration.

Till now different cases of the task scheduling with moving executors concerning different performance indices have been investigated. The makespan, the maximum lateness and the sum of completion times were considered, e.g. Józefczyk (1997, 2001a, 2003b). The similar approach referred to as a routing-scheduling with application to flow-shop problems and with travelling machines is presented in Averbakh and Berman (1999). The problem of task scheduling with moving executors is also connected with the travelling salesman problem with time windows (TSPTW) which has been stated in Christofides et al. (1981). Further works develop it. For example in (Langevin et al., 1993) the makespan problem with time windows (MPTW) is considered (see Mingozzi et al., 1997 and Desrosiers et al., 1992 for a survey), the additional resources are taken into account which leads to TSP-TWPC. New solution methods are presented e.g. in Dumas et al. (1995). In all these works, all tasks are performed only by one executor. The problem considered

*Tel.: +48 71 3418787; fax: +48 71 3418888x335.
E-mail address: Jerzy.Jozefczyk@pwr.wroc.pl.

0952-1976/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engappai.2005.08.001

can be also treated as a generalization of the scheduling of tasks with setup times, e.g. Potts and Kovalyov (2000); Rajendran and Ziegler (2003).

Each task scheduling problem with moving executors is a generalization of the corresponding version without movement of executors which can be treated as its special case, e.g. Józefczyk (2003b). Consequently, in most cases NP–hard optimization problem should be solved. Therefore, apart from exact and approximate solution algorithms the determination of more and more effective heuristic solution algorithms is crucial. The artificial intelligence approaches and methods are a good basis for the determination of the heuristic algorithms.

In the paper two artificial intelligence-based approaches are used to solve the selected problem under consideration which consists in scheduling of independent, non-preemptive tasks on one executor to minimize the sum of completion times. The performance index as the criterion of the average flow time type is important for customers of the scheduling system because it is correlated with the mean execution time of the task, e.g. Blazewicz et al. (2001). The form of the performance index makes a distinction between the scheduling problems considered in the paper and different travelling salesman as well as vehicle routing problems, where the total cost is minimized, although the result in the form of routes of executors is the same. The case with one executor is only considered because the solution algorithms proposed are not straightforwardly extendable for the problems with many executors. On the basis of previous considerations presented e.g. in Józefczyk (2002, 2003b), where the evolutionary approach as well as the simulated annealing (SA) metaheuristics were separately investigated, hybrid solution algorithms have been proposed and are presented in the paper. The algorithms are a combination of both approaches. Three different versions are considered. The formulation in Section 3 of the task scheduling problem under consideration follows the more general description of the task scheduling with moving executors. In Section 4 the hybrid solution algorithms are introduced as a whole as well as their parts, i.e. the evolutionary and the SA algorithms are described. The important contribution of the paper is the evaluation of the hybrid solution algorithms proposed via computer simulation experiments. The selected parameters of the algorithms have been tuned in the experimental way. Then the influence of the number of tasks on the quality of the scheduling and on the time of computation has been investigated and the examples of the results are presented.

2. Task scheduling with moving executors

Let us shortly introduce the problem of task scheduling with moving executors. We assume that at every workstation only one task is performed on a plant being produced which is located there. There is no distinction in notation between a set of tasks and a set of workstations. Both are denoted by $\boldsymbol{H} = \{1, 2, \ldots, H\}$, where H is a number of tasks (workstations). The element of $\boldsymbol{H}$, i.e. $h \in \boldsymbol{H}$ is an index of the current task (the current workstation). Analogously, $\boldsymbol{R}$, R, $r \in \boldsymbol{R}$ are a set of executors, a number of executors and an index of the current executor. To perform the tasks on plants, executors should move among workstations and a depot being the additional workstation, where the executors begin and end their movement. Therefore, $\bar{\boldsymbol{H}} = \boldsymbol{H} \cup \{H+1\}$ is the set of workstations with the depot, where $h = H+1$ denotes the depot. Each task consists of two parts: the performance of a job at the workstation and the driving-up of the executor towards the workstation from the other workstation belonging to the set $\bar{\boldsymbol{H}}$. In the consequence, for each executor a route should be determined in the form of Hamilton cycle with the beginning and the end at the depot. The routes are the admissible solution of the problem of task scheduling with moving executors. The example of such a solution is given in Fig. 1 for $H = 15$ and $R = 2$. The assumption of the movement of executors and the twofold form of the task have the influence on the data of the scheduling problem, in particular on the execution times τ_h, $h = 1, 2, \ldots, H$ which are the main data for every task scheduling problem. For unrelated executors the time $\boldsymbol{\tau}_h$ is the vector:

$$\boldsymbol{\tau}_h = [\tau_{1,h}, \tau_{2,h}, \ldots, \tau_{r,h}, \ldots, \tau_{R,h}]^{\mathrm{T}},\quad h = 1, 2, \ldots, H, \tag{1}$$

where $\tau_{r,h}$ is the execution time of the task h performed by the executor r. Using this notation employed in the scheduling theory, for the case under consideration, i.e. when executors are mobile, the time $\tau_{r,h}$ is the sum of two elements: $\bar{\tau}_{r,h}$ and $\tilde{\tau}_{r,g,h}$ being the time the job h is performed at the workstation h by the executor r and the driving-up time of the executor r to the workstation h from the workstation g, respectively. Thus,

$$\tau_{r,h} = \bar{\tau}_{r,h} + \tilde{\tau}_{r,g,h},\quad r = 1, 2, \ldots, R,\quad h = 1, 2, \ldots, H,\quad g = 1, 2, \ldots, H+1. \tag{2}$$

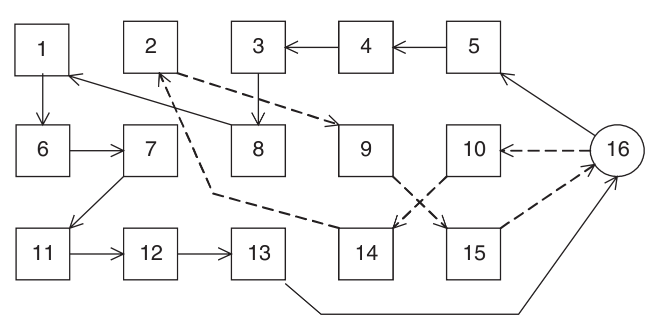

Fig. 1. Example of the layout of workstations and the routes of executors.