Capacitated lot-sizing and production sequence problem with setups complexity

J. Behnamian[1], S.M.T. Fatemi Ghomi[2*], B. Karimi[2], M. Fadaei Moludi[2]

[1]*Department of Industrial Engineering, Faculty of Engineering, Bu-Ali Sina University, Hamedan, Iran*
[2]*Department of Industrial Engineering, Amirkabir University of Technology, Hafez Avenue, 15916-34311 Tehran, Iran.*

Abstract: This paper considered the multi-product multi-level multi-period capacitated lot-sizing and sequencing problem with setup carry-over and sequence-dependent family setup times and costs. A formulation of the problem was provided as a mixed-integer nonlinear programming model. To propose this formulation, first, the mixed-integer nonlinear of the problem was linearized, and then converted to a mixed-integer linear program. To solve large-size instances of the problem, then, a lower bound was provided. The results confirmed the efficiency of the proposed model compared to previous models in terms of the runtime and the number of defined variables and constraints. Since this problem is NP-hard and adding other factors such as family setups, setup carry over and sequence-dependent setups increase its complexity, in this paper, a genetic algorithm (GA) was applied in large-size dimensions and its results were compared with the proposed lower bound. The numerical results showed that there is no significant difference between the results of the proposed GA and lower bound, and, so, the GA had been able to approach the optimal solution.

Keywords: Lot-sizing and scheduling problem, Family setups, Sequence-dependent setup times and costs, Carry-over setups, Lower bound, Genetic algorithm.

1. Introduction

The subject of production and its importance has long been considered by researchers and craftsmen. Given the importance and complexity of this issue, production planning, which is one of the most important problem in this field, is of particular importance. The goal of production planning is the optimal use of production resources to produce products and goods according to market demand and unforeseen demands along the planning horizon. In this regard, determining lot sizing for each product is undoubtedly one of the key issues in production planning. This involves determining the time, amount and sequence of production, and it follows that at each stage of production, a setup time and cost is required to change production from one product to another; thus the concept of economics of scale requires that products are produced in the lot. On other hand, with this decision, by increasing the size of the lot, the amount of inventory and maintenance costs will increase. In fact, reducing the number of lots increases the cost and time of commissioning and reduces inventory and maintenance costs. Therefore, determining the size of production lots is intended to minimize the total system costs. Also, the problem under consideration will be much more difficult and complex if it is accompanied by capacity constraints. Setup is the preparation of the workstation to perform the operation. Two types of simple and complex setup structures are defined, considering each of which, due to changes in the number of binary variables in the model, affects the complexity of the problem [1]. If the setup time and cost in a period are independent of the sequence in the current period and previous one, the setup structure is simple, and otherwise, it is complex. The complex setup structure is divided into several categories. In this regard, if the change of production from one product to another leads to spending time and cost whose amount depends on the product sequence, it is called

* Corresponding author, E-mail address: Fatemi@aut.ac.ir, Tel: (+9821) 6454538, Fax: (+9821) 66459569

sequence-dependent setup times. In more complex cases, setup may perform at the end of a period, production may take place in the next time period, or it is possible to maintain the setup from the previous period to the current period [2]. On other hand, products with similarities in the manufacturing process can be classified into a family or group. In this case, the products have a major setup for the family or the main group with major time and cost, and have a minor setup for members of each family or subgroup with less time and cost. The problems with a complex setup, both in terms of modeling and solution, are much more difficult and important than problems with a simple setup structure [3].

One of the most common systems in the manufacturing industry is multi-level production systems with capacity limitations, each of which is likely to perform activities such as manufacturing, assembly, inspection, rework, or disposal. The main feature of these systems is to perform the process of each of the levels in a lot manner, which is due to the cost and time of setup at each level to perform the operation. Therefore, one of the most important and effective ways to control and reduce the cost of a production system is to make the right decision on the issue of determining the lot-size of products at each level of the production system. The capacitated lot-sizing problem (CLSP) is a tactical production problem that consists in deciding when and how many items to produce, minimizing the production costs assuring the demand constraints [4]. The multi-level CLSP with linked lot sizes (MLCLSP-L) is an extension of the big-bucket MLCLSP, allows to carry over the setup state of a resource to the next periods following the setup. However, it incorporates partial sequencing of the production orders in the sense that the first and the last products produced in a period are determined by the model [5]. The problem with setup carry-over, besides reducing setup cost, provides additional capacity with the lack of setup time between consecutive periods [6].

The proposed model in this paper determines the lot-size and production sequence at each level and period. In addition, production planning with family setup times and costs has been studied in this paper. This problem is a development of the classic MLCLSP-L. Since the multi-level multi-period multi-item capacitated lot-sizing problem is an NP-hard problem, adding other factors such as family setups, setup carry over and sequence-dependent setups increases its complexity. In this problem, product families are groups of items with similar setups. Thus, the total setup cost and time are reduced, so there are two types of major and minor setups. Major setup is done when to produce an item, setup from one family to another family is performed. Minor setup is the type of setup that from one product to another product in the same family is made. Florian et al. [7] and Bitran and Yanasse [8] investigated a class of production planning problems that known demand has to be satisfied over the finite horizon at minimum total costs and production and storage cost functions are specified for each period. They show that single item CLSPs are NP-hard. Chen and Thizy [9] show that the multi-item capacitated lot-sizing problem is strongly NP-hard. So multi-level multi-period multi-item production planning problem with setup carry-over and sequence-dependent family setup times that have more binary variables for the complexity of setting up is also strongly NP-hard. So, a genetic algorithm (GA) was applied in large-size instances and its results were compared with the proposed lower bound.

The remainder of this paper is organized as follows. In Section 2, the related literature is reviewed. In Section 3, a nonlinear mathematical model of the problem is introduced. In this section, linearization of the problem and a lower bound will also be presented. Section 4 describes the proposed GA. The performance of the presented model is validated in Section 5. Finally, we discuss the conclusions and future research directions in Section 6.

2. Literature review

Baldo et al. [10] studied a production lot-sizing and scheduling problem. In this research, as well as a mixed-integer programming (MIP), they presented MIP-based heuristics for the brewery industry. Gicquel and Minoux [11] considered the multi-product lot-sizing and scheduling problem with sequence-dependent changeover costs. They proposed a branch & cut with some valid inequalities. For the multi-level capacitated lot-sizing and scheduling problem in hen egg production planning, Boonmee and Sethanan [12] proposed a MIP model. To minimize the total cost, they also applied a variant of particle swarm optimization. Ceschia et al. [13] studied the discrete single-machine, multi-item lot-sizing and scheduling problem. After modeling the problem as a MIP model, to solve it, they proposed a simulated annealing algorithm.

