Engineering Applications of Artificial Intelligence 25 (2012) 242–253

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Guided restarting local search for production planning

Gregor Papa *, Vida Vukašinović, Peter Korošec

Computer Systems Department, Jožef Stefan Institute, Ljubljana, Slovenia

ARTICLE INFO

Available online 16 July 2011

Keywords:
Local search
Production
Planning
Genetic algorithm
Parameter-less

ABSTRACT

Planning problems can be solved with a large variety of different approaches, and a significant amount of work has been devoted to the automation of planning processes using different kinds of methods. This paper focuses on the use of specific local search algorithms for real-world production planning based on experiments with real-world data, and presents an adapted local search guided by evolutionary metaheuristics. To make algorithms efficient, many specifics need to be considered and included in the problem solving. We demonstrate that the use of specialized local searches can significantly improve the convergence and efficiency of the algorithm. The paper also includes an experimental study of the efficiency of two memetic algorithms, and presents a real-world software implementation for the production planning.

1. Introduction

Planning problems exist in many real-world-industrial production settings and can be solved with a large variety of different approaches. Due to the growing complexity of these real-world planning problems, significant work has been devoted to the automation of planning processes using different kinds of methods. Furthermore, as a result of the increasing deployment of planning and scheduling systems, we often have to deal with very large search spaces, real-time performance demands, and dynamic environments (Nareyek, 2000). A review of the research on integrated process planning and scheduling, along with a discussion on the extent of the applicability of various approaches is presented in Li et al. (2010) and Shaotan et al. (2010). This paper focuses on the use of specific local search algorithms for real-world production planning based on experiments with real-world data and presents the adapted, local search, evolutionary-based metaheuristics for the real-world industrial problem. It also includes an experimental study of the efficiency of two memetic algorithms, and presents a real-world software implementation for the production planning.

The motivation of this research is to solve an industrial problem. The goal of our real-world production-planning problem is to find a production plan that satisfies the production time constraints and minimizes the production costs. This involves many specific constraints that need to be considered. The main problem is the exchange delay, caused by adapting the production lines to different types of products and supplying the appropriate parts. Namely, the manufacturing processes of multiple types of products require many different steps and different product parts for the completion of each product type. Such a kind of problem can be represented as a job shop scheduling problem (Brucker, 1998).

For solving scheduling problems, many scheduling methods are reported in the literature (Xing et al., 2010; Wang and Tang, 2009; Guo et al., 2009; Chiong and Dhakal, 2009; Jarboui et al., 2009). One of the nature-inspired approaches (Chiong, 2009) is the genetic algorithm (GA). A heuristic for the open job shop scheduling problem using the GA to minimize makespan was developed by Senthilkumar and Shahabudeen (2006). On the other hand, a scheduling method based on the GA was developed by considering multiple criteria in Chryssolouris and Subramaniam (2001). Other implementations of the GA for scheduling can be found in Vazquez and Whitley (2000).

The advanced local search approaches are usually guided by evolutionary algorithms, simulated annealing, tabu search, min-conflicts, and others (Voß, 2001). The basics of local search are in the successive changing of solutions with moves that alter solutions locally. These moves are defined by the neighborhood, which contains all the solutions that can be reached with one move/change. As the solution quality of the local optima may be unsatisfactory, we need some mechanisms that guide the search to overcome the local optimality. A simple strategy is to iterate/restart the local search process after a local optimum has been obtained, while more structured ways, e.g., the use of an evolutionary search, to overcome the local optimality might be advantageous. In Hamiez and Hao (2001), the problem of sports-league scheduling is addressed, presenting the results achieved by a tabu search method based on a neighborhood of value swaps. Furthermore, some works deal with planning systems that are able to

* Corresponding author.
E-mail addresses: gregor.papa@ijs.si (G. Papa), vida.vukasinovic@ijs.si (V. Vukašinović), peter.korosec@ijs.si (P. Korošec).

0952-1976/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engappai.2011.07.001

incorporate resource reasoning. In Chien et al. (2001) the replanning capabilities of local search methods in a continuous planning process clearly outperform a restart strategy. In Engelhardt and Chien (2001), learning is used to speed up the search for a plan, where the goal is to find a set of search heuristics that guide the search, while Knight et al. (2001) proposes a technique for aggregating single search moves, so that distant states can be reached more easily.

GAs in combination with local search procedures are known as memetic algorithms (MAs) (Ong and Keane, 2004). They represent a synergy of the evolutionary approach with separate individual learning or local improvement procedures (local searches) for the problem search. Various MAs were developed (Hasan et al., 2009; Caumond et al., 2008; García-Martínez and Lozano, 2008; Siarry and Michalewicz, 2008; Ombuki and Ventresca, 2004) to obtain even better results than the GA for various scheduling applications. With the use of local search techniques the results were further improved. MAs do not only improve the quality of the solutions, but they also reduce the overall computational time (Hasan et al., 2009). Special attention should also be given to the dynamic nature of the production. Local search is well suited for anytime requirements because the optimization goal is improved iteratively (Nareyek, 2000). Here, a variety of algorithms have been proposed to solve dynamic optimization problems (Moser and Chiong, 2010; Tfaili and Siarry, 2008; Engelhardt and Chien, 2001).

