Expert Systems with Applications 38 (2011) 10770–10779

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Modified genetic algorithms for manufacturing process planning in multiple parts manufacturing lines

F. Musharavati, A.S.M. Hamouda *

Department of Mechanical and Industrial Engineering, Qatar University, P.O. Box 2713, Doha, Qatar

ARTICLE INFO

Keywords:
Manufacturing process planning (MPP)
Modified genetic algorithm (MGA)
Simple genetic algorithm (SGA)
Simulated annealing (SA)
Multiple parts manufacturing lines (MPMLs)

ABSTRACT

Manufacturing process planning for multiple parts manufacturing is cast as a hard optimization problem for which a modified genetic algorithm is proposed in this paper. A cyclic crossover operation for an integer-based representation is implemented to ensure that recombination will not result in any violation of processing constraints. Unlike classical approaches, in which the mutation operator alone is used to foil the tendency towards premature convergence, a combination of a neighborhood search based mutation operator and a threshold operator were implemented. This combined approach was designed to; (a) improve *the exploring potential and* (b) *increase population diversity of neighborhoods*, in the genetic search process. Capabilities of a modified genetic algorithm method were tested through an application example of a multiple parts reconfigurable manufacturing line. Simulation results show that the proposed modified genetic algorithm method is more effective in generating manufacturing process plans when compared to; a simple genetic algorithm, and simulated annealing. A computational analysis indicates that improved, near optimal manufacturing process planning solutions for multiple parts manufacturing lines can be obtained by using a modified genetic algorithm method.

1. Introduction

One of the most important functions in manufacturing engineering is process planning. Manufacturing process planning has a wide engineering background. As pointed out in Scallan (2003), the actual activities in manufacturing process planning depends on the type of manufacturing system as well as the nature of the products of manufacture. Most of the process planning methods, tools and techniques discussed in manufacturing engineering literature are more related to mature manufacturing system technologies such as; job shop systems (Zhang & Nee, 2001), cellular manufacturing systems (Morad & Zalzala, 1999) and flexible manufacturing systems (Saygin & Kilic, 1999). Little, however, is available to guide process planners for recent, new and innovative manufacturing styles. An example of such deficiencies is found in reconfigurable manufacturing systems (RMSs) that manufacture multiple parts in a given planning horizon. There is, therefore, a need to review conventional process planning methods, tools and techniques for application in new manufacturing styles.

For multiple parts production, interface issues such as parts flow intensities, machine load limits, part loading constraints as well as manufacturing capability limits cannot be ignored during the process planning phase. This requirement makes it difficult to implement process plans of a linear nature. Moreover, the dynamics in the operational and structural complexity in multiple parts manufacturing lines (MPMLs) that allow reconfigurable flow of parts will render linear process plans infeasible with time. To circumvent this challenge, flexible process plans that can accommodate reconfigurable flow of multiple parts should be implemented. To generate such flexible process plans, a multidimensional approach that captures some of the issues mentioned above is described in this paper.

In light of the discussion above, the manufacturing process planning problem for reconfigurable MPMLs can be described as follows. Given a set of flexible production machines for the manufacture of a multiple parts production scenario with multiple parts of total number, *np*, belonging to a number of part families, *npf*, the total number of feasible manufacturing process plans can be estimated mathematically as follows:

Let each part, *p*, require *Y* activities for manufacture, where *Y* is multi-dimensional. Let the *i*th sub-activity have $y(i)$ alternatives, and let each part require x_i elements in dimension *i* of *D* dimensions, where $D \in Y$. Then the total number of feasible plans, *z*, can be approximated by the following expression

$$z = \prod_{i=2}^{D} \left[\frac{x!}{(x_i - x_1)!} \right] \sum_{i=1}^{npf} \sum_{i=1}^{np} y(i) \quad (1)$$

* Corresponding author.
E-mail address: hamouda@qu.edu.qa (A.S.M. Hamouda).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.01.129

Although the expression in Eq. (1) is approximate, it serves to demonstrate that it is difficult to find an optimal solution by enumerating all the feasible solutions. Consequently, this manufacturing process planning problem is a hard optimization problem. Since manufacturing process planning is crucial in the operations of a reconfigurable MPML, it is important to develop effective optimization methods that are capable of recommending high-quality process planning solutions with a reasonable computational effort.

Over the past years, hard optimization problems have been solved through algorithm design techniques that are robust in nature and non-math-knowledge dependent (Hromkovič, 2004). This includes heuristics such as simulated annealing (SA) and genetic algorithms (GAs). Genetic algorithms have found wide applications in many engineering fields including manufacturing process planning (Zhang & Nee, 2001; Zhang et al., 1997). However, it is difficult to ensure the convergence of GAs since there is a possibility of premature convergence in the genetic search process. Moreover, it is time consuming to parameterize a genetic algorithm implementation. For successful practical applications, it is almost always necessary to devise techniques that enhance a genetic algorithm application in a specific problem domain.