Curcio et al. [14] investigated the flexible production system in the lot-sizing and scheduling problem under multistage demand uncertainty on a rolling-horizon planning scheme. They proposed an approximate heuristic algorithm based on the robust optimization concept. Wichmann et al. [15] introduced the energy-efficient lot-sizing and scheduling problem. They analysed the cost-saving potential in this approach in comparison with classical planning. Toscano et al. [16] studied a two-stage lot-scheduling problem with buffers, mandatory temporal cleanings for the preparation tanks and production lines, and production synchronization. They proposed a two-phase heuristic algorithm for this problem. Kaczmarczyk [17] addressed the proportional lot-sizing and scheduling problem with fictitious micro periods. In this regard, they designed a mixed-integer linear programming (MILP) model in which some valid inequalities were utilized. For lot-sizing and scheduling problem considering supplier flexibility in satisfying a fraction of demand, Hu and Hu [18] presented a stochastic/robust optimization approach. In their proposed approach, the demand and overtime processing cost were simultaneously uncertain. Considering a complex setup structure with hazardous materials, Mohammadi [19] integrated the lot-sizing with scheduling problem. This paper proposed a MIP model to maximize profit with demand choice flexibility in a trade-off and minimize the risk of hazardous materials incidents. The author also extended the chance-constrained programming model with uncertain demand.

Chen and Zhang [20] incorporated capital flow constraints and trade credit to lot-sizing problems. They formulate a mathematical model for this problem. After dividing the model into sub-linear problems without integer variables, it was solved with dynamic programming and a heuristic. To find the optimal order quantity, the number of shipments and the number of defective item disposals in Cheng et al. [21], after proving a vendor-buyer inventory model, a heuristic was proposed. Stadtler and Meistering [22] presented a model for the lot-sizing problem with capacity constraint in which both the periodic and cyclic service levels were studied. To minimize setup and holding costs, a lower bound is provided. Abrishami et al. [23] considered the integrated lot-sizing model with supplier problem considering multiple products and multiple purchasing items over multiple periods. To minimize the total purchase, inventory, production, and transportation costs, after proposing a MILP model, they proposed a GA. Slama et al. [24] considered the capacitated dynamic lot-sizing problem with external procurement, defective and back-ordered items, setup times, and extra capacity. In this study, to maximize the disassembly process, the MIP model was proposed. Li et al. [25] investigated a production planning problem under uncertain demand in which the considered problem consists of two subproblems: an assembly line balancing problem and a capacitated lot-sizing problem. They modeled this problem as two-stage stochastic programming and validated it by a case study. Malekian et al. [26] studied the single-item capacitated lot-sizing problem considering a 1-breakpoint all-units quantity discount. They, first, proved the properties of the optimal solution. Then, they proposed an implicit enumeration exact algorithm with speed-up techniques to reduce its time complexity. Finally, they presented a heuristic algorithm for large-size instances. Mula et al. [27] addressed the capacitated lot-sizing problem with sequence-dependent setups and parallel machines in a bi-part injection molding. They provided a MILP model for a second-tier supplier of the automotive sector. For the off-the-road tire industry, Koch et al. [28] considered a lot-sizing problem. Considering several real-world constraints in this problem, they introduced a mixed-integer linear program. In the context of cloud manufacturing, Ganstereet al. [29] introduced the capacitated multi-level lot-sizing problem with transshipments and setup carry-over with components that can only be produced by one specific agent. After modeling this problem mathematically, they proposed a matheuristic based on a fix-and-optimize procedure. Rezaei and Behnamian [30] studied a single-item lot-based supply and batch production under a bilateral capacity reservation contract based on a partnership structure. They proposed a mathematical model and dynamic programming algorithm for solving this problem.

The proposed model in the current paper is monolithic and is quickly solved. In addition to sequence-dependent family setups, this multi-level model, for the first time, considers setup carry-over.

3. Mathematical modeling

As stated in the previous sections, economic lot-size determination problems in the field of flowshop problems are modelled by the capacitated resources available in the subject literature, have a simple setup structure, or are assumed to have only one of the complex setups. Given that this assumption was not considered in any of the reviewed papers in the field of production planning, this paper generalizes the assumption of a complex setup structure in flowshop systems with limited production capacity.

In the following, after introducing the parameters, decision variables, and indices used in this research, based on Mohammadi and Fatemi Ghomi [31] and Behnamian et al. [32], the presented mathematical model will be described.

Parameters, and indices

T : number of periods

Z : number of levels

m : number of families

r_j : number of products in family *j*

c_{ijzt} : production cost per unit of product i of family *j* in level *z* during period *t*

h_{ijzt} : holding cost per unit of product *i* of family *j* in level *z* during period *t*

d_{ijt} : net demand for product *i* of family *j* during period *t*

c_{ijzt} : production cost per unit of product *i* of family *j* in level *z* during period *t*

q_{ijzt} : production time per unit of product *i* of family *j* in level *z* during period *t*

s_{ikjzt} : setup cost from product *i* of family *j* to product *k* of family *j* in level *z* during period *t*

ST_{ikjzt} :setup time from product *i* of family *j* to product *k* of family *j* in level *z* during period *t*

s_{ijzt} : setup cost from family *i* to family *j* in level *z* during period *t*

ST_{ijzt} : setup time from family *i* to family *j* in level *z* during period *t*

R_{zt} : available capacity in level z during period *t*

M : a big number

Decision variables

x_{ijzt} : production quantity (lot-size) of product *i* of family *j* in level *z* during period *t*

I_{ijzt} : inventory level of product *i* of family *j* in level *z* during period *t*

y_{jzt} : 1, if the setup of family *j* occurs in level *z* during period *t*; 0, otherwise.

z_{ijzt} : 1, if the setup of product *i* of family *j* occurs in level *z* during period *t*; 0, otherwise.

T_{kjzt} : 1, if the setup of family *j* occurs after setup of family *k* in level *z* during period *t*; 0, otherwise.

T_{kijzt} : 1, if the setup of product *i* occurs after setup of product *k* of family *j* in level *z* during period t; 0, otherwise.

a_{jzt} : 1, if family *j* is the latest family produced in level *z* during period *t*; 0, otherwise.

b_{jzt} : 1, if family *j* is the first family produced in level *z* during period *t*; 0, otherwise.

u_{jzt} : 1, if family *j* is single-family produced in level *z* during period *t*; 0, otherwise.

a_{ijzt} : 1, if product *i* of family *j* is the latest product produced in level *z* during period *t*; 0, otherwise.

b_{ijzt} : 1, if product *i* of family *j* is the first product produced in level *z* during period *t*; 0, otherwise.

u_{ijzt} : 1, if product *i* of family *j* is single product produced in level *z* during period *t*; 0, otherwise.

MA_{kijzt} : 1, if the setup of product *i* of family *j* that has been retained, occurs after setup of product *k* of family *j* in level *z* at the end of period *t*; 0, otherwise.

MB_{kijzt} : 1, if the setup of product *i* of family *j* that has been retained, occurs after setup of product *k* of family *j* in level *z* at the beginning of period *t*; 0, otherwise.

LA_{ijzt} : 1, if the setup of family *j* occurs after setup of family *i* in level *z* at the end of period *t*; 0, otherwise.

LB_{ijzt} : 1, if the setup of family *j* occurs after setup of family *i* in level *z* at the beginning of period *t*; 0, otherwise.

