European Journal of Operational Research 235 (2014) 102–114

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

A variable neighborhood search with an effective local search for uncapacitated multilevel lot-sizing problems

Yiyong Xiao [a], Renqian Zhang [b], Qiuhong Zhao [b,*], Ikou Kaku [c], Yuchun Xu [d]

[a] School of Reliability and Systems Engineering, Beihang University, Beijing 100191, China
[b] School of Economics and Management, Beihang University, Beijing 100191, China
[c] Department of Environmental Management, Tokyo City University, Yokohama 224-8551, Japan
[d] School of Applied Sciences, Cranfield University, Cranfield, Bedford MK43 0AL, UK

ARTICLE INFO

Article history:
Received 4 May 2012
Accepted 8 October 2013
Available online 20 October 2013

Keywords:
Metaheuristics
Multilevel lot-sizing (MLLS) problem
ADTS local search
Variable neighborhood search (VNS)

ABSTRACT

In this study, we improved the variable neighborhood search (VNS) algorithm for solving uncapacitated multilevel lot-sizing (MLLS) problems. The improvement is twofold. First, we developed an effective local search method known as the Ancestors Depth-first Traversal Search (ADTS), which can be embedded in the VNS to significantly improve the solution quality. Second, we proposed a common and efficient approach for the rapid calculation of the cost change for the VNS and other generate-and-test algorithms. The new VNS algorithm was tested against 176 benchmark problems of different scales (small, medium, and large). The experimental results show that the new VNS algorithm outperforms all of the existing algorithms in the literature for solving uncapacitated MLLS problems because it was able to find all optimal solutions (100%) for 96 small-sized problems and new best-known solutions for 5 of 40 medium-sized problems and for 30 of 40 large-sized problems.

1. Introduction

The multilevel lot-sizing (MLLS) problem addresses how to best determine the trade-off cost in a production system with the purpose of satisfying customer demand with a minimum total cost. The MLLS problem plays an important role in modern production systems of manufacturing and assembly firms. Many planning systems, such as Material Request Planning (MRP) and Master Product Scheduling (MPS), depend heavily on the basic mathematical model and solution approaches for the MLLS problem. Nevertheless, the MLLS problem was proven to be strongly NP-hard (Arkin, Joneja, & Roundy, 1989). Optimal solutions to large-sized problems with complex product structures are notably difficult to find. In addition, optimal algorithms exist only for small-sized MLLS problems, and these algorithms include dynamic programming formulations (Zangwill, 1968, 1969), an assembly–structure-based method (Crowston & Wagner, 1973), and branch-and-bound algorithms (Afentakis & Gavish, 1986; Afentakis, Gavish, & Kamarkar, 1984). Many heuristic approaches have been developed to solve the MLLS problem and its variants with near-optimal solutions. Early studies first applied sequential applications of single-level lot-sizing models to each component of the product structure (Veral & LaForge, 1985; Yelle, 1979), and later studies used an approximate application of multilevel lot-sizing models (Blackburn & Millen, 1982, 1985; Coleman & McKnew, 1991).

The uncapacitated MLLS acts as a fundamental problem, and its solution approach could be highly meaningful to many of its extended versions, including the capacitated MLLS, the MLLS with time-windowing, and the MLLS with order acceptance. In practice, many SME firms in China's electromechanical industry are more willing to adopt dynamic capability policies because they can improve their capacities during busy seasons with many methods, such as extra working-time, temporal employment, and rented machines. Therefore, the uncapacitated MLLS model caters to the situations of their ERP systems.

Over the past decade, several metaheuristic algorithms have been developed to solve uncapacitated MLLS problems with complex product structures. It has been reported that these algorithms are capable of providing highly cost-efficient solutions with a reasonable computing load. Dellaert and Jeunet (2000) and Dellaert, Jeunet, and Jonard (2000) first presented a hybrid genetic algorithm (HGA) for solving uncapacitated MLLS problems with a general product structure and introduced a competitive strategy for mixing the use of five operators in the evolution of the chromosomes from one generation to the next. Homberger (2008) presented a parallel genetic algorithm (PGA) and an empirical policy for deme migration (rate, interval, and selection) for the MLLS problem. These researchers used the power of parallel calculations

* Corresponding author. Address: 37 Xueyuan Road, Haidian District, Beijing 100191, China. Tel.: +86 10 82316181; fax: +86 10 82328037.

E-mail addresses: xiaoyiyong@buaa.edu.cn (Y. Xiao), zhangrenqian@buaa.edu.cn (R. Zhang), qhzhao@buaa.edu.cn (Q. Zhao), kakuikou@tcu.ac.jp (I. Kaku), yuchun.xu@cranfield.ac.uk (Y. Xu).

0377-2217/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2013.10.025

to decentralize the large calculation load over multiple processors (30 processors were used in their experiments). In addition to genetic algorithms, other metaheuristic algorithms, such as simulated annealing (SA) algorithms (Homberger, 2010; Tang, 2004), the particle swarm optimization (PSO) algorithm by Han, Tang, Kaku, and Mu (2009), the MAX–MIN ant colony optimization (ACO) system by Pitakaso, Almeder, Doerner, and Hartl (2006, 2007), and the soft optimization approach (SOA) based on segmentation by Kaku and Xu (2006) and Kaku, Li, and Xu (2010), have been developed for solving uncapacitated MLLS problems.