The rest of the paper is organized as follows: in Section 2, we describe the problem and give its mathematical formulation; in Section 3, we present preliminary evaluations for the algorithm selection; in Section 4, we describe the evolutionary algorithms; in Section 5, we introduce the used local search procedures; in Section 6, we show the integration of guiding evolutionary algorithms with local search procedures; in Section 7, we present the experimental environment and the results of the evaluation with the real-world data; in Section 8, we present the implemented application for automatic production planning; and in Section 9, we conclude and propose possible future work.

2. Production planning problem

The planning problem was introduced while designing the application for planning and monitoring of the production in the company Eta Cerkno d.o.o. The company produces components for domestic appliances, including hot plates, thermostats, and heating elements.

Among the various production stages, the most demanding was the production of cooking hot plates. Namely, the production of the components and parts for all the types of plates is more or less similar, but the standard plate models of the current range differ in size (height, diameter), connector type and power characteristics (wattage). There are so many different models due to the various demands of other companies that need those plates for their own cooking appliances. Orders for some particular models vary in quantities and deadlines. Many orders from the same company for different models are also connected with the same deadline. Therefore, their production must be planned very carefully to fulfill all the demands (quantities and deadlines), to maintain the specified amount of different models in the stock, to optimally occupy their workers, and to efficiently use all the production lines. Furthermore, not all the production lines are equal, since each of them can produce only a few different models.

2.1. General formulation of the problem

Our production planning problem can be represented as a job shop scheduling problem that is NP-hard (Brucker, 1998). A schedule is an allocation of one or more time intervals for each job to one or more machines. Let us assume we have a finite set of n jobs, where each job consists of a chain of operations. The jobs in our production planning problem correspond to the orders of products. Each order may consist of different type of products. Every such order is split into more orders that consist of only one type of product. Production process of each product consists of a set of operations. Then, we have a finite set of m machines, where each machine can handle at most one operation at a time. Machines correspond to production lines. Each operation needs to be processed during an uninterrupted period of a given length on a given production line. The objective is to find a schedule satisfying certain restrictions, while minimizing the overall execution time, i.e., the time for the execution of all the operations. In our planning problem each production line has its own time schedule. Furthermore, each order has its own deadline, which should not be missed, but can be executed anytime before the deadline. Each product can only be made on some production lines and on each of them the execution time is different. Changing the manufacturing process from one product to another may cause an exchange delay, which also depends on the used production line. There is also a stock. If the product we want to produce is in stock then we use the stock and produce the missing quantity of product. There might be orders with fixed deadlines, which need to be produced on the exact day, but not before. Nevertheless, the results in this paper do not consider such orders.

2.2. Mathematical formulation of the problem

Our production planning problem is defined by a set of orders $J = \{j_1, j_2, \ldots, j_n\}$ and a set of production lines $M = \{M_1, M_2, \ldots, M_m\}$, where the orders have to be processed. Let F_i denote a finishing time of order j_i and let D_i denote its deadline for every $i \in \{1,2,\ldots,n\}$. Each order j_i consists of a set of n_i operations $O_i = \{o_{i1}, o_{i2}, \ldots, o_{in_i}\}$. For the process of every $o_{ik} \in O_i$, $k \in \{1,2,\ldots,n_i\}$, only some production lines $\mu_{ik} \subseteq M$ are appropriate and the processing time τ_{ik} of o_{ik} depends on the production line used. By $S[p]$ we denote the number of products p available in the stock. In this paper we consider the problem, where all operations of a product are done on the same production line; its processing time equals the processing time of the product p on the production line used times the number of p products. In this case the chain of operations is formally unite into one operation. If some products are already in stock, the order processing time is appropriately shorter.

Let us assume that N is the number of operations:

$$N = \sum_{k=1}^{n} |O_k|.$$

A schedule is denoted by

$$C = g_{11} g_{12} g_{21} g_{22} \cdots g_{N1} g_{N2}, \tag{1}$$

where g_{k1} is an index on some operation $o_k \in O_1 \cup \cdots \cup O_n$ and g_{k2} is the production line used to perform the operation o_k, for every $k \in \{1,2,\ldots,N\}$. In the following, we use the notation of index g_{k1} and operation o_k interchangeably. Furthermore, let $F(o_k)$ denote a finishing time of the operation o_k and let $exd(l, o_j, o_k)$ denote the exchange delay caused by the exchanging of operation o_j for the operation o_k on the production line M_l, for $o_j, o_k \in O_1 \cup \cdots \cup O_n$. The task is to find the schedule that minimizes the number of delayed orders, exchange delays and the time to finish all the orders. The number of delayed orders n_{orders} is

$$n_{\text{orders}} = \sum_{i=1}^{n} \text{delay}_i,$$