γ_{jzt} : 1, if the setup of family *j* is retained in level *z* from period *t*-1 to period *t*; 0, otherwise.

3.1. Mathematical formulation

The proposed model, denoted as Model P[1], can be formulated as follows:

$$\text{Min } Z = \sum_{t=1}^{T}\sum_{z=1}^{Z}\sum_{j=1}^{m}\sum_{i=1}^{r_j}\left(c_{ijzt}.x_{ijzt} + h_{ijzt}.I_{ijzt}\right) + \sum_{t=1}^{T}\sum_{z=1}^{Z}\sum_{j=1}^{m}\sum_{i=1}^{r_j}\sum_{k=1}^{r_j}\left(s'_{kijzt}.T'_{kijzt}\right) + \sum_{t=1}^{T}\sum_{z=1}^{Z}\sum_{j=1}^{m}\sum_{i=1}^{m}\left(s_{ijzt}.T_{ijzt}\right) \quad (1)$$

$$\text{s.t.: } I_{ijZt-1} + x_{ijZt} - I_{ijZt} = d_{ijt} \qquad \forall i, j, t \quad (2)$$

$$x_{ijzt} = I_{ij(z-1)(t-1)} + x_{ij(z-1)t} - I_{ij(z-1)t} \qquad \forall i, j, z, t \quad (3)$$

$$\sum_{j=1}^{m}\sum_{i=1}^{r_j}\left(x_{ijzt}.q_{ijz}\right) + \sum_{j=1}^{m}\sum_{i=1}^{m} ST_{ijzt}.(T_{ijzt} + LA_{ijzt} + LB_{ijzt}) + \sum_{j=1}^{m}\sum_{i=1}^{r_j}\sum_{k=1}^{r_j} ST'_{kijzt}.(T'_{kijzt} + MA_{kijzt} + MB_{kijzt}) \le R_{zt} \forall z, t \quad (4)$$

$$z_{ijzt} \le y_{jzt} \qquad \forall i, j, z, t \quad (5)$$

$$x_{ijzt} \le M.z_{ijzt} \qquad \forall i, j, z, t \quad (6)$$

$$u_{jzt-1} + a_{jzt-1} + b_{izt} + u_{izt} - 1 \le LA_{jizt-1} + LB_{jizt} \qquad \forall j, i, z, t = 2, \ldots, T \quad (7)$$

$$LA_{jizt-1} + LB_{jizt} \le 1 \qquad \forall i, j, z, t \quad (8)$$

$$\sum_{k=1,k\neq j}^{m} T_{\mathrm{kjzt}} = y_{jzt} - a_{jzt} - u_{jzt} \qquad \forall j,z,t \qquad (9)$$

$$\sum_{k=1,k\neq j}^{m} T_{\mathrm{jkzt}} = y_{jzt} - b_{jzt} - u_{jzt} \qquad \forall j,z,t \qquad (10)$$

$$\left(\sum_{j=1}^{m} y_{jzt}\right)\cdot\left(\sum_{j=1}^{m} u_{jzt}\right) \leq 1 \qquad \forall z,t \qquad (11)$$

$$u_{jzt} \leq y_{jzt} \qquad \forall j,z,t \qquad (12)$$

$$u_{jzt-1} + a_{jzt-1} + b_{jzt} + u_{jzt} - 1 \leq \gamma_{jzt} \qquad \forall j,z,t = 2,\ldots,T \qquad (13)$$

$$u'_{ijzt-1} + a'_{ijzt-1} + b'_{kjzt} + u'_{kjzt} - 1 \leq MA_{ikjzt-1} + MB_{ikjzt} \qquad \forall i,k,j,z,t = 2,\ldots,T \qquad (14)$$

$$MA_{ikjzt} + MB_{ikjzt} \leq \gamma_{jzt} \qquad \forall i,k,j,z,t \qquad (15)$$

$$\sum_{\mathrm{k=1,k\neq i}}^{\mathrm{r_j}} T'_{ik\mathrm{jzt}} = \mathrm{z_{ijzt}} - a'_{\mathrm{ijzt}} - u'_{\mathrm{ijzt}} \qquad \forall i,j,z,t \qquad (16)$$

$$T'_{ik\mathrm{jzt}} = \mathrm{z_{ijzt}}.\mathrm{z_{kjzt}} \qquad \forall i,j,z,t \qquad (17)$$

$$\left(\sum_{i=1}^{\mathrm{r_j}} z_{ijzt}\right)\cdot\left(\sum_{i=1}^{\mathrm{r_j}} u'_{ijzt}\right) \leq 1 \qquad \forall j,z,t \qquad (18)$$

$$u'_{\mathrm{ijzt}} \leq \mathrm{z_{ijzt}} \qquad \forall i,j,z,t \qquad (19)$$

$$\sum_{\mathrm{k=1,k\neq i}}^{\mathrm{r_j}} T'_{k\mathrm{ijzt}} = z_{ijzt} - b'_{ijzt} - u'_{ijzt} \qquad \forall i,j,z,t \qquad (20)$$

$$a_{jzt} + b_{jzt} + u_{jzt} \leq 1 \qquad \forall j,z,t \qquad (21)$$

$$a'_{ijzt} + b'_{ijzt} + u'_{ijzt} \leq 1 \qquad \forall j,z,t \qquad (22)$$

$$\sum_{j=1}^{m} a_{jzt} + u_{jzt} \leq 1 \qquad \forall z,t \qquad (23)$$

$$\sum_{j=1}^{m} b_{jzt} + u_{jzt} \leq 1 \qquad \forall z,t \qquad (24)$$

$$\sum_{i=1}^{r_j} a'_{ijzt} + u'_{ijzt} = y_{jzt} \qquad \forall j,z,t \qquad (25)$$

$$\sum_{i=1}^{r_j} b'_{ijzt} + u'_{ijzt} = y_{jzt} \qquad \forall j,z,t \qquad (26)$$

$$y_{\mathrm{jzt}},\ z_{\mathrm{ijzt}},\ T_{\mathrm{kjzt}},\ T'_{k\mathrm{ijzt}},\ a_{jzt},\ b_{jzt},\ u_{jzt},\ a'_{ijzt},\ b'_{ijzt},\ u'_{ijzt}, MA_{\mathrm{kijzt}}, MB_{\mathrm{kijzt}}, LA_{ijzt}, LB_{ijzt},\ \gamma_{jzt} \in \{0,1\} \qquad (27)$$

$$x_{ijzt}, I_{ijzt} \geq 0 \qquad (28)$$

$I_{ijz0} = 0$ (29)

