Manufacturing Cell Formation: A Dual-Objective Simulated Annealing Approach

S. Lee and H. P. Wang
Department of Industrial Engineering, The University of Iowa, Iowa City, Iowa, USA

A dual-objective simulated annealing approach is developed for the purpose of concurrently forming machine cells and part families for cellular manufacturing. Simulated annealing has proved to be very effective in handling a variety of NP-hard problems. Originally, the simulated annealing algorithm was designed for applications with a single objective. For forming machine cells based on a machine-component incidence matrix, the result from single-objective simulated annealing was found to have room for improvement. In order to improve the performance of simulated annealing, a dual-objective approach is developed. The result clearly demonstrates the outstanding capabilities of the dual-objective approach. The discussions are illustrated with examples.

Keywords: Cellular manufacturing; Cell formation; Fuzzy set theory; Simulated annealing

1. Introduction

It is said that cellular manufacturing is the second generation of group technology [1]. Cell manufacturing has demonstrated its effectiveness in dealing with low-quantity, high-variety production. In a cellular manufacturing system, machined parts of similar features and/or requiring similar production processes are grouped into families. The machines required to produce families of parts are then placed close together to form cells. Such arrangements streamline the production within each cell. As such, the production within the cell is just like that of mass production. Various benefits, for instance high machine utilisation, short throughput time, less material handling distance, low work in process (WIP), and so on, have been reported. Having both productivity and flexibility, cellular manufacturing has become one of the important strategies for flexible manufacturing.

Stecke classified those important activities in flexible manufacturing systems (FMS) production planning stages into five areas: 1. machine grouping, 2. loading, 3. part type selection, 4. production ratio, and 5. resource allocation [2]. Among them, machine grouping is considered to be the basis of the others. The results from such an activity are used in either the physical or logical layout of machines. Based upon which machines are in the machine cells, part families can be formed, and subsequent activities can be carried out. Therefore, machine grouping not only determines the layout of the machines but also decides how smoothly the production activities will be. Therefore, much effort has been dedicated to developing methods for grouping machines into cells and parts into families. As a result, a number of algorithms and methods have been reported in the literature.

These grouping algorithms and methods can be classified into two basic categories, namely coding and cluster analysis. For the purpose of forming part families and machine cells, cluster analysis methods appear to be better than coding methods and have become the choice of most applications. Cluster analysis methods can be further divided into three subgroups according to the problem formulation. They are: 1. matrix-based algorithms [1, 3-6], 2. mathematical programming-based algorithms [7, 8], and 3. graph-based algorithms [9]. Among them, matrix-based algorithms are most popular. For those matrix-based algorithms, no matter how the algorithms work, they are all based on the same class of machine-component matrices: binary matrices. In a binary matrix, every matrix element can only take a value of either “0” or “1”. A positive element means that a component will visit the corresponding machine on the matrix. The elements of a binary matrix are basically obtained from the routing information of parts. They are applicable only after the routing information is known and fixed. For those cases where the routing information is not
available or alternative routing must be taken into consideration, non-binary machine-component matrices are preferred. The elements in a non-binary matrix can take a value between “0” and “1”. The value of any matrix element reflects the “degree of appropriateness” that a component will require the service of the corresponding machine. Generally, the elements of these non-binary matrices are calculated based upon fuzzy set theory [1, 10].

Most matrix-based clustering algorithms lack an explicitly defined objective function in the form of a mathematical formula. On the other hand, those mathematical programming approaches with an explicitly defined objective function usually have difficulty formulating and solving the models using currently available software and hardware when the problem grows beyond a certain size. Besides, the traditional approaches provide only one solution to each problem, resulting in a lack of alternatives which are usually desired in real world applications.

In order to overcome the shortcomings mentioned in the previous paragraphs, this paper presents a new approach which uses a simulated annealing algorithm and explicitly defined objective functions to work on non-binary machine-component matrices. The initial results revealed the need for a more effective simulated annealing approach. The original simulated annealing algorithm was developed to handle problems that have only one objective: minimising or maximising one function or measure. In manipulating the machine-component incidence matrix, at least two objectives are required; one to cluster non-zero cell entries, the other to arrange the clusters along the diagonal line of the matrix.

This paper is organised as follows. Section 2 briefly introduces the basic simulated annealing algorithm, followed by the discussion of fuzzy sets in Section 3. Two single-objective simulated annealing algorithms for cell formation are introduced with examples in Section 4. Section 5 presents a dual-objective simulated annealing algorithm with examples. Section 6 concludes the discussion.

2. Simulated Annealing

Annealing is the physical process of heating up a solid metal to a temperature which is slightly higher than its recrystallisation temperature, followed by cooling it down until it crystallises into a state with a perfect lattice. During this process, the free energy of the solid metal is minimised. Practice shows that the cooling must be done carefully in order not to get trapped in locally optimal lattice structures with an imperfect crystal.

In combinatorial optimisation, one can define a similar process which can be formulated as the problem of finding, among a potentially very large number of solutions, a solution with a minimal objective function value. By establishing the correspondence between the objective function and the free energy, and between the solutions and the physical states, an optimisation procedure, named “simulated annealing”, in the field of combinatorial optimisation was first introduced by Kirkpatrick et al. in 1983 [11]. As the name implies, it is a simulation of the physical annealing process. Fig. 1 shows the procedure of this algorithm.

The simulated annealing algorithm is very similar to an iterative improvement algorithm but differs in that the latter accepts only improvements resulting in being easily trapped in a local optimum and depends heavily on its initial guess at the solution. As illustrated in Fig. 2, the procedure will stop when a local optimum, $L$, is found. On the other hand, the former allows perturbations to move uphill, i.e. to a worse solution, in a controlled fashion. By allowing such uphill moves, it is possible to jump out of a local minimum and potentially fall into a more promising downhill path. However, because the uphill moves are carefully controlled, one need not worry about randomly jumping uphill to some far worse solution without getting close to a good, final one. As shown in Fig. 3, an uphill move will be accepted with a probability of $e^{-\frac{\Delta}{T}}$ such that it is possible to avoid getting trapped in a local optimum. Besides, with the capability of accepting uphill moves, the annealing algorithm attempts to be independent of the initial guess at the solution. In other words, the annealing algorithm consistently gives solutions which are very close to the optimal one, if not the optimum, regardless of the initial solutions.

Because of its potential for handling many classes of combinatorial problems, especially those which have