Computers and Mathematics with Applications 57 (2009) 1748–1755

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

Solving uncapacitated multilevel lot-sizing problems using a particle swarm optimization with flexible inertial weight[☆]

Yi Han[a], Jiafu Tang[a,*], Iko Kaku[b], Lifeng Mu[a]

[a] Key Laboratory of Integrated Automation of Process Industry of Ministry of Education, Northeastern University, Shenyang, 110004, China

[b] Department of Management Science and Engineering, Akita Prefectural University, Honjo, 015-0015, Japan

ARTICLE INFO

Keywords:
Particle swarm optimization algorithm
Multilevel lot-sizing problem
Assembly structure
Uncapacitated
Genetic algorithm

ABSTRACT

The multilevel lot-sizing (MLLS) problem is a key production planning problem in materials requirements planning (MRP) system. The MLLS problem deals with determining the production lot-sizes of various items appearing in the product structure over a given finite planning horizon to minimize the production cost, the inventory carrying cost, the back ordering cost and etc. This paper proposed a particle swarm optimization (PSO) algorithm for solving the uncapacitated MLLS problem with assembly structure. All the mathematical operators in our algorithm are redefined and the inertial weight parameter can be either a negative real number or a positive one. The feasibility and effectiveness of our algorithm are investigated by comparing the experimental results with those of a genetic algorithm (GA).

1. Introduction

Material requirements planning (MRP) is an old field of study within business, but it still plays an important part in coordinating replenishment decisions for complex finished goods. The MLLS problem in MRP systems belongs to those problems that industry manufacturers daily face in organizing their overall production plans [1]. The objective of the problem is to decide the optimal production lot size and the inventory volume to minimize the production cost, the inventory carrying cost, the back ordering cost and etc [2]. The MLLS problem is a combinatorial optimization problem which can be classified into different categories according to the product structures (e.g., single level system, serial, assembly, and general systems) and the capacity structures (e.g., uncapacitated, capacitated single resource, and capacitated multiple resources) [3]. Table 1, which is based on Ref. [3], gives a brief review of some important literature for the different categories of the capacitated lot-sizing problem.

In both MRP and manufacturing resource planning (MRPII), capacity is an important factor that is always checked by a capacity requirements planning (CRP) module, but it does not mean that the uncapacitated problem is an out-of-date problem. One can justify this by the fact that, in practice, uncapacitated lot-sizing models continue to be largely used since the implementation of capacitated approaches requires much data which firms are often reluctant to collect or maintain [4]. So the uncapacitated problem still has significance.

For solving the MLLS problem, people used to adopt heuristics (e.g. Wagner–Whitin, Silver–Meal and etc.) [2]. Recently, the applications of evolutionary computing methods (ECM) were seen in some papers. Tang [14] adopted simulated annealing to solve uncapacitated serial structure problems; Dellaert an Jeunet [1,24] used genetic algorithms for solving an uncapacitated general structure problem; Xie and Dong [3] proposed a heuristic genetic algorithm for a capacitated general

☆ The project was supported by the National Natural Science Foundation of China (Grant Nos. 70625001, 70721001).
* Corresponding author.
E-mail addresses: arctic_wind@yahoo.cn (Y. Han), jftang@mail.neu.edu.cn (J. Tang), ikou_kaku@akita-pu.ac.jp (I. Kaku).

0898-1221/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2008.10.024

Table 1
Different lot-sizing problems and some important literature.

Production structure	Uncapacitated	Single resource	Multiple resources
Single level	[5–7]	[8–12]	
Serial	[5,13,14]	[13]	
Assembly	[5,15–17]	[18–21]	
General	[1,5,22–26,4]	[27,28]	[3,29,30]

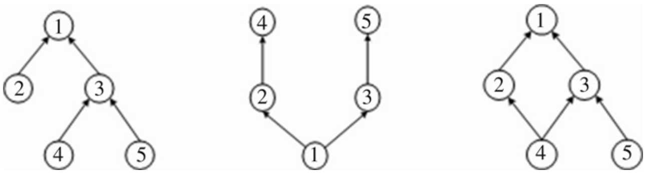

Fig. 1. Three major types of product structures.

structure problem; Dellaert and Jeunet [25] designed randomized heuristics for uncapacitated general structure problems; Jeunet and Jonard [26] developed single-point stochastic search algorithms for uncapacitated general structure problems and Pitakaso et al. [4] presented a max–min ant system for uncapacitated general structure problems. ECM may not get the optimal solution to a problem, but it takes little effort to reach a near-optimal solution. A PSO algorithm, which is one of the ECM, was developed by Kennedy and Eberhart in 1995 [31]. The original intent of PSO was to graphically simulate the graceful but unpredictable choreography of a bird flock. PSO exhibits common evolutionary computation attributes including: (1) it is initialized with a population of random solutions, (2) it searches for optima by updating generations, and (3) potential solutions, called particles, are then "flown" through the problem space by following the current optimum particles [32]. So far, few people have adopted a PSO algorithm to solve the MLLS problem. Here we proposed a PSO algorithm for solving an uncapacitated MLLS problem with assembly structure and the feasibility and effectiveness of our algorithm are investigated. Also, we plan to extend this approach to general structure problems with limited and unlimited capacities.

This paper is organized as follows: Section 2 is dedicated to the presentation and mathematical formulation of the MLLS problem. In Section 3, a brief introduction of a PSO algorithm and the framework of the proposed algorithm will be stated. The experimental frameworks and the computational results will be presented in Section 4. Finally, the conclusion and outlook can be found in Section 5.

2. Model formulation

In the MLLS problem, there are three major product structures: (1) assembly structure (2) arborescent structure (3) general structure. Fig. 1 shows three major types of product structures.

It is rather common to represent the bill of materials as a directed acyclic graph. In such a graph each node corresponds to an item and each edge (i, j) between node i and node j indicates that item i is directly required to assemble item j. Here, $\Gamma^{-1}(i)$ and $\Gamma(i)$ are used to present the sets of immediate predecessors and immediate successors of node i. The set of ancestors-immediate and non-immediate predecessors of item i is denoted by $\hat{\Gamma}^{-1}(i)$ and the set of all successors by $\hat{\Gamma}(i)$. In the last structure, product 1 is a finished good. So, we have for example $\Gamma^{-1}(1) = \{2, 3\}$; $\hat{\Gamma}^{-1}(1) = \{2, 3, 4, 5\}$; $\Gamma(4) = \{2, 3\}$; $\hat{\Gamma}(4) = \{1, 2, 3\}$ [24].

We assume that the production structure includes only one finished good (product). Variable cost parameters and variable purchase or production costs are not taken into account. In addition, we assume that no component is sold to an outside buyer, i.e. independent demands only exist for finished goods. Furthermore, no backlogging is allowed and lead-times of all items are zero. For the sake of simplicity, we assume that neither positive initial inventories ($I_{i,0} = 0, \forall i$) nor scheduled receipts are introduced. In such a context net requirements equal gross requirements for any item in the product structure [24]. The MLLS problem is a mixed integer programming problem. So we describe this problem with the following notations:

- i the index of item
- $C_{i,j}$ quantity of item i required to produce a unit of item j
- H_i unit inventory carrying cost for item i
- K_i set up cost for item i
- l_i lead time to assemble, to manufacture or to purchase item i
- $I_{i,t}$ inventory level of item i at the end of period t
- $a_{i,t}$ a binary decision index addressed to capture the set-up cost for item i delivered in period t
- $D_{i,t}$ requirements for item i in period t
- $P_{i,t}$ the amounts of production/replenishment for item i in period t
- M a very big number