The objective function (1) minimizes the sum of production, holding and sequence-dependent setup costs for families and products. Constraints (2) and (3) are the production-inventory balance equations. Constraints (4) ensure that the total production and setup time in each period and the level do not exceed the available capacity. Requirement (5) guarantees that a product is produced only if its family has been setup. Constraints (6) ensure that a product is produced if that product has been setup. Constraints (7) represent if family *j* is latest or single-family in level *z* during period *t*-1 and family *i* is first or single-family in level *z* and period *t*, setup from family *j* to family *i* occurs in level *z* at the end of period *t*-1 or at the beginning of period *t*. Constraints (8) show that at least one of the last two variables is 1. Constraints (9) represent that if the setup of family *j* has occurred and family *j* is latest or single-family in level *z* and during period *t*, then $\sum_{k=1,k\neq j}^{m} T_{kjzt} = 0$ necessarily, otherwise $\sum_{k=1,k\neq j}^{m} T_{kjzt} \leq 1$. Constraints (10) represent that if the setup of family j has occurred and family *j* is the first or single-family in level *z* and during period *t*, then $\sum_{k=1,k\neq j}^{m} T_{jkzt} = 0$ necessarily, in otherwise $\sum_{k=1,k\neq j}^{m} T_{jkzt} \leq 1$. Constraints (11) and (12) ensure that a family is single-family that is produced in level *z* and period *t* if the value of $\sum_{j=1}^{m} y_{jzt} = 1$, otherwise the value of $\sum_{j=1}^{m} u_{jzt} = 0$. Constraints (14) represent if product *i* of family *j* is the latest or single product in level *z* during period *t*-1 and product *k* of family *j* is first or single product in level *z* and period *t*, setup from product i to product *k* occurs in level *z* at the end of period *t*-1 or at the beginning of period *t*. Constraints (15) show that at least one of the last two variables is 1. Constraints (16) represent that if the setup of product *i* of family *j* has occurred and product *i* is the latest or single product of family *j* in level *z* and during period *t* then $\sum_{k=1,k\neq j}^{m} T_{ikjzt} = 0$ necessarily, otherwise $\sum_{k=1,k\neq j}^{m} T_{ikjzt} \leq 1$. Constraints (17) ensure that $T'_{ikjzt} = 1$ if the only product *i* and product *k* of family *j* produced in level *z* during period *t*. Constraints (18) and (19) ensure that a product is a single product that is produced in level *z* and period *t* if the value of $\sum_{i=1}^{r_j} z_{ijzt} = 1$, otherwise the value of $\sum_{i=1}^{r_j} u'_{ijzt} = 0$. Constraints (20) represent that if the setup of product *i* of family *j* has occurred and product *i* is first or single product of family *j* in level *z* and during period *t* then $\sum_{k=1,k\neq i}^{r_j} T'_{kijzt} = 0$ necessarily, otherwise $\sum_{k=1,k\neq j}^{m} T_{ikjzt} \leq 1$. Constraints (21) and (22) show that a product or a family at most is one of first, single, or the latest families and products, respectively. Constraints (23) to (26) represent that at most, one of the families or one of the products is first, single and the latest family or product in level *z* during period *t*. Constraints (27) show binary variables. Constraints (28) ensure that lot-size and inventory level can be greater than or equal to zero. And constraints (29) represent that the initial inventory is 0.

3.2. Model linearization

Model P^1 is nonlinear due to the quadratic terms in constraints (11), (17) and (18). The nonlinear model can be converted into an equivalent linear model through two non-negative variables. These variables are defined as follows:

ω_{zt} : positive if at least one family is produced in level *z* during period *t*; otherwise, 0.

ω'_{jzt} : positive if at least one product of family *j* is produced in level *z* during period *t*;

Otherwise, 0.

The linearized integer programming model (called Model P^2) is as follows:

$$\text{Min}\, Z = \sum_{t=1}^{T}\sum_{z=1}^{Z}\sum_{j=1}^{m}\sum_{i=1}^{r_j}\left(c_{ijzt}.x_{ijzt} + h_{ijzt}.I_{ijzt}\right) + \sum_{t=1}^{T}\sum_{z=1}^{Z}\sum_{j=1}^{m}\sum_{i=1}^{r_j}\sum_{k=1}^{r_j}\left(s'_{kijzt}.T'_{kijzt}\right) + \sum_{t=1}^{T}\sum_{z=1}^{Z}\sum_{j=1}^{m}\sum_{i=1}^{m}\left(s_{ijzt}.T_{ijzt}\right) \tag{1}$$

s.t.: $$\sum_{i=1}^{r_j} z_{ijzt} \geq y_{jzt} \qquad \forall j, z, t \tag{30}$$

$$\sum_{j=1}^{m} y_{jzt} \geq \omega_{zt} \qquad \forall z, t \tag{31}$$

$$\sum_{i=1}^{r_j} z_{ijzt} \geq \omega'_{jzt} \qquad \forall z, t \tag{32}$$

$$\sum_{j=1}^{m} a_{jzt} \leq \omega_{zt} \qquad \forall z, t \tag{33}$$

$$\sum_{i=1}^{r_j} a'_{ijzt} \leq \omega'_{jzt} \qquad \forall z, t \tag{34}$$

$$0 \leq \omega_{zt} \leq 1 \qquad \forall z, t \tag{35}$$

$$0 \leq \omega'_{jzt} \leq 1 \qquad \forall j, z, t \tag{36}$$

And Constraints (2)-(10), (12)-(16), (19)-(29).

3.3. A numerical example

To better describe the mathematical model, a small-size instance is designed and solved. In this regard, consider a flowshop with two machines that produces five products in two families for a three-period programming horizon (T = 3; Z = 2; N = 5; F = 2; r_1 = 2; r_2 = 3). Tables (1) to (5) show the parameters used in this instance.

******** Insert Table 1 here ********

******** Insert Table 2 here ********

******** Insert Table 3 here ********

******** Insert Table 4 here ********

******** Insert Table 5 here ********

The family setup, sequence-dependent setup and carry-over setup times are generated randomly, but mentioning them prolongs the subject. The obtained optimal solution of the problem is 3120 and the sequence of production of products and the lot-size of production are shown in Figure (1).

******** Insert Figure 1 here ********

3.4. Development of a lower bound

One of the common ways to evaluate the performance of algorithms is to use the lower bound of the optimal solution [33]. In this regard, first, suppose a lower bound is obtained from solving model P^1. Model P^1 is the mathematical model that its binary variables are relaxed between 0 and 1. The proposed lower bound is obtained from solving model P^2. Model P^2 is achieved by adding Constraints (37) and (38) to the model M_1.

$$\sum_{k=1}^{m} T_{kjzt} \leq 1 \qquad \forall j, z, t \tag{37}$$

$$\sum_{i=1}^{r_j} T'_{ikjzt} \leq 1 \qquad \forall k, j, z, t \qquad (38)$$

Theorem 1: P^2 is a lower bound for the problem.