The variable neighborhood search (VNS) algorithm initiated by Mladenovic and Hansen (1997) is a top-level methodology for solving optimization problems. Because its principle is simple and easy to understand and implement, various VNS-based algorithms have been successfully applied to many optimization problems (Hansen, Mladenovic, & Moreno-Perez, 2008a, 2008b, 2010). Mladenovic, Urosevic, Hanafi, and Ilic (2012) presented a new schema of the general variable neighborhood search (GVNS), which is an extended version of the basic VNS that considers multiple neighborhood structures. Labadie, Mansini, Melechovsky, and Calvo (2012) proposed a VNS procedure based on the idea of exploring, most of the time, granular instead of complete neighborhoods in order to improve the algorithms efficiency without loosing effectives. A brief summarization of recent successful VNS applications can be found in Mladenovic, Kratica, Kovacevic-Vujcic, and Cangalovic (2012).

Xiao, Kaku, Zhao, and Zhang (2011a) first developed a VNS-based algorithm for basic schema and a *shift rule* to solve small- and medium-sized MLLS problems; this algorithm performed better than the HGA in small- and medium-sized problems. Xiao, Kaku, Zhao, and Zhang (2011b) developed a reduced VNS (RVNS) combined with six SHAKING techniques to solve large-sized MLLS problems. The term "reduced" indicates a simplified version of the classical VNS algorithm because the *local search* (the most time-consuming component of VNS) was removed from the basic scheme. Although RVNS is still a generate-and-test algorithm, it differs significantly from the single-point stochastic search (SPSS) algorithm (Jeunet & Jonard, 2005) because it uses a systematic method to change multiple bits (with a maximum $K_{\max}$) in the incumbent to generate a candidate, whereas the latter changes only a single bit. In the study conducted by Xiao, Kaku, Zhao, and Zhang (2012), three indices (i.e., the *distance*, *changing range*, and *changing level*) were proposed for a neighborhood search based on which three hypotheses were verified and can be used as common rules to enhance the performance of any existing generate-and-test algorithm. Using these three hypotheses, the proposed iterated neighborhood search (INS) algorithm delivered notably good performance when tested against 176 benchmark problems.

In our previous research, the neighborhood structure is defined under a distance-based metric that measures the distance (of two solutions) using the number of different bits, and another type of neighborhood structure based on problem decomposition was also studied in the recent literature (Helber & Sahling, 2010; Lang & Shen, 2011; Seeanner, Almada-Lobo, & Meyr, 2013). Several decomposition methods (i.e., product-oriented, time-oriented, and resource-oriented) combined with fix-and-optimized (or partial optimization) strategies were adapted to decompose the original problem into multiple sub-problems in order to restrain the optimization to a smaller area of the binary variables.

In this paper, we developed an effective local search procedure known as the Ancestors Depth-first Traversal Search (ADTS) for the RVNS algorithm such that the RVNS algorithm can be restored to a standard VNS. Although the ADTS procedure adds a considerable amount of computing load to the algorithm, we successfully developed an efficient method (known as trigger) using a new formulation of MLLS problems to rapidly calculate the change in the objective cost during the neighborhood search process. Thus, the new VNS algorithm is both effective and efficient in solving MLLS problems with high-quality solutions and within an acceptable computing time.

The remainder of this paper is organized as follows. In Section 2, we describe the new formulation of the MLLS problem. In Section 3, we detail a local search procedure known as ADTS which was added to our previously presented RVNS algorithm for effectively solving the MLLS problem. Section 4 outlines a highly efficient approach for rapidly calculating the cost variation of the objective function as the incumbent solution changes. In Section 5, we test the proposed algorithm on 176 benchmark MLLS problem instances of different scales (small, medium, and large) and compare its performance with that of existing methods. Finally, Section 6 presents our concluding remarks.

2. Problem formulation and neighborhood definition

The MLLS problem under investigation is considered an uncapacitated, discrete-time, multilevel production/inventory system with a general product structure[1] and multiple-end items. We assume that external demands for the end items are known throughout the planning horizon and that backlog is not allowed. Below, we present the notations used to model the MLLS problem, which also can be found in the reports by Dellaert and Jeunet (2000) and Xiao et al. (2012).

- i: Index of items, $i = 1,2,\ldots,m$.
- t (and t'): Index of periods, $t = 1,2,\ldots,n$.
- H_i: Unit inventory holding cost for item i.
- S_i: Setup cost for item i.
- d_{it}: External demand for item i in period t.
- D_{it}: Total demand for item i in period t.
- C_{ij}: Quantity of item i required to produce one unit of item j.
- Γ_i: The set of immediate successors of item i.
- Γ_i^{-1}: The set of immediate predecessors of item i.
- l_i: The lead time required to assemble, manufacture, or purchase item i.

The decision problem focuses on how to set the production setup for all of the items in all of the planning periods such that the decision variable is an $m \times n$ matrix denoted as follows:

- Y_{it}: Binary decision variable addressed to capture the setup cost for item i in period t.

Depending on the decision variable, two other important variables are addressed to quickly capture the inventory holding costs. These can be introduced as follows:

- P_{it}: The period in which the demands of item i in period t will be set for production.
- X_{it}: The production quantity for item i in period t.

The objective function is to minimize the sum of the setup cost and the inventory holding cost for all of the items over the entire planning horizon and is denoted by TC (total cost). We extend the formulation described by Xiao et al. (2012) to cover the external demand for non-end items. Thus, the uncapacitated MLLS problem can be modeled as follows:

[1] In a pure assembly structure, each item has multiple immediate predecessors but at most only one direct successor; in a general structure, each item can have multiple immediate predecessors and multiple direct successors.