It has been observed that more recent successful applications of GAs in complex solution spaces have tended to focus more on modified genetic algorithms in a bid to implement GAs that are more competent than the simple genetic algorithm (Chen, 2006; Li & Wang, 2007; Wang, Cai, Guo, & Zhou, 2007). This work explores the effectiveness of a modified genetic algorithm (MGA) method in obtaining optimal manufacturing process planning solutions for reconfigurable MPMLs.

In the practical implementations of genetic algorithms, it has been observed that the effectiveness of an implemented genetic algorithm in solving a given problem depends on the chosen fitness function and the appropriateness of the genetic operators used (Chen, 2006). Therefore, in modifying the genetic algorithms, the emphasis in this work was to model the problem rigorously by adopting a decision-making perspective that focuses on generic issues such as capturing relevant production information, evaluating alternatives and selecting the best alternative. Unlike conventional approaches, this approach helps to reduce alternative choices in the decision making process towards optimization.

The steps in developing the modified genetic algorithm method used in this work were as follows. Firstly, a comprehensive fitness function that captures and models a rigorous evaluation criterion was devised. Secondly, a cyclic crossover operator for an integer-based representation was implemented to ensure that recombination will not result in any violation of processing constraints. Thirdly, heuristics that support the progression of the genetic algorithm towards convergence were devised. Fourthly, a combination of a neighborhood search based mutation operator and a customized threshold operator were implemented to foil the tendency towards premature convergence in the genetic search process.

In exploring the effectiveness of the proposed modified genetic algorithm, an application example of a multiple parts reconfigurable manufacturing line was used. The appropriateness of the MGA in generating manufacturing process plans for a MPML was compared with that of a simple genetic algorithm (SGA) and a simulated annealing (SA) algorithm.

The reminder of the paper is organized as follows: In Section 2, the proposed modified genetic algorithm method is discussed. In Section 3, the representation of the solution and optimization operators for the MPMLs is presented. In Section 4, the proposed MGA is applied to an instance of a manufacturing process planning optimization problem and the corresponding computational and comparative results are presented. Finally, concluding remarks are given in Section 5.

2. Modified genetic algorithm method

Genetic algorithms are stochastic search techniques that start with a set of random solutions, i.e. a population, and each individual in the population is called a chromosome. The chromosomes evolve through successive iterations, i.e. generations, during which the chromosomes are evaluated by means of a fitness function. As the evolution process continues, the genetic search process eventually converges to an optimal solution. In developing a genetic algorithm solution, it is always necessary to select an appropriate representation scheme.

2.1. Representation scheme

Most genetic algorithms (GAs) in the literature employ a binary code for encoding various problems. However, it has been discussed that in certain situations, the binary code is not always suitable (Lian et al., 2005). In this work, an integer-based code was devised to encode the manufacturing process planning problem. In this code, an array of strings is used to represent a valid manufacturing process plan. Thus, for x operations or preferably x processing types (PTs), an x-PT string is used to represent a sequence of various processing types in the manufacturing process plan for a specific part. In this representation, each string is composed of x segments. Each segment represents a combination of processing types (PTs), selected from those available, for the complete manufacture of a part. This means that if the production scenario is composed of a total of np parts, then the process planning solution will contain np strings, each string specifying the required processes for the manufacture of a particular part in the production scenario. Therefore, in providing a solution, the aim is to find an optimal manufacturing process plan that is composed of np strings, for which each string represents the required processing types for each of the multiple parts flowing in the system.

Suppose each part is assumed to have v alternative manufacturing process plans. If the parts belong to a family of products, and the manufacturing line has a total number of stages N, composed of w processing machines (or processing modules) that can be used for processing the parts then;

$$1 < v < u$$

where u is the maximum number of alternative processing machines that can be selected for routing parts in the system. As a result, the chromosome representation of the process planning problem takes the general form shown in Fig. 1.

In Fig. 1, a_i represents the selected process routing for part i. In the proposed encoding, an integer string represents all feasible permutations of operational states, as defined by the selected processes, that make up the process plan in the manufacturing system for a given production scenario.

In the genetic algorithm analogy to the natural evolution process, a gene specifies the locus, i.e. the position of the gene location within the chromosome structure, and an allele specifies the value the gene takes. In this application, the locus represents a processing identity (ID) composed of a specific stage, N, at which processing takes place by the selected processing machine, w. The allele represents the state of a processing type (PT) chosen according to a decision value specified by decision criteria. The processing types (PTs) are used in the encoding scheme, while the processing machines (PMs), selected from a pool of possible alternatives, are used in the decoding scheme. An example representation of the decoding scheme for part i is shown in Table 1. In Table 1, N is the number of processing stages in the manufacturing system; w represents the processing machines (PMs) selected for the manufacture of a specific part in the production scenario. In the decoding scheme in Table 1, the first row represents specific production stages while