Proof 1: Suppose there is a (*j*,*z*,*t*) in the optimal solution that infracts Constraint (37), namely $\sum_{k=1}^{m} T_{kjzt} \geq 2$. Assume $T_{k_1jzt} = 1$ and $T_{k_2jzt} = 1$. Suppose only families k_1, k_2 and *j* are produced in level *z* and period *t*. These variables do not include carry-over variables. If at the best case setup cost is:

$$setupcost = T_{k_1jzt}.S_{k_1jzt} + T_{jk_2zt}.S_{jk_2zt} + T_{k_2jzt}.S_{k_2jzt} \qquad (39)$$

According to triangle inequality:

$$S_{k_1jzt} + S_{jk_2zt} \geq S_{k_1k_2zt} \qquad (40)$$

Thus:

$$T_{k_1jzt}.S_{k_1jzt} + T_{jk_2zt}.S_{jk_2zt} + T_{k_2jzt}.S_{k_2jzt} \geq T_{k_1k_2zt}.S_{k_1k_2zt} + T_{k_2jzt}.S_{k_2jzt} \qquad (41)$$

Now suppose only families k_1, k_2, k_3 and j are produced in level *z* and period *t*. If at the best case setup cost is:

$$setupcost = T_{k_1jzt}.S_{k_1jzt} + T_{jk_2zt}.S_{jk_2zt} + T_{k_2k_3zt}.S_{k_2k_3zt} + T_{k_3jzt}.S_{k_3jzt} \qquad (42)$$

According to triangle inequality:

$$S_{k_1jzt} + S_{jk_2zt} \geq S_{k_1k_2zt} \qquad (43)$$

Thus:

$$T_{k_1jzt}.S_{k_1jzt} + T_{jk_2zt}.S_{jk_2zt} + T_{k_2k_3zt}.S_{k_2k_3zt} + T_{k_3jzt}.S_{k_3jzt} \geq T_{k_1k_2zt}.S_{k_1k_2zt} + T_{k_2k_3zt}.S_{k_2k_3zt} + T_{k_3jzt}.S_{k_3jzt} \qquad (44)$$

This proof is true for all possible sequences and $\sum_{k=1}^{m} T_{kjzt} \leq 1$. Therefore Constraint (37) always applies to the optimal solution and it can prove similarly that Constraint (38) always applies to the optimal solution thus P^2 is a lower bound for the main model. P^2 without binary variables reduces the region of possible solution and runtime. This shows the superiority of this lower bound.

The proposed model, due to its NP-hardness, cannot be solved in large-size dimensions. Therefore, in the next subsection, a GA will be presented for solving large-size instances.

4. Genetic algorithm

As mentioned in the previous subsections, the problem under consideration is strongly NP-hard. So a meta-heuristic algorithm is required for solution in real dimensions. Meta-heuristic methods are efficient approaches to solving complex integer programming problems, all of which use an intelligent random search process in problem-solving to achieve a near-optimal solution. In this regard, to solve capacitated lot-sizing problems in real dimensions, different meta-heuristic methods such as tabu search, simulated annealing, etc., have been developed [34]. In this paper, a GA is used to solve the problem. This algorithm has always been one of the most widely used meta-algorithms [35] that has always been considered by researchers in various fields of optimization [36].

Genetic algorithm is an intelligent and probabilistic search method that simulates Darwin's evolution theory by considering a population of solutions (each solution is called a chromosome) and the use of its operators, *i.e*., crossover and mutation, in each reproduction. In the GA, each solution in the population is evaluated according to its fitness function, and the solutions with better fitness have more opportunities for reproduction. At this stage, offspring are generated and replaced with unsuitable solutions in the current population. In other words, the combination of existing solutions through the crossover as well as mutation and selection operators produces new solutions and this cycle is repeated until it reaches a stopping criterion [37]. In this section, the implementation detail of a GA is presented.

Chromosome representation: In this paper, as shown in Figure 2, chromosomes are represented as matrices with $\sum_{j=1}^{m} r_j \times Z \times T$ dimensions. Fig. 1depicts a sample chromosome for a specific iteration of a problem with m=2, Z=5, T=3, r_1=3 and r_2=5.

******** Insert Figure 2 here ********

According to the model structure, a product is produced or not produced at a specific level and period. So a gene is considered as a product whose value is 1 if the product would be produced, otherwise 0.

The binary system is the best system to display chromosomes. Thus variables 0 and 1 are identified and the problem is converted to a linear programming problem. So production and inventory levels are specified through solving the problem.

- *Initial population:* Chromosomes are produced randomly. There are no repetitive chromosomes in this stage. If the chromosome is not feasible, it will be removed. Population size in this GA is the same as the algorithm of Mohammadi and Fatemi Ghomi [31]; they tested population size (3Z, 4Z and 5Z) in their article.
- *Selection operator:* The roulette wheel has been used to undergo a selection operation.
- *Crossover operation:* In this study, a two-point crossover (SB2OX) has been used. In addition, the two-point crossover is another used operator in this article. In this operator, two crossover points are selected randomly within a chromosome; then, the two parent chromosomes are interchanged between these points to produce two new offsprings.
- *Mutation operator:* Here, the shift mutation is used in this paper because this mutation has been appropriate performance in the sequence-dependent scheduling problems [38].
- *Stopping criterion*: In this paper, the maximum number of generations has been set to reach the pre-set number of generations without improvement in the final best solution.

5. Computational results

In this section, the numerical results in two, small and large, dimensions are investigated. All computational experiments in this section were performed on an Asus laptop Intel core i7 with 2.2 GHz CPU and 4GB Ram. The integer linear programming model was solved with LINGO 8 and the algorithm was coded in Matlab.

5.1. Test problems

In order to evaluate the performance of our model, the numerical examples were generated randomly. In this regard, consider a production system with eight products in the plant is aggregated into two families. Each family contains multiple products. Family 1 is comprised of products 1 to 4 and family 2 is comprised of products 5 to 9. This production system has three levels. The planning horizon is 12 months and every four months is one period. The demands of the two families can be forecasted accurately according to production planners' expertise and are summarized, for each period, in Table 6.

******** Insert Table 6 here ********

All cost and time parameters can be collected from the plant. The sequence-dependent setup times and costs among the families and products and other parameters are specified as random numbers. Also, the demands generated in this paper follow normal distributions with different means and standard deviations determined by the actual data in Table 6.

5.2. Parameters tuning

In this subsection, with the aim of tuning the proposed algorithm, the full factorial design method is chosen in which all possible combinations of the following factors are tested. Note that for determining the size of the algorithm population, the choice is based on the number of levels of the problem to be solved because it is a very important factor in the complexity of the problem.

- Crossover type (CRT): 2 levels (two-point and SB2OX).
- Crossover probability (CRP): 3 levels (0.5, 0.6, 0.7).
- Mutation probability (MUP): 3 levels (0.1, 0.2, 0.3).
- Population size (PS): 3 levels (3M, 4M, 5M).

All the cited factors result in 2×3×3×3=54 different combinations and, thus 54 different GAs. Every algorithm is tested with a set of problems presented in Table 7.

******** Insert Table 7 here ********

There is one replicate for each combination; therefore, 4×54=216 problems have been solved. As a percentage of increase over the lower bound, based on Mohammadi et al. [39], the response variable is $\sum_{i=1}^{4}[(Heu_{sol_i} - LB_i)/LB_i]\times 100/4$, in which Heu_{sol_i} is the solution obtained by a specific problem and LB_i is the lower bound for this particular problem. Performing the same procedure for all combinations and parameters, the following ‘‘best case’’ calibrations of the proposed algorithm are CRT: SB2OX, CRP: 0.6, MUP: 0.2, and PS: 5M.

