Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 29 (2012) 1137 – 1141

Procedia Engineering

www.elsevier.com/locate/procedia

2012 International Workshop on Information and Electronics Engineering (IWIEE)

A Note on "A MAX-MIN Ant System for Unconstrained Multi-Level Lot-Sizing Problems"

Yi Han[a], Jianhu Cai[a,*], Ikou Kaku[b], Yingsong Li[c], Huazhen Lin[a], Xuhong Ye[a]

[a]College of Economics and Management, Zhejiang University of Technology, Hangzhou 310023, China
[b]Department of Management Science and Engineering, Akita Prefectural University, Yulihonjo 015-0055, Japan
[c]College of Information and Communications Engineering, Harbin Engineering University, Harbin 150001, China

Abstract

Pitakaso et al.[1] presented an ant systems based on random cumulative Wagner-Whitin (RCWW) (RCWW-STVS) for uncapacitated multilevel lot-sizing (MLLS) problems and gave out a computational result for Dellaert's instance [2] with a random variable r = 0.43. The result is not quite right. This paper highlighted the error and presented a revision to the result.

Keywords: Multilevel lot-sizing, RCWW; ant systems, material requirements planning, metaheuristic

1. Introduction

Pitakaso et al. [1] presented a RCWW-STVS algorithm to solve multilevel lot-sizing problems without resource constraints. To prove their algorithm is superior to Dellaert et al. [2]'s RCWW algorithm, they presented a result reached by RCWW-STVS algorithm for reference [2]'s instance with product order of {2, 5, 1, 3, 4, 6, 8, 7, 9}, setup cost accumulative parameters r_i ={0.43, 0.43, 0.43, 0.43, 0.43, 0.43, 0.43, 0.43, 0.43} and the inventory costs for material 1 to 9 is {13, 8, 4, 4, 3, 3, 2, 1, 1}.

The product structure is shown in Fig.1. The underlined number is lead time for each material and the right side number is setup cost for each material. The inner demand relation between materials is 1 to 1.

The result from [1] is presented in Table 1. There are no errors for parameters but the result for

* Corresponding author. Tel.: +86-18957144810.
E-mail address: caijh@zjut.edu.cn .

1877-7058 © 2011 Published by Elsevier Ltd.
doi:10.1016/j.proeng.2012.01.101

reference [2]'s instance. Here we note in Table 1, the demand for material 1 is {0, 0, 0, 0, 10, 15, 10, 12, 20}. Material 1's production lot, according to [1], is {0, 0, 0, 0, 10, 15, 10, 12, 20}. Then for material 3, taking into consideration of material 1's lead time, the demand is {0, 0, 0, 10, 15, 10, 12, 20, 0}. As we seen from Table 1, the revised setup cost of material 3 is {0, 0, 0, 41.5, 30, 30, 30, 30, 30}.

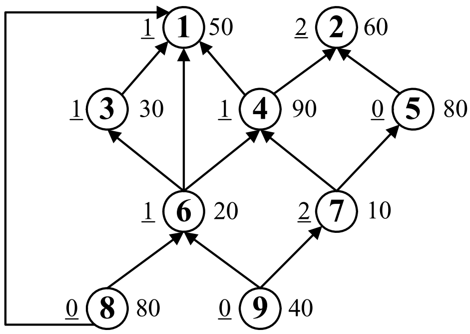

Fig.1 Product Structure.

The production lot of material 3 reached by [1] is {0, 0, 0, 10, 25, 0, 12, 20, 0}. However, we can find that the setup cost in period 6 of material 3 is 30 and the inventory cost in period 5 of material 3 is 4. If producing 10 products in period 6, the cost is 30. However, according to Pitakaso et al.[1]'s result, 10 units of material 3 are kept in inventory for 1 period. Then the cost is 10*4=40. So we can easily tell a production decision {0, 0, 0, 10, 15, 10, 12, 20, 0} is better than another one {0, 0, 0, 10, 25, 0, 12, 20, 0}. After a practical verification with C program language, we got a result, which is a revision to reference [1]'s result, for Dellaert et al.[2]'s instance and presented it in Table 2. The material presentation of MLLS problem is shown in Fig.2. Fig.3 shows the output interface of VC++ program.

```
Struct MATERIAL     //material structure
{
  int info;     //index of current material
  int level;    // level in which a material lies
  int processable;   //index to show whether this material can be processed
  float s;    //Setup cost
  float S1;    //transitional variable
  float h;    // inventory cost
  int leadtime; //lead time of a material
  float r;     // random cumulative variable
  int predecessor_number;   // the number of predecessors
  int successor_number;   //the number of successors
  int predecessor [M+1];    // all direct predecessors of current material
  int successor [M+1];    // all direct successors of current material
};
```

Fig.2 Struct Definition in C Language for Materials.

By the way, the results of Pitakaso et al. [1]'s are based on different product sequence. So, detailed presentations of each sequence to each solution are strongly needed. Only by doing so, the other scholars can reproduce those benchmark results.