5.3. Numerical results in small-size instances

In this subsection, we compare the proposed model with the model presented by Mohammadi and Fatemi Ghomi [31] in a small-size dimension. The number of positive variables, integer variables, and constraints of the proposed model is 2106, 1611, and 1579, respectively, while these values for the modified model proposed by them are 8100, 6561, and 6870, respectively. Therefore, our model with more elaborate features can be solved with less computational effort than the model proposed by Mohammadi and Fatemi Ghomi [31], in addition to considering family setup times and costs. The lower bound P^2 for this problem size has 2106 positive variables, 0 integer variables and 4333 constraints. This model does not contain an integer variable, so it is a linear model and can be solved more quickly. Table 8 shows the results of such tests. Note that the last column of this table shows the difference between the lower bound and the optimal solution objective functions.

******** Insert Table 8 here ********

Table 8 shows that the lower bound proposed in this paper is a tight bound and reduces runtime significantly. Also, we solved several problems with different dimensions by our proposed model and compared them with modified Mohammadi and Fatemi Ghomi [31]. The results are shown in Table 9.

******** Insert Table 9 here ********

This table indicates that the model presented in this paper is more efficient. According to the table above, two points must be noticed. First of all, the runtime of the two models is different, and as you can see in the table, this difference is very significant, and this is our proposed model, which takes much less time. The next point, which is the result of the difference in runtimes, is the limitation of the Mohammadi and Fatemi's model in solving larger-size instances so that, as shown in Table 4, due to the limitation of the runtime, the Mohammadi and Fatemi's model has not been able to find the solution in problem 4, while the proposed model of this research has been able to achieve an optimal solution in a much shorter time.

5.4. Numerical results in large-size instances

Table 10 shows a comparison between the developed lower bound against the proposed GA.

******** Insert Table 10 here ********

To verify the statistical validity of the results shown in Table 10 and confirm which the best algorithm between GA and LB is, a Kruskal–Wallis test as a non-parametric method has been performed. The obtained results are shown in Figures 3-4.

******** Insert Figure 3 here ********

As you can see, there is no significant difference between the results of the proposed GA and the lower bound, and this is an indication of the quality of the proposed algorithm to achieve the optimal solution.

******** Insert Figure 4 here ********

6. Conclusions and future research

This paper proposed a nonlinear model for multi-level, multi-product and multi-period production planning with setup carry-over and sequence-dependent family setup time. Then, it was converted into an equivalent linear model through two non-negative variables and eliminating three constraints and adding seven linear constraints. We compared this linear model and the existing model in the literature. The numerical results indicated the efficiency of our model because we could solve problems in a shorter time and a larger dimension. We also proposed a lower bound in order to decrease the solution space. Due to the Np-hardness of the considered problem, a genetic algorithm was also proposed to solve large-size instances. The proposed lower bound and the genetic algorithm were compared and the result showed that the obtained results showed that there is no significant difference between the genetic algorithm and lower bound. Applying other meta-heuristic algorithms to face this complex problem is suggested as a direction for future research. Also, considering stochastic demand and processing times is another area for future studies.

References

[1] Allahverdi, A. Ng, C.T. Cheng, T.C.E. and Kovalyov, M.Y. "A survey of scheduling problems with setup times or costs" *European Journal of Operational Research*, **187**(3), pp. 985–1032 (2008).

[2] Almada-Lobo, B., Klabjan, D., Carravilla, M.A. et al. "Single machine multi-product capacitated lot sizing with sequence-dependent setups" *International Journal of Production Research*, **45**(20), pp. 4873-4894 (2007).

[3] Jin, F. Dong, S. and Wu, C. "A simulated annealing algorithm for single machine scheduling problems with family setups. *Computers & Operations Research*, **36**(7), pp. 2133-2138 (2009).
[4] Poursabzi, O. Mohammadi, M. and Naderi, B. "An improved model and a heuristic for capacitated lot sizing and scheduling in job shop problems" *Scientia Iranica*, **25**(6), pp. 3667-3684 (2018).
[5] Sahling, F., Buschkühl, L., Tempelmeier, H. et al. "Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic" *Computers & Operations Research*, **36**(9), pp. 2546–2553 (2009).
[6] Carvalho, D.M. and Nascimento, M.C.V. "A kernel search to the multi-plant capacitated lot sizing problem with setup carry-over" *Computers & Operations Research*, **100**, pp. 43-53 (2018).
[7] Florian, M. Lenstra, J. K. and Rinnooy Kan, A.H.G. "Deterministic production planning algorithms and complexity" *Management Science*, **26**(7), pp. 669-679 (1980).
[8] Bitran, G. R. and Yanasse, H. H. "Computational complexity of the capacitated lot size problem" *Management Science*, **28**(10), pp. 1174-1186 (1982).
[9] Chen W. H. and Thizy, J. M. "Analysis of relaxations for the multi-item capacitated lot-sizing problem" *Annals of Operations Research*, **26**, pp. 29-72 (1990).
[10] Baldo, T.A., Santos, M.O., Almada-Lobo, B. et al. "An optimization approach for the lot sizing and scheduling problem in the brewery industry" *Computers & Industrial Engineering*, **72**, pp. 58-71 (2014).
[11] Gicquel, C. and Minoux, M. "Multi-product valid inequalities for the discrete lot-sizing and scheduling problem" *Computers & Operations Research*, **54**, pp. 12-20 (2015).
[12] Boonmee, A. and Sethanan K. "A GLNPSO for multi-level capacitated lot-sizing and scheduling problem in the poultry industry" *European Journal of Operational Research*, **250**(216), pp. 652-665 (2016).
[13] Ceschia, S. Di Gaspero, L. and Schaerf, A. "Solving discrete lot-sizing and scheduling by simulated annealing and mixed integer programming" *Computers & Industrial Engineering*, **114**, pp. 235-243 (2017).
[14] Curcio, E., Amorim, P., Zhang, Q. et al. "Adaptation and approximate strategies for solving the lot-sizing and scheduling problem under multistage demand uncertainty" *International Journal of Production Economics*, **202**, pp. 81-96 (2018).
[15] Wichmann, M.G. Johannes, C. and Spengler, T.S. "An extension of the general lot-sizing and scheduling problem (GLSP) with time-dependent energy prices" *Journal of Business Economics*, **89**, pp. 481–514 (2019).
[16] Toscano, A. Ferreira, D. and Morabito, R. "A decomposition heuristic to solve the two-stage lot sizing and scheduling problem with temporal cleaning" *Flexible Services and Manufacturing Journal*, **31**, pp. 142–173 (2019).
[17] Kaczmarczyk, W. "Valid inequalities for proportional lot-sizing and scheduling problem with fictitious microperiods" *International Journal of Production Economics*, **219**, pp. 236-247 (2020).
[18] Hu, Z. and Hu, G. "Hybrid stochastic and robust optimization model for lot-sizing and scheduling problems under uncertainties" *European Journal of Operational Research*, **284**(216), pp. 485-497 (2020).
[19] Mohammadi, M. "Designing an integrated reliable model for stochastic lot-sizing and scheduling problem in hazardous materials supply chain under disruption and demand uncertainty" *Journal of Cleaner Production*, **2020**, pp. 122621 (2021).
[20] Chen, Z. and Zhang, R. "A capital Flow-constrained lot-sizing problem with trade credit" *Scientia Iranica*, **25**(5), 2775-2787 (2018).
[21 Cheng, Y., Wang, W., Wei, C. et al. "An integrated lot-sizing model for imperfect production with multiple disposals of defective items" *Scientia Iranica*, **25**(2), pp. 852-867 (2018).
[22] Stadtler, H. and Meistering, M. "Model formulations for the capacitated lot-sizing problem with service-level constraints" *OR Spectrum*, **41**, pp. 1025–1056 (2019).
[23] Abrishami, S. Vahdani, H. and Rezaee, B. "An integrated lot-sizing model with supplier and carrier selection and quantity discounts considering multiple products" *Scientia Iranica*, **27**(4), pp. 2140-2156 (2020).
[24] Slama, I., Ben-Ammar, O., Dolgui, A. et al. "Genetic algorithm and Monte Carlo simulation for a stochastic capacitated disassembly lot-sizing problem under random lead times" *Computers & Industrial Engineering*, **159**, pp. 107468 (2021).

[25] Li, Y. Saldanha-da-Gama, F. Liu, M. et al. "A risk-averse two-stage stochastic programming model for a joint multi-item capacitated line balancing and lot-sizing problem" *European Journal of Operational Research*, Doi: 10.1016/j.ejor.2021.09.043 (2022).
[26] Malekian, Y. Mirmohammadi, S.H. and Bijari, M. "Polynomial-time algorithms to solve the single-item capacitated lot sizing problem with a 1-breakpoint all-units quantity discount" *Computers & Operations Research*, **134**, pp. 105373 (2021).
[27] Mula, J., Díaz-Madroñero, M., Andres, B. et al. "A capacitated lot-sizing model with sequence-dependent setups, parallel machines and bi-part injection moulding" *Applied Mathematical Modelling*, **100**, pp. 805-820 (2021).
[28] Koch, C., Arbaoui, T., Ouazene, Y. et al. "Capacitated multi-item lot sizing problem with client prioritization in tire industry" *IFAC-PapersOnLine*, **54**(1), pp. 564-569 (2021).
[29] Gansterer, M. and Födermayr, P. and Hartl, R.F. "The capacitated multi-level lot-sizing problem with distributed agents" *International Journal of Production Economics*, **235**, pp. 108090 (2021).
[30] Rezaei, S. and Behnamian, J. "Single-item lot-based supplying and batch production under a bilateral capacity reservation: A partnership structure" *RAIRO - Operations Research*, **55** (2021), pp. 2633-2652 (2021).
[31] Mohammadi, M. and Fatemi Ghomi, S.M.T. "Genetic algorithm-based heuristic for capacitated lot sizing problem in flow shops with sequence-dependent setups" *Expert Systems with Applications*, **38**, pp. 7201–7207 (2011).
[32] Behnamian, J., Fatemi Ghomi, S.M.T., Karimi, B. et al. "A Markovian approach for multi-level multi-product multi-period capacitated lot-sizing problem with uncertainty in levels" *International Journal of Production Research*, **55**(18), pp. 5330-5340 (2017).
[33] Behnamian, J., Fatemi Ghomi, S.M.T., Jolai, F. et al. "Realistic two-stage flowshop batch scheduling problems with transportation capacity and time" *Applied Mathematical Modelling*, **36**, pp. 723–735 (2012).
[34] Jans, R. and Degraeve, Z. "Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches" *European Journal of Operation Research*, **177**(3), pp.1851-1875 (2007).
[35] Swan, J., Adriaensen, S., Brownlee, A.E.I. et al. "Metaheuristics "In the Large"" *European Journal of Operational Research*, **297**(2), pp. 393-406 (2022).
[36] Behnamian, J., Fatemi Ghomi, S.M.T., Jolai, F. et al. "Minimizing makespan on a three-machine flowshop batch scheduling problem with transportation using genetic algorithm" *Applied soft computing*, **12**, pp. 768-777 (2012).
[37] Gen, M. and Cheng, R. "Genetic Algorithms and Engineering Designs". 1th ed. New York: Wiley (1997).
[38] Reeves, C.R. "A genetic algorithm for flow shop sequencing" *Computer & Operation Research*, **22**(1), pp. 5-13 (1995).
[39] Mohammadi, M., Fatemi Ghomi, S. M. T., Karimi, B. et al. "Rolling horizon and fix-and-relax heuristics for the multi-product multi-level capacitated lot sizing problem with sequence-dependent setups" *Journal of Intelligent Manufacturing*, **21,** pp. 501–510 (2010).

J. Behnamian obtained his Ph.D. in Industrial Engineering from Amirkabir University of Technology, Tehran, Iran. Currently, he is an Associate Professor at Industrial Engineering Department, Bu-Ali Sina University, Hamedan, Iran. His main research areas are: production scheduling, distributed systems, multicriteria decision making, and soft computing.

S.M.T. Fatemi Ghomi was born in Ghom, Iran on 11 March 1952. He received his B.Sc. degree in industrial engineering from Sharif University, Tehran in 1973, and the Ph.D. degree in industrial engineering from University of Bradford, England in 1980. He is currently a Professor in the Department of Industrial Engineering at the Amirkabir University of Technology at Tehran. His research and teaching interests are in stochastic activity networks, production planning and control, scheduling, queueing systems, and statistical quality control.

Behrooz Karimi is a Professor at the Department of Industrial Engineering and Management Systems at Amirkabir University of Technology. His education includes a bachelor's degree in Industrial Engineering from Amirkabir University of Technology, a master's degree in Industrial Engineering from Iran University of Science and Technology, and a Ph.D. in Industrial Engineering from Amirkabir University of Technology. His research and consulting activities are mainly in the areas of supply chain management, production planning and inventory management.

Mrs Marzieh Fadaei received her MSc degree in Industrial Engineering from Amirkabir University of Technology in Nov. 2012 under supervision of Dr. Fatemi Ghomi and advisory of Dr. Karimi.

Figures' list

Tables' list

Figures

	Level 1					Level 2				
Family	Family1		Family2			Family1		Family2		
Period1	Product 1	Product 2	Product 3	Product 4	Product 5	Product 1	Product 2	Product 3	Product 4	Product 5
	80	50	50	20	50	10	50	50	20	50
Products' sequence	2→ 1		5→ 4 → 3			1→ 2		3→ 5 → 4		
Family's sequence	2→ 1					1→ 2				
Period2	Product 1	Product 2	Product 3	Product 4	Product 5	Product 1	Product 2	Product 3	Product 4	Product 5
	0	0	0	70	30	70	0	0	70	30
Products' sequence	-		4→ 5			1		5→4		
Family's sequence	2					2→ 1				
Period3	Product 1	Product 2	Product 3	Product 4	Product 5	Product 1	Product 2	Product 3	Product 4	Product 5
	0	40	50	0	40	0	40	50	0	40
Products' sequence	2		3→5			2		3→5		

Family's sequence	1→ 2		2→ 1

Figure 1: the solution of the numerical example

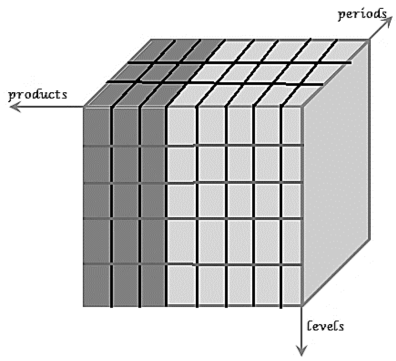

Figure 2: A sample chromosome

Kruskal-Wallis Rank Sum Test Results

Parameter	Value
Kruskal-Wallis chi-squared	0.1479
DF	1
p-value	0.7005

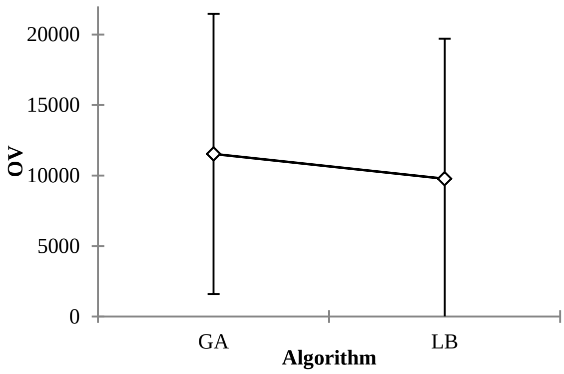

Figure 3: Statistical analysis

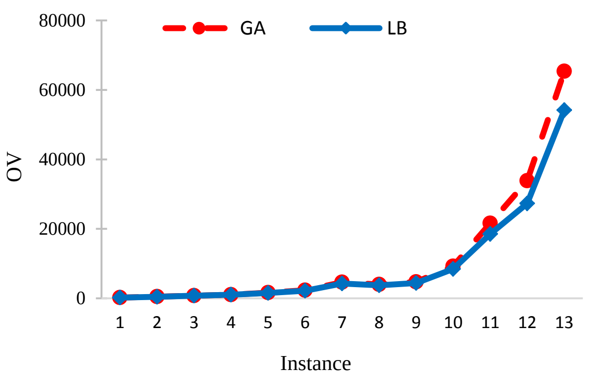

Figure 4: Comparison results

Tables

Table 1: The demand for products

Family, Product		1,1	1,2	2,3	2,4	2,5
Period	**1**	10	30	20	20	50
	2	20	40	30	30	30
	3	50	20	50	40	40

Table 2: The variable production time of products

Machine 1					Machine 2				
Family 1		Family 2			Family 1		Family 2		
Product 1	Product 2	Product 3	Product 4	Product 5	Product 1	Product 2	Product 3	Product 4	Product 5

Period 1	1	2	1.5	3	1	2	1.5	1	1	2
Period 2	2	1	2.5	1.5	2	1.5	2	2.5	2	2.5
Period 3	1.5	1	2	2.5	1.5	1.5	1	1	1	1

Table 3: The variable production cost of products

	Machine 1					Machine 2				
	Family 1		Family 2			Family 1		Family 2		
	Product 1	Product 2	Product 3	Product 4	Product 5	Product 1	Product 2	Product 3	Product 4	Product 5
Period 1	3	5	3	1	4	1	3	2	1	4
Period 2	4	2	3	4	2	2	2	4	1	3
Period 3	5	4	2	3	2	4	2	2	3	5

Table 4: The holding cost of products

	Machine 1					Machine 2				
	Family 1		Family 2			Family 1		Family 2		
	Product 1	Product 2	Product 3	Product 4	Product 5	Product 1	Product 2	Product 3	Product 4	Product 5
Period 1	0.5	0.3	0.5	0.3	0.25	0.2	0.6	0.3	0.8	0.6
Period 2	1	0.25	0.1	0.75	0.45	0.5	0.7	0.2	0.3	0.25
Period 3	0.25	0.7	0.3	0.1	0.25	0.5	0.3	0.4	0.2	0.5

Table 5: The capacity of machines

	Machine 1	Machine 2
Period 1	600	400
Period 2	400	800
Period 3	600	500

Table 6: The demands of Families 1 and 2 in the planning horizon.

	Family	1				2				
	Product	1	2	3	4	5	6	7	8	9
Period	**1**	10	20	50	30	20	30	50	20	30
	2	20	40	10	20	40	50	30	40	20
	3	50	30	20	40	50	20	30	20	40

Table 7: Dimension of problems used in the calibration of parameters

Problems / **Parameters**	**1**	**2**	**3**	**4**
M	2	5	7	10
Z	2	5	7	10
T	2	5	7	10
$\sum_{j=1}^{m} r_j$	4	15	21	50

Table 8: Comparison of the optimal solution and the lower bound of model P^2

No.	**Main model**		**Lower bound**		**Difference**
	OV	**Runtime (s)**	**OV**	**Runtime (s)**	
1	3815	12	3745	<1	%1.8
2	4295	30	4200	<1	%2.2
3	6633	417	6475	<1	%2.4
4	9463	1264	9223	1	%2.5

OV: Objective function value

Table 9: Comparison of the proposed model and modified Mohammadi and Fatemi Ghomi (2011)

No.	Problem size	Proposed model		Modified Mohammadi and Fatemi Ghomi's model	
		OV	Runtime (s)	OV	Runtime (s)
1	m=2; Z=2; T=2; r_1=2; r_2=1;	674	<1	674	130
2	m=2; Z=2; T=3; r_1=5; r_2=4;	4295	16	4295	357
3	m=2; Z=3; T=3; r_1=5; r_2=4;	6633	417	6633	9592
4	m=3; Z=3; T=4; r_1=3; r_2=4; r_3=3;	12328	853	-	>14400

Table 10: Comparison of developed lower bound and proposed GA

Problem size				Genetic algorithm		Lower bound	Percentage of differences
M	Z	T	$\sum_{j=1}^{m} r_j$	OV	Runtime	OV	%
2	2	2	5	216	52.41	211.93	%1.9
2	3	3	5	461.21	89.46	450.86	%2.2
2	3	3	9	766.36	64.23	749.5	%2.3
2	3	5	9	1058.2	91.09	1017.9	%3.8
3	5	5	12	1633	295.62	1563.4	%4.3
3	5	7	12	2341.25	312.53	2169.87	%7.3
3	7	7	14	4624.81	439.31	4250.2	%8.1
4	3	3	15	3992.43	250.87	3736.914	%6.4
4	3	5	15	4731.43	747.29	4395.498	%7.1
4	7	7	15	9247.21	1562.45	8433.456	%8.8
5	7	10	20	21580.64	3939.82	18537.77	%14.2
5	10	10	20	33862.44	2793.5	27293.13	%19.4
5	15	15	20	65371.39	3738.29	54192.88